UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE INFORMÁTICA

Processing Temporal Information in Unstructured Documents

Francisco Nuno Quintiliano Mendonça Carapeto Costa

DOUTORAMENTO EM INFORMÁTICA
ESPECIALIDADE CIÊNCIA DA COMPUTAÇÃO

2012

UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE INFORMÁTICA

Processing Temporal Information in Unstructured Documents

Francisco Nuno Quintiliano Mendonça Carapeto Costa

Tese orientada pelo Prof. Doutor António Horta Branco, especialmente elaborada para a obtenção do grau de doutor em Informática (especialidade Ciência da Computação).

2012

Abstract

Temporal information processing has received substantial attention in the last few years, due to the appearance of evaluation challenges focused on the extraction of temporal information from texts written in natural language.

This research area belongs to the broader field of information extraction, which aims to automatically find specific pieces of information in texts, producing structured representations of that information, which can then be easily used by other computer applications. It has the potential to be useful in several applications that deal with natural language, given that many languages, among which we find Portuguese, extensively refer to time. Despite that, temporal processing is still incipient for many language, Portuguese being one of them.

The present dissertation has various goals. On one hand, it addresses this current gap, by developing and making available resources that support the development of tools for this task, employing this language, and also by developing precisely this kind of tools. On the other hand, its purpose is also to report on important results of the research on this area of temporal processing. This work shows how temporal processing requires and benefits from modeling different kinds of knowledge: grammatical knowledge, logical knowledge, knowledge about the world, etc. Additionally, both machine learning methods and rule-based approaches are explored and used in the development of hybrid systems that are capable of taking advantage of the strengths of each of these two types of approach.

Keywords: Natural Language Processing, Information Extraction, Temporal Information Processing, Machine Learning, Rule-Based Methods

Resumo

O processamento de informação temporal tem recebido bastante atenção nos últimos anos, devido ao surgimento de desafios de avaliação focados na extração de informação temporal de textos escritos em linguagem natural.

Esta área de investigação enquadra-se no campo mais lato da extração de informação, que visa encontrar automaticamente informação específica presente em textos, produzindo representações estruturadas da mesma, que podem depois ser facilmente utilizadas por outras aplicações computacionais. Tem o potencial de ser útil em diversas aplicações que lidam com linguagem natural, dado o caráter quase ubíquo da referência ao tempo cronólogico em muitas línguas, entre as quais o Português. Apesar de tudo, o processamento temporal encontra-se ainda incipiente para bastantes línguas, sendo o Português uma delas.

A presente dissertação tem vários objetivos. Por um lado vem colmatar esta lacuna existente, desenvolvendo e disponibilizando recursos que suportam o desenvolvimento de ferramentas para esta tarefa, utilizando esta língua, e desenvolvendo também precisamente este tipo de ferramentas. Por outro serve também para relatar resultados importantes da pesquisa nesta área do processamento temporal. Neste trabalho, mostra-se como o processamento temporal requer e beneficia da modelação de conhecimento de diversos níveis: gramatical, lógico, acerca do mundo, etc. Adicionalmente, são explorados tanto métodos de aprendizagem automática como abordagens baseadas em regras, desenvolvendo-se sistemas híbridos capazes de tirar partido das vantagens de cada um destes dois tipos de abordagem.

Palavras Chave: Processamento de Linguagem Natural, Extração de Informaçao, Processamento de Informação Temporal, Aprendizagem Automática, Métodos Baseados em Regras

Acknowledgements

I feel indebted to my family and my friends for all the help and support during the years of research leading to this dissertation.

I would also like to thank my supervisor and all the colleagues at the NLX group, where I developed this work over the last years. Obviously, my work has benefited directly from the tools that have been developed in the group, and which I made use of for this dissertation, as well as from their cooperation, but more important than that was what I learned from their insights and comments, and also the motivation I gained from their support.

The work described in this thesis was funded by Fundação para a Ciência e Tecnologia, under the scholarship SFRH/BD/40140/2007. My research leading to this thesis would not have been possible without this funding.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Consider the following couple of paragraphs, taken from the English Wikipedia article on James Joyce:[1]

> James Augustine Aloysius Joyce was **born** on **2 February 1882** to John Stanislaus Joyce and Mary Jane "May" Murray in the Dublin suburb of Rathgar. He was **baptized** in the nearby St. Joseph's Church in Terenure on **5 February** by Rev. John O'Mulloy. (. . .) In **1887**, his father was **appointed** rate collector (i.e., a collector of local property taxes) by Dublin Corporation; the family subsequently **moved** to the fashionable adjacent small town of Bray 12 miles (19 km) from Dublin. Around **this time** Joyce was **attacked** by a dog, which **engendered** in him a lifelong cynophobia. He also suffered from keraunophobia, as an overly superstitious aunt had described thunderstorms to him as a sign of God's wrath.
>
> In **1891**, Joyce **wrote** a poem, *Et Tu Healy*, on the death of Charles Stewart Parnell. His father was angry at the treatment of Parnell by the Catholic church and at the resulting failure to secure Home Rule for Ireland. The elder Joyce had the poem printed and even sent a part to the Vatican Library. In **November** of that same year, John Joyce was **entered** in Stubbs Gazette (a publisher of bankruptcies) and **suspended** from work. In **1893**, John Joyce was **dismissed**

[1] http://en.wikipedia.org/wiki/James_joyce, retrieved on October 23, 2012.

> with a pension, beginning the family's slide into poverty caused mainly by John's drinking and general financial mismanagement.

The highlighted expressions in this small text represent some of the mentioned events and dates. They can be organized in a time line, which associates the events with the points in time when they are said to have occurred:

1882-02-02	Joyce is born
1882-02-05	Joyce is baptized
1887	Joyce's father is appointed rate collector
	Joyce's family moves to Bray
	Joyce is attacked by a dog, developing a lifelong fear
1891	Joyce writes the poem *Et Tu Healy*
1891-11	Joyce's father is suspended from work and bankrupt
1893	Joyce's father is dismissed with a pension

The topic of this thesis is how this can be performed automatically and with high quality. Essentially, the goal is to extract structured information about time from unstructured text. This sort of task can be useful in several applications, as presented later in this chapter, but as this example shows, even on its own it can be used to organize data in a way that can make it faster and easier to grasp and visualize.

In this introductory chapter, we start by presenting natural language processing in general in Section 1.1. Section 1.2 is a quick introduction to the problems that temporal information processing is concerned with. In Section 1.3, we list some of the applications that temporal processing can help improve. They consist of other tasks of natural language processing that can benefit from knowing detailed information about the temporal information conveyed in natural language texts. Section 1.4 then describes the scope of our work and why the problems of temporal processing are challenging. In Section 1.5 we present the goals and contributions of our research. Finally, Section 1.6 outlines the way this thesis is organized and briefly talks about the content of each of the remaining chapters.

1.1 Background

The field of **natural language processing** (NLP) is concerned with creating computational systems that can deal with natural language text in a way similar to the way humans do. The motivation behind this goal is not merely the creation of such applications but also a theoretical one: in achieving that goal we may also develop a model that helps us understand how humans handle language.

Natural language technology has become essential for the efficient access to the ever-increasing amount of information made available in the information society (Branco *et al.*, 2012). Accordingly, this technology is increasingly present in every day life, materialized in computer programs that are able to:

- perform spell checking and grammar checking,
- produce a short summary of a document or of a collection of documents (automatic summarization),
- translate text between different natural languages (machine translation),
- search the World Wide Web (or any other knowledge base) for specific answers to questions input by a human user (question-answering),
- extract key-words from documents,
- identify a document's topic, textual genre, author or language, etc. (text classification).

Other kinds of systems and applications in this field are maturing and present great potential.

Much of NLP consists of extracting information that is conveyed in an unstructured manner (natural language) and present it in a structured representation. This allows computation to be done on the information conveyed by the original unstructured data. The popularity of the World Wide Web is making available increasing amounts of information encoded in natural language. Concomitantly, NLP has seen unprecedented interest in recent years.

The world is dynamic, and change is a part of it. Similarly, natural language has several different means to describe when and for how long something happens or stays unchanged. Indeed, a large proportion of the information carried by natural language is temporally circumscribed, as reference to time is widespread in natural language.

Despite the importance and the high frequency with which time is referred to in natural language, a rigorous and comprehensive understanding of it and the development of automatic systems able to extract the temporal meaning of texts have been difficult tasks. Still, the last few years have seen a large amount of research focused on the problem of temporal information processing, with the goal of making NLP systems more sophisticated.

1.2 Temporal Information Processing

Temporal processing focuses on extracting from a text a specific kind of information—namely information about time. The task of temporal information processing can be illustrated with the following paragraph, taken from a news article:

> In Washington *today*, the Federal Aviation Administration **released** air traffic control tapes from *the night* the TWA Flight eight hundred **went** down.

A temporal information system that receives a document containing this piece of text as input will produce a structured representation of its temporal content, that may include:

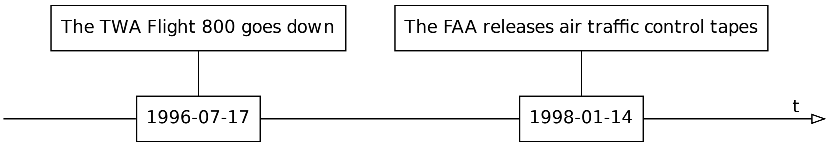
- **The date when the document was created**
  The example paragraph does not make it possible to identify this date, but other information in the same document might allow to extract the date of e.g. 1998-01-14.

- **The dates and times referred to in the text**
  The word *today* refers to the document creation date, and this paragraph also mentions *the night* when a plane crashed (which is 1996-07-17; even though this sentence does not contain that information, it may be present in other parts of the same document).

- **The situations that are described in the document**
  This paragraph describes a situation in which tapes are **released** as well as another one in which a plane **went** down.

- **The temporal relations holding between these situations and dates or times**
  The night the plane went down temporally precedes the day that sentence was written. The situation in which the tapes are made public temporally overlaps this date, and the date associated with the expression *the night* overlaps the plane crash.

Such a representation can be depicted graphically:


The TWA Flight 800 goes down
The FAA releases air traffic control tapes
1996-07-17
1998-01-14
t

In this graph, temporal overlap is represented by a vertical line and temporal precedence is represented horizontally with an arrow. The temporal representation output by a temporal information system is typically not a graphical representation, but it is precise enough that it can be automatically converted into one if needed.

1.3 Applications

Temporal processing can be useful to improve many of the natural language applications mentioned in the beginning of this text.

For instance, it can be useful to **question-answering** systems (Prager *et al.*, 2000). Question-answering systems search a large text collection in order to output a short sentence or phrase that is a precise answer to a user's question. The integration of temporal components in question-answering systems can be useful when the question is of a temporal nature, but it is just starting and is still quite limited, as in Bobrow *et al.* (2007).

> In **1957**, Tolkien was to travel to the United States to accept honorary degrees from Marquette, Harvard, and several other universities, and to deliver a series of addresses, but the trip was canceled due to the ill health of his wife Edith. He retired **two years later** from his professorship at Oxford.

> "The Adventures of Tom Bombadil" was published in **1962**, **three years after** Tolkien retired from his professorship at Oxford.

> Tolkien makes a brief allusion to the future of Middle-Earth in a letter written in **1958**. **The following year**, after his retirement from teaching at Oxford, he . . .

Figure 1.1: Example documents accessible to a question-answering system

In fact, the motivation behind much of the initial work in this field was to improve question-answering systems. Consider for instance the examples in Figure 1.1, taken from Negri & Marseglia (2004), and the question *When did J. R. R. Tolkien retire from his professorship at Oxford?*. It is not at all trivial for a question-answering system to answer that question based on those example documents. Question-answering systems over the web still fail short from adequately addressing questions such as *Is Bill Clinton* **currently** *the President of the United States?*. These difficulties motivate research specifically on temporal question answering, e.g. that of Saquete *et al.* (2004), Harabagiu & Bejan (2005), Ahn *et al.* (2006), or Tao *et al.* (2010).

Temporal information is also important for **automatic summarization** (Mani, 2001). This research field is concerned with automatically producing a summary of a text or a collection of texts. Automatic summaries are often produced by selecting some of the sentences that appear in the original texts. During the step of ordering these sentences, temporal information is useful, because summaries are more readable if the sentences are presented according to the chronological order of the events they describe. Systems that produce summaries for a document at a time often just present the sentences in the order that they appear in the text. This might not correspond to the chronological order. Additionally, recent summarization systems can produce a summary for a collection of documents input by the user. In this case, this heuristic for ordering the sentences in the final summary is not even available.

Most of the Web's content today is still designed for humans to read, not for computer programs to manipulate meaningfully, but there has been immense progress in that direction: today, the **Semantic Web** (Berners-Lee *et al.*, 2001) is assumed to be attainable. (Shadbolt *et al.*, 2006). Natural language processing can help us reach the Semantic Web, as it can be used to automatically annotate human readable content with structured content that can easily be further manipulated by computers. Temporal processing would play an important role here, as temporal information is ubiquitous in natural language.

Temporal processing can be seen as a special case of **information extraction** (Cowie & Lehnert, 2000), which therefore also benefits from the former. Information extraction consists precisely in automatically extracting structured information from unstructured documents. Due to the difficulty of the problem, current approaches to information extraction are restricted to specific domains or sub-tasks. For instance, an information extraction system may focus on extracting attributes of entities (such as biographical data of people) or specific relations between entities (e.g. which company or institution employs who). Because some of this information changes with time (e.g. the last example might rather be about which company or institution employs who when), temporal processing is relevant to information extraction in general.

Finally, temporal processing can help other, more specific tasks. These include the task of event co-reference resolution (Bejan & Harabagiu, 2010), which is about determining whether different terms or expressions referring to events in a text refer to the same event: they cannot relate to the same event if they point to events that happen in different times. Another example is the area of medical natural language processing and decision support systems in medicine (Augusto, 2005; Zhou & Hripcsak, 2007), as this field often deals with the clinical history of patients.

1.4 Scope and Challenges

Temporal Processing of Portuguese As already mentioned, the present work focuses on extracting temporal information from text. In addition, it is centered around the processing of text written in a specific natural language—Portuguese. As the following chapters will show, the state of the art can be improved upon by using increasingly language specific solutions. For this reason, it is important to extend the work on temporal processing to other languages besides the one most studied in the literature, which is English.

An initial obstacle to the temporal processing of many languages, including Portuguese, is the lack of language data annotated with explicit information about time. Collections of texts annotated with temporal information exist for very few languages. This sort of data supports the development of machine learning solutions as well as the evaluation of the temporal processing systems resulting from work on this topic. In order to address this issue, we created a corpus of Portuguese text with temporal annotations. This corpus, TimeBankPT, is described in Chapter 3. It is freely available, and it is one of the central contributions of the present work.

Challenges of Temporal Processing Temporal processing involves dealing with several problems. They are explored in this thesis, too:

- Determining the **document creation time**
  The document creation time is necessary to interpret many temporal expressions (for instance, *hoje* "today"). Sometimes this may be trivial to determine (e.g. in a set of similar documents, each document's creation time may be systematically placed at a specific conventional point in the document). In other cases, it deserves deeper processing.

- Delimiting, classifying and normalizing **temporal expressions**
  Temporal expressions (or timexes, for short), like *hoje* "today" or *o ano passado* "last year", need to be identified and delimited. Temporal expressions can take a variety of linguistic forms, such as noun phrases or adverbial phrases. In Portuguese, they can also be verb phrases headed by the verb *haver* "there to be" or *fazer* "to do":

  – *o século XX* "the 20th century", *23 de Outubro* "October 23", *o dia de Natal* "Christmas day", *esse ano* "that year", *dois anos antes* "two years before", *hoje* "today", *recentemente* "recently", *há dois dias* "two days ago", *faz hoje um ano* "a year ago".

Furthermore, they are to be classified, for instance, as denoting a time or a date, like the examples just above, or a duration, as in *(durante) dois dias* "(for) two days", *(em) vinte segundos* "(in) twenty seconds". A temporal expression can be ambiguous between denoting a temporal location and a duration. For example, the expression *em dois dias* "in two days" can denote a date (with the meaning *two days from now*, as in the example sentence in (1), or a duration, as in (2).

(1) Tomará posse em dois dias.
will take possession in two days
He will assume the position in two days.

(2) O rio subiu cinco metros em dois dias.
the river rose five meters in two days
The river rose five meters in two days.

Temporal expressions should also be normalized. That is, it is necessary to determine the specific date or period that they refer to, and represent that in a standardized, unambiguous way. For instance, *22/02/2010*, *22 de Fevereiro de 2010*, and *22.Fev.2010*, etc. should receive a common, normalized representation, as they all refer to the date of February 22, 2010.

It is necessary to know the document creation time in order to normalize expressions like *ontem* "yesterday" (which refers to the day before the date when the document was created) or *na próxima semana* "next week." However, many other expressions are anchored in a different time: *o ano seguinte* "the following year", *o dia anterior* "the previous day", *dois dias depois* "two days after". These expressions are anchored in a date or time that has been previously mentioned in the document.

- Recognizing **events** and veridicality

  Most verb occurrences denote events that can be temporally located or delimited. Some nouns can also denote events (*conferência* "conference", *venda* "sale", *acidente* "accident", etc.) but not all of them (*pessoa* "person", *comboio* "train", etc.).

Additionally not all event terms denote events that happen in the real world. They can instead refer to alleged or hypothetical events. In that case it may not even make sense to order them with respect to the real events. Determining this depends on syntactic context (for instance, being in the complement of a verb like *dizer*/*say*). In the following examples, (3a) and (3b) are equivalent, but (3c) does not entail (3a) or (3b) as it does not assert that Maria arrived (and the first two examples do). In these examples, the temporal connectors (*antes de* vs. *antes que*) are responsible for this contrast. As these examples show, it is important to consider the "details."

(3) a. O Pedro saiu **antes de** a Maria chegar.
the Pedro left before the Maria arrive
Pedro left before Maria arrived.

b. A Maria chegou depois de o Pedro sair.
the Maria arrived after the Pedro leave
Maria arrived after Pedro left.

c. O Pedro saiu **antes que** a Maria chegasse.
the Pedro left before the Maria arrived
Pedro left before Maria arrived.

- Classifying **temporal relations** between events
  Many linguistic devices are used to describe the temporal order of events. The order in which sentences and events are presented in the text is relevant, as in the example in (4a), where it reflects the chronological order between the described events.

(4) a. *Kim came in. Sue left.*

b. *Kim came in after Sue left.*

c. *Kim came in. Sue had left.*

d. *Kim came in. Sue was leaving.*

e. *Kim came in. Sue had the flu.*

f. *Kim fell down. Sue pushed him.*

However, several other factors come into play, and they can override textual order. First, the order of events can be explicitly stated, as in (4b). Grammatical tense and grammatical aspect are also relevant, as exemplified in (4c) and (4d): in (4c) the order of events is reversed with respect to textual order; and in (4d) the two mentioned events overlap.

Another important factor is *Aktionsart*, also called aspectual type, situation type or lexical aspect. Verbs and other event terms can be classified according to the physical properties of the situations that they describe as they are perceived by humans. This has consequences in the syntactic behavior of these elements, therefore the aspectual classes can be empirically tested. For instance, a major distinction is between states and other types. One syntactic test to differentiate between them is whether the situation can be described as occurring in the present by means of the simple present tense or if the progressive must be used: for states the progressive is not necessary (e.g. *I am happy*, *I want this*, *Sue has the flu* can denote situations that hold true at the exact time that these sentences are uttered), but for non-states it is necessary (e.g. *he paints* does not mean *he is painting at this moment* but rather *he knows how to paint* or *he often paints*). In (4e) the two situations also overlap. The difference between (4a) and (4e) is that the second sentence of (4e) is stative, but the second sentence of (4a) is not. Indeed, it is often claimed that, in simple narratives, non-stative sentences move the action forward in time, while the state sentences instead describe how things are at the time of the last-mentioned event (Hinrichs, 1986; Kamp & Rohrer, 1983; Lascarides & Asher, 1993; Partee, 1984).

Finally, pragmatics and world knowledge are also relevant. In (4f) the order of the two events described does not correspond to the order in which they are presented in the most natural interpretation, because of a causality link between pushing and falling.

When there are temporal connectors like *before*/*after*/*when*, there may still be ambiguity to resolve. For instance, *when* does not always mark temporal overlap between two situations. Depending on other factors, like tense, (grammatical or lexical) aspect and causality, it can convey other temporal relations, as in the sentence in (5), where the second mentioned event (the one of publishing the altered photograph) temporally precedes the first one (the one of receiving many protests).

(5) A National Geographic recebeu muitos protestos quando publicou uma fotografia alterada das pirâmides.
National Geographic saw many protests when it published an altered photograph of the pyramids

- Classifying **temporal relations** between events and times or durations

  This task consists in determining the temporal ordering between mentioned events and mentioned times.

  There are several textual cues that can be explored to address this problem. The most obvious cue is the linguistic material used to connect the two, as in *The gallery opened in/before February*: here, *in* signals temporal overlap between the opening event and the time interval denoted by *February*, whereas *before* signals temporal precedence.

  This task is relatively easy when the time and event to be temporally ordered are mentioned close to each other in the text. When they are further apart, the relation is indirect and often determined by inference or world knowledge:

(6) O presidente e os seus assessores principais decidiram provisoriamente em 11 de fevereiro que uma guerra terrestre seria necessária.
The president and his top aides tentatively decided on Feb. 11 that a ground war would be necessary.

According to this example, the president's decision temporally overlaps the date of February 11 and precedes the war (if this war ever occurs). The temporal relation between this date and the war is indirect. Assuming there is a gap of at least one day between the decision and the start of the war, then the war does not overlap the mentioned date, and if this is so then it must necessarily follow it (since it follows the decision, which overlaps that date).

There is ambiguity to resolve also with this task. For example, the sentence *Kim had finished the cake yesterday* is ambiguous. In one interpretation, the date and the event temporally overlap. In the other one, the event precedes the date.

1.5 Goals and Contributions

The main goal of this thesis is to improve temporal processing, not just of Portuguese (which, before our work, was almost non-existent), but also in general.

The present work describes a novel contribution to the processing of the linguistic expression of time by means of the integration of data-driven and knowledge-rich methods at different stages of processing. At an early stage of processing, temporal extraction technology based on probabilistic approaches is enriched with sophisticated information of different kinds, such as linguistic knowledge and logic. The outcome of this temporal information extraction system is then, at a later stage, combined with the meaning representations produced by a deep, rule-based, processing grammar.

With the present contribution towards a full-fledged processing of time, our work adds to the overall discussion and quest on how to obtain progresses in natural language processing by means of hybrid systems that combine the complementarity of the symbolic and probabilistic approaches in a way that their strengths can be amplified and their shortcomings mitigated.

The present thesis contains a detailed account of the issues related to the temporal processing of natural language. The main contributions of this research are:

- Developing a corpus of Portuguese text with temporal annotations

  This data set supports the development and testing of temporal processing technology for Portuguese. A collection of texts annotated with temporal information is necessary for the development of temporal processing solutions for a new language. Such a data set enables the development of machine learning approaches by making training data available. Additionally, it provides the means to objectively test the developed solutions, by making available test data. This data set is also made publicly available. Because of this, it also enables direct comparison of our results with any future work that makes use of it. This corpus is described in Chapter 3.

- Developing state-of-the-art temporal extraction technology for Portuguese

  This contribution is effected by carrying out all the tasks described previously, namely: (i) detecting events mentioned in text; (ii) detecting time expressions in text and representing the times and dates they denote in a standard, unambiguous way; and (iii) classifying the temporal relations holding between these entities. Combining all of these tasks, it is possible to automatically organize the information that is presented in a text in a time line. This endeavor is described in Chapter 5.

- Improving the automatic classification of temporal relations

  The most interesting task within temporal processing is temporal relation classification. At the moment, it is the hardest to solve, presenting the highest error rates. The present study aims to experiment with different and novel strategies that can improve this task. More specifically, we incorporate many different types of knowledge sources: not only different types of grammatical information but also lexical information, reasoning, and even knowledge about the world. Chapter 4 discusses this work.

- Improving the deep language processing of temporality

  An existing computational grammar for Portuguese was extended with a temporal module. This grammar delivers the phrase structure of an input sentence, as well a representation of its meaning in terms of truth conditions. This meaning representation was extended with information about time. Chapter 5 describes this implementation. It includes some novel and improved analyses of challenging linguistic phenomena related to temporality.

- Improving full-fledged temporal processing

  The temporal module implemented in this grammar makes exclusive use of grammatical information. Also, since the grammar processes sentences in isolation, the larger context is missing. In order to address these and other limitations that will be explained, we also implement a post-processor that extends the meaning representations output by the grammar with the information coming from the temporal extractor. This combination illustrates an application of temporal processing and leads to an enhanced representation of time in the meaning representations output by a computational grammar for the deep processing of Portuguese. This part is described in Chapter 5.

1.6 Outline of the Thesis

This thesis is organized in the following manner.

Chapter 2 presents the related work. Here some fundamental concepts about the way time is mentioned in natural language are introduced. The chapter starts by presenting the work on which temporal processing is based, drawing from the fields of Linguistics, Logic and Artificial Intelligence. These disciplines have been concerned with the way that temporality is conveyed in natural language and with reasoning about time. It then addresses more recent work, specifically in the area of temporal information processing, which has flourished with the recent development of annotation standards, annotated data sets, evaluation competitions and a large body of research based on these resources.

Chapter 3 describes TimeBankPT. This data set is used for the development and testing of the technology presented in the following chapters. To develop TimeBankPT, an existing resource of English data with temporal annotations was translated to Portuguese, adapting the existing annotations. We explain how this adaptation was carried out, and we also explain the format and meaning of the temporal annotations that are used in the original data and in TimeBankPT. A quantitative comparison between the original English corpus and TimeBankPT is also presented. Finally, we check whether the size of TimeBankPT is adequate, and describe an automated error mining procedure that was applied to the corpus in order to guarantee consistent annotations.

Chapter 4 focuses on the most difficult and interesting problem of temporal processing: classifying temporal relations between various kinds of elements (events and times). The approach taken in this chapter is to use machine learning techniques to tackle this issue. The chapter presents a series of different classifier features that are tested with the purpose of improving this task. We explore many different types of information, from morphology and syntax to semantics and even pragmatics, presenting motivating examples for trying them out, and discussing how they are implemented and then tested.

Chapter 5 is about applications. The first part of this chapter presents an effort to replicate for Portuguese the remaining tasks of temporal processing. Together with the temporal relation classifiers developed in the previous chapter, the result of this is full temporal annotation for Portuguese, materialized in a temporal extraction system. A second contribution of this chapter is the expansion of an existing deep computational grammar for Portuguese with a temporal module. Because this module does not make it possible to extract as much temporal information from input text with the grammar as what the temporal extraction system can, the two are combined, extending the output of the grammar with information coming from the temporal extractor.

Finally, Chapter 6 summarizes the main achievements of this study and discusses its limitations, proposing ways forward.

Chapter 2

Related Work

This chapter presents the work on which temporal processing is based, as well as some recent work on the computational processing of time phenomena in natural language. A large contribution comes from the fields of Linguistics and Logic, which have focused on the issues of time in natural language and temporal reasoning for decades now. The field of Artificial Intelligence also produced work that is relevant to our problem.

Temporal information processing has flourished quite recently. The present century has seen the development of annotation standards, annotated data sets, evaluation competitions and a large body of research based on these resources.

2.1 Outline

This chapter is organized in the following way. In Section 2.2 we present some of the foundational work on the topics of tense, aspect and temporal reasoning. It draws from related areas, like Linguistics, Logic and Artificial Intelligence. We then turn our attention to the computational processing of time phenomena in natural language, mentioning some of the early approaches in Section 2.3.

Recent years have seen the appearance several competitions and the development and maturing of annotation schemes and data sets relevant to this task. In Section 2.4 we talk about TERN 2004, which was a competition focusing on the processing of time expressions, such as those used to refer to times and dates, in

English and Chinese. In Section 2.5 we discuss TimeML, the current *de facto* annotation standard for temporal phenomena, as well as available corpora annotated with it. In addition to time expressions, TimeML covers the annotation of events mentioned in text, as well as the temporal relations holding between these events and the times and dates mentioned in the same text. The following sections, Section 2.6 and Section 2.7, are about the two TempEval competitions, that made use of similarly annotated data. They attracted participants working on English and Spanish. There have been efforts on the temporal processing of other languages. Section 2.8 lists some of the more recent corpora annotated with time phenomena. Many of them feature new languages.

These data sets have fueled much of the recent research on temporal processing. They have been used not only by the participants of these competitions (TERN 2004, TempEval, etc.), but also by much of the work published outside them. Section 2.9 presents some of the more recent approaches to the problem of temporal information processing.

2.2 Seminal Work

A large body of work on information on time and the ways in which it can be conveyed in natural languages—such as tense and aspect—can be found in the areas of Linguistics, Logic and Artificial Intelligence. Here we try to present some of the most important early work on the topics of handling references to time in natural language and reasoning about time. It is fundamental to much of the subsequent work done in the area of computational linguistics and natural language processing.

Many of these seminal papers have been collected in Mani *et al.* (2005). The book is organized in several parts, and each part contains introductory material that provides a very good summary of the problems concerning time in natural language. Both theoretical work and more practical approaches, concerned with applications, are referred.

2.2.1 Tense

According to Comrie (1985), **tense** is “the grammaticalized expression of location in time.” In this respect, grammaticalization requires that it is expressed obligatorily and that it is morphologically bound. That is, tense is typically conveyed by inflectional morphology rather than by separate words or phrases that are optional in a sentence (such as adverbs or adverbial phrases), even though the latter can also be employed to describe time. Many languages, such as English or Portuguese, use verbal morphology to encode tense.

The description of the meaning of tense owes much to the work of Reichenbach (1947). He observes that three points in time are necessary to account for past perfect forms, such as the sentence in (7).

(7) John had left on Monday.

These three points are the **point of speech** S (when the sentence is uttered), the **point of the event** E (when the event being described occurred) and the **point of reference** R (a third point that can sometimes be described by modifiers such as the phrase *on Monday* in (7)). A sentence with a past perfect verb can be ambiguous. For instance, the sentence in (7) has two readings. In one reading, the event of John leaving occurs on Monday. In the other reading, this event has already happened on Monday, i.e. it precedes Monday. In both readings, E is when the event of John leaving takes place. In the first reading, the phrase *on Monday* is used to describe the temporal location of E, but in the second reading it describes R.

Reichenbach's contribution consists in (i) defining these three times, (ii) using this three-point system for all tenses, (iii) defining the meaning of the different tenses through temporal relations between S and R on the one hand and R and E on the other—the temporal relation between S and E is not represented directly—and (iv) resorting to only two temporal relations: simultaneity and precedence.

Table 2.1 presents Reichenbach's analysis of the English tense system. Each cell in that table corresponds to a combination of two temporal relations that results in a specific semantic value of tense (simultaneous present, simultaneous future, etc.), which in turn is associated with different grammatical tenses (such as the English simple present, simple future, etc.). Temporal simultaneity is represented with a comma (,) and temporal precedence with a dash ($-$). It must be noted that English makes no grammatical distinction between posterior present (e.g. *Now I shall go*, where *now* identifies R) and simultaneous future (e.g. *I shall go tomorrow*, where

	S, R	$S - R$	$R - S$
E, R	*simultaneous present* simple present: *I see John*	*simultaneous future* simple future: *I'll see John*	*simultaneous past* simple past: *I saw John*
$R - E$	*posterior present* simple future: *I shall go*	*posterior future* (none)	*posterior past* conditional: *I would see John*
$E - R$	*anterior present* present perfect: *I have seen John*	*anterior future* future perfect: *I'll have seen John*	*anterior past* past perfect: *I had seen John*

Table 2.1: Reichenbach's representation of English tenses

tomorrow is R), lacks a posterior future, and the conditional can be used in many different ways besides denoting a posterior past.

Reichenbach's analysis offers a natural account of the difference between the English simple past and present perfect. The contrast in (8) and (9) shows that while the simple past in (8) can combine with time expressions denoting a past time, such as *last month* and *1957*, this is not the case of the present perfect, as the ungrammatical examples in (9) illustrate.[1] This is because these temporal expressions (*last month* and *1957*) arguably refer to R.

(8) a. *I visited the Parthenon last month.*

b. *I visited the Parthenon in 1957.*

(9) a. * *I have visited the Parthenon last month.*

b. * *I have visited the Parthenon in 1957.*

Reichenbach's theory has received some criticism, one of the reasons being that it allows more tenses than the ones usually found in natural languages.

Nevertheless, this sort of decomposition of tenses as involving more points in time than just the speech time and the event time has been enormously influential in the subsequent literature on tense and aspect. For instance, Comrie (1985) distinguishes

[1] We follow the common practice in the linguistics literature of presenting ungrammatical examples preceded by a star (*).

absolute tense (simple present, simple past and simple future) and *relative tense* (present perfect, past perfect and future perfect), and argues that only the latter need to be represented with a reference point. Prior (1967) suggests the use of two reference points R_1 and R_2 in order to account for examples like *I shall have been going to see John*, with the tense structure $S - R_2 - E - R_1$. Hornstein (1990) uses Reichenbach's system to explain the possible combinations of tenses with temporal adverbs in English. An adverb like *now* is represented with the semantics of the simultaneous present, *yesterday* is considered similar to the simultaneous past, and *tomorrow* is akin to the simultaneous future. He focuses on why combinations like *John leaves tomorrow* are possible (where the simple present tense form *leaves* gets a future interpretation), whereas combinations like ** John has left yesterday* are not.

Another line of research is concerned with the temporal flow of discourse. Lascarides & Asher (1993) present a formal account of how to determine discourse relations between propositions introduced in a text, and the relations between the events they describe. They seek to explain the different temporal orderings in narratives such as:

(10) a. Max stood up. John greeted him.

b. Max fell. John pushed him.

In (10a) the first sentence describes a situation that temporally precedes the situation described in the second sentence. In (10b) the temporal ordering is the opposite. The authors explain this difference through defeasible constraints ($\phi > \psi$ "ϕ normally entails ψ") and non-monotonic inference:

- Defeasible Modus Ponens
  $\phi > \psi, \phi \models \psi$
  E.g., birds normally fly, Tweety is a bird $\models$ Tweety flies

- Penguin Principle
  $\phi \to \psi, \phi > \neg\chi, \psi > \chi, \phi \models \neg\chi$
  E.g., penguins are birds, penguins normally don't fly, birds normally fly, Tweety is a penguin $\models$ Tweety doesn't fly

The sentence in (10a) illustrates the default interpretation of discourse according to which situations happen in the temporal order in which they are described in a discourse. This is defeasible, however, as (10b) exemplifies. Here, the reverse temporal ordering is imposed by a causation relation between *push* and *follow*. The piece of knowledge that causes precede effects is not defeasible.

Precisely determining the temporal relation holding between situations mentioned in consecutive sentences can, however, depend on more factors. For instance, tense seems to have anaphoric properties (Webber, 1988): just like pronouns pick up entities previously introduced in a discourse, verb tense can also refer to times previously mentioned. In (11), the second sentence picks up the event time of the first sentence (the two playing events happen at the same time).

(11) a. John played the piano.
b. Mary played the kazoo.

2.2.2 Aspect and Aspectual Type

Aspect and **aspectual type** are related to the way situations are described in natural language with respect to their internal structure (Binnick, 1991; Comrie, 1976; Moens, 1987; Smith, 1997; Vendler, 1957; Verkuyl, 1993).

Vendler (1957, 1967) notes that not all verbs behave identically as far as grammatical tense is concerned. For instance, *I am running* is a valid English sentence, but *I am knowing* is nonsense. This phenomenon is now known as **aspectual type**, aspectual class, situation type, Aktionsart, lexical aspect, Vendler class or Vendler/Dowty class.

Vendler introduced four aspectual classes: states, activities, accomplishments and achievements. In this text we will use the terminology of Dowty (1979), though, and talk about **states**, **processes** (Vendler's activities), **culminated processes** (Vendler's accomplishments), and **culminations** (Vendler's achievements).[1]

Examples of states are *to hate beer*, *to know the answer*, *to own a car*, *to stink*. *to be sick*. Examples of processes are *to work*, *to eat ice cream*, *to grow*, *to play the piano*. Among culminated processes we find *to paint a picture*, *to burn down*,

[1] In some of the literature, culminations are further divided into culminations, *stricto sensu*, and points, but we will ignore this distinction.

to deliver a sermon. Finally the class of culminations contains phrases such as *to explode*, *to win the game*, *to find the key.*

States and processes are **atelic** situations in that they do not make salient a specific instant in time. Culminated processes and culminations are **telic** situations: they have an intrinsic, instantaneous endpoint, called the culmination (e.g. in the case of *to paint a picture*, it is the moment when the picture is ready; in the case of *to explode*, it is the moment of the explosion). Culminated processes consist of a process followed by a culmination (e.g. *to paint a picture* is a process of painting a picture and a culmination of finishing it).

These classes are distinguished by several linguistic tests. One such test is their occurrence in the progressive: processes and culminated processes have no problem appearing in the progressive (*He is running*, *He is painting a picture*), whereas states and culminations often produce ungrammatical sentences (** He is knowing French*, ** He is recognizing his friend*). Another test is the preposition used in durational adverbials: the duration of processes is indicated by durational phrases headed by *for* (*John swam for two hours*), whereas *in* is used with culminated processes to indicate the duration of the process that precedes the culmination (*John painted a picture in two hours*).

Aspectual type is not a property of words, but rather of phrases. Different phrases with the same head verb can have different aspectual types. For instance *to paint a picture* is a culminated process (cf. *John painted a picture in two hours*), but *to paint pictures* is a process (cf. *John painted pictures for two hours*) (Garey, 1957; Krifka, 1992; Platzack, 1979; Verkuyl, 1972). Additionally, some phrases have an aspectual type different from the aspectual type of their composing elements: *to paint a picture* is a culminated process but *to paint a picture every day* is a process (cf. *John painted a picture every day for two years*). In this example, the phrase *every day* combines with a phrase that describes a culminated process to produce a larger phrase that describes a process. This is known as **aspect shift** (or *Aktionsart* shift).

A phenomenon related to aspect shift is **aspect coercion**: clashes of constraints on aspectual type often do not result in ungrammatical expressions but rather force a coercion of their aspectual type, with a noticeable shift in their meaning. For instance, *for* adverbials, as mentioned above, combine with processes. However, a sentence like *John painted a picture for two hours* is grammatical, but the culminated process *to paint a picture* is coerced into a process, with a change in meaning: the sentence no longer means that John finished the painting (the culmination is stripped as the result of the coercion).

Aspectual coercion provides an explanation for the progressive/imperfective paradox (Bach, 1986; Dowty, 1979), illustrated by the examples in (12) and (13).

(12) a. John was swimming.
b. John swam.

(13) a. John was painting a picture.
b. John painted a picture.

The paradox is that (12a) entails (12b), but (13a) does not entail (13b). This contrast is due to both (12a) and (12b) describing atelic situations, but (13b) contains a culminated process (the picture was finished), whereas (13a) is an atelic situation (the picture was not finished). The idea is that the progressive construction combines with processes, which are atelic. In (13a), the progressive construction coerces the culminated process of *painting a picture* into a (non-culminated) process.

The work of de Swart (1998b, 2000) analyzes aspectual coercion as the occurrence of implicit aspectual operators that are used only when clashes occur. Just like the progressive is an aspectual operator, namely a function from processes to states, there are other aspectual operators that are different in that they are silent. The sentence in (14), together with a schematic representation of the relative scope between the different temporal and aspectual elements involved and taken from de Swart (1998b), illustrates this idea of implicit aspectual operators. In this case the silent operator is represented with $\mathrm{C_{eh}}$, and it is a function from events (the author reserves this term to refer to telic situations; *John played the sonata* is a telic situation) to homogeneous (i.e. atelic) situations (as required by the *for* adverbial, as mentioned above).[1]

(14) John played the sonata for eight hours.
[PAST [FOR eight hours [$\mathrm{C_{eh}}$ [John play the sonata]]]]

[1] Atelic situations are called homogeneous because they exhibit the subinterval property: if they hold in some time interval t, they hold in every subinterval of t.

Moens & Steedman (1988) introduce the concept of "event *nucleus*": an event has a nucleus made of a preparatory process followed by a culmination followed by a consequent state. The examples in (15) mention an event of building a bridge:

(15) a. *When they built the 59^{th} bridge, they used the best materials.*

b. *When they built the 59^{th} bridge, they solved most of their traffic problems.*

The preparatory process is the process of actively building the bridge, the culmination is the point when bridge is finished, and the consequent state is the existence of that bridge.

In these examples, the *when* clause can refer to different parts of the nucleus of *building a bridge*. The *when* clause refers to the preparatory process in (15a), and to the consequent state in (15b).

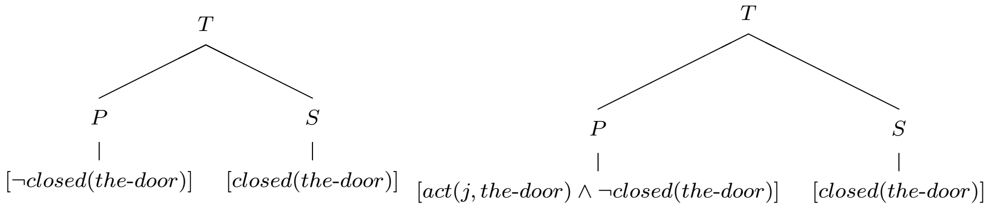
According to Moens & Steedman (1988), the components of this nucleus are optional, and their presence or absence is what determines aspectual type. Aspectual coercion can thus be viewed as adding or removing parts of the nucleus. The authors introduced an oft-cited diagram describing the possible transitions involved in aspectual type coercion, which we show in Figure 2.1.

Pustejovsky (1991) explains aspectual phenomena by viewing situations as structures composed of other situations. For instance, the situations described in (16) are analyzed as having an internal structure. More specifically and as depicted below in (17), each of the two sentences is viewed as describing a transition T between a first situation when the door is not closed (P) and a second situation when the door is closed (S).

(16) a. The door closed.

b. John closed the door.

(17)



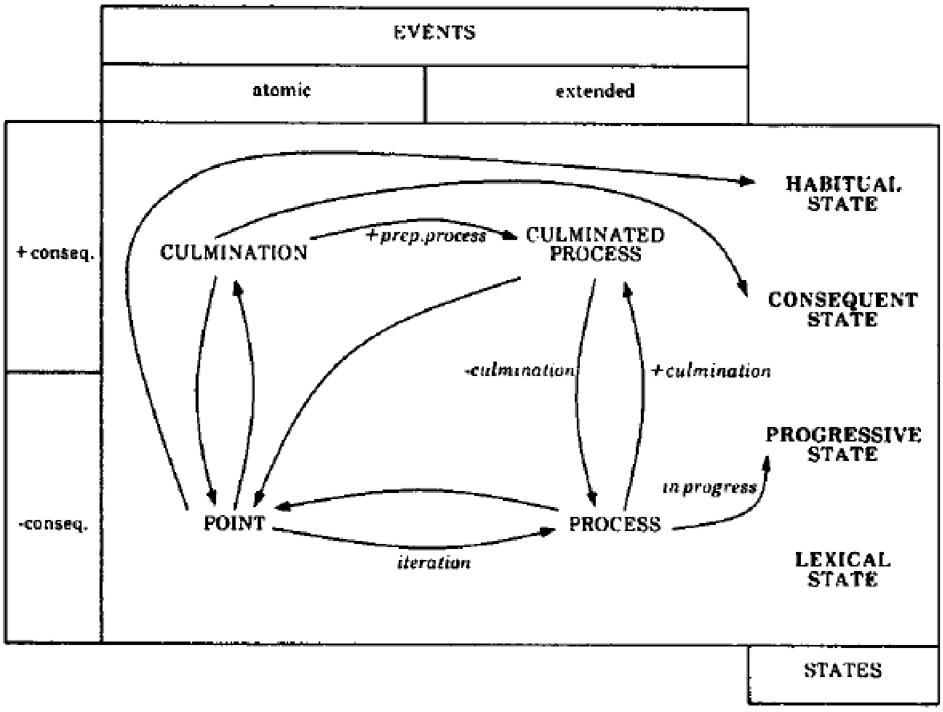


Figure 2.1: Possible kinds of aspectual type coercion according to Moens & Steedman (1988)

(16a) is a culmination and (16b) is a culminated process. For Pustejovsky (1991), the difference between culminations and culminated processes is that the P part of the latter also includes an $act(ivity)$ predicate (as seen in (17)) between the two participants of the situation and this activity causes the change of state (the transition from $[\neg closed(the\text{-}door)]$ to $[closed(the\text{-}door)]$). Such a representation captures the fact that some phrases can modify parts of the situations described in sentences. For instance, *almost* is ambiguous with culminated processes. A sentence like *John almost closed the door* can mean that John never started the process of closing it or that he did but he did not finish it. In the second interpretation *almost* scopes only over the S structure in the representation above.

Aspectual type has several consequences for the way in which the meaning of sentences can be computed, i.e. compositional semantics. Discourse Representation Theory (DRT; Kamp & Reyle (1993)) is one of the most influent current theories of compositional semantics. It assumes a representation of tense inspired by the work of Reichenbach (1947), describing tense with the help of several points in time. DRT features different modes of composing meaning representations in the presence of temporal location adverbials (e.g. *yesterday*, *last week*, *in 1974*, etc.), depending on the aspectual type of the verb. For states, it assumes that the time in which the state is true overlaps the time picked up by these expressions (cf. *John was ill yesterday*). In the case of non-stative situations, this relation is more specifically one of inclusion (cf. *John broke his ankle yesterday*).

The work of Móia (2000) is relevant to our work, because it is concerned with data from Portuguese. It studies this interdependence between the semantics of temporal location adverbials and aspectual type. More specifically, other factors are identified that also affect this relation. These factors include causality and quantification. For instance, the interaction between quantification and some kinds of temporal location adverbials can be seen in the example sentences in (18). (18a) is ungrammatical in Portuguese whereas (18b) is a possible sentence.

(18) a. * O Paulo comprou este apartamento desde 1980.
Paulo has bought this apartment since 1980.

b. O Paulo comprou três apartamentos desde 1980.
Paulo has bought three apartments since 1980.

2.2.3 Temporal Reasoning

Tense Logic, developed by Prior (1957, 1967, 1969), extends traditional logic with four modal operators:

- P "at some time in the past it was the case that ... "
- F "at some time in the future it will be the case that ... "
- H "in the past it was always the case that ... "
- G "in the future it will always be the case that ... "

Given a proposition ϕ, $P\phi \equiv \neg H \neg \phi$ and $F\phi \equiv \neg G \neg \phi$. Prior also posits several axioms, such as $G(\phi \rightarrow \psi) \rightarrow (G\phi \rightarrow G\psi)$ "if ϕ will always imply ψ, then if ϕ will always be the case, so will ψ".

Prior's system allows the arbitrary iteration of these operators, and originates many expressions that do not correspond to any tense found in natural languages: $FFFFFFF\phi$ "it will be the case that it will be the case that it will be the case that it will be the case that it will be the case that it will be the case that it will be the case that ϕ". For this reason it is usually considered inadequate to describe the tense system of natural languages.

Nevertheless, Priorean logic has been extended since its inception. For instance, Kamp (1968) adds the temporal binary operators S "since" and U "until": $S\phi\psi$ "ψ has been true since a time when ϕ was true" and $U\phi\psi$ "ψ will be true until a time when ϕ is true."

Davidson (1967) reifies situations by adding an event variable to each predicate that forms a situation located in time. For instance, *John saw Mary* gets a representation where the predicate for *see* has an extra argument that stands for the event itself:

(19) a. John saw Mary. $\exists e[see'(e, john', mary')]$

b. John saw Mary in Paris. $\exists e[see'(e, john', mary') \wedge in(e, paris')]$

This idea allows for a formal account of some inferences. In particular, (19a) following from (19b) can be accounted for by conjunction elimination.

Aspectual type, as presented above in Section 2.2.2, has consequences to truth conditions. States (such as *John was asleep*) are homogeneous in the sense that if a state holds over some interval t then it must hold over every subinterval of t: if *John was asleep between 2 and 5 p.m.* then it must be true that *John was asleep between 2 and 3 p.m.* Telic situations (such as *John woke up*) do not have this property: if it is true that *John woke up between 2 and 5 p.m.*, then it is not necessarily true that *John woke up between 2 and 3 p.m.* because he might have woken up only at 4 p.m.

For this reason, Allen (1984) represents the temporal location of states and dynamic situations differently, using a HOLDS predicate to describe the time at which a state holds and an OCCUR predicate to describe the time of dynamic situations:

- $\text{HOLDS}(asleep(j), (3pm, 5pm))$
- $\text{OCCUR}(wake\text{-}up(j), (3pm, 5pm))$

The homogeneity property of states is then captured by the definition of the HOLDS predicate:

- $\text{HOLDS}(p, T) \Leftrightarrow (\forall t.\text{IN}(t, T) \Rightarrow \text{HOLDS}(p, t))$

Here, IN stands for inclusion.

The work of Allen (1983, 1984) also identifies "a basic set of mutually exclusive primitive relations that can hold between temporal intervals", and the logic he develops represents each one by a predicate. These relations are:

- $\text{DURING}(X, Y)$: time interval X is fully contained within Y;
- $\text{STARTS}(X, Y)$: time interval X shares the same beginning as Y, but ends before Y ends;
- $\text{FINISHES}(X, Y)$: time interval X shares the same end as Y, but begins after Y begins;
- $\text{BEFORE}(X, Y)$: time interval X is before interval Y, and they do not overlap in any way;

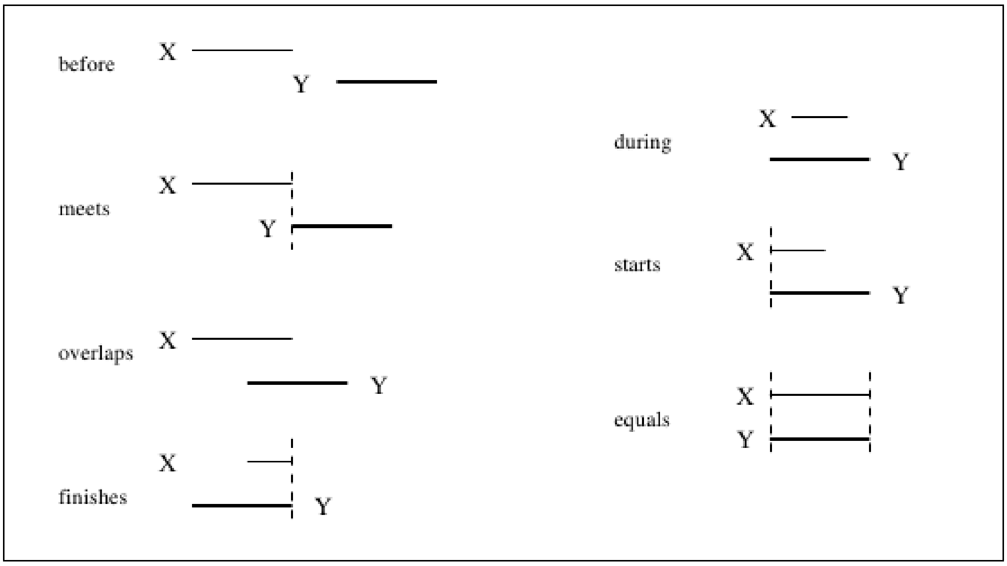


Figure 2.2: The temporal relations of Allen (1983, 1984)

- OVERLAP(X, Y): interval X starts before Y, and they overlap;

- MEETS (X, Y): interval X is before interval Y, but there is no interval between them, i.e., X ends where Y starts;

- EQUALS(X, Y): X and Y are the same interval.

Note that his OVERLAP predicate is not very intuitive (it is not symmetric). Figure 2.2, taken from Denis & Muller (2010), shows these relations in a graphical fashion.

Only EQUALS is symmetric. There is an additional relation type for the inverse of every other relation, leading to a total of 13 types of temporal relations. The additional relations are: AFTER (the inverse of BEFORE), CONTAINS (the inverse of DURING), OVERLAPPED-BY (the inverse of OVERLAPS), MET-BY (the inverse of MEETS), STARTED-BY (the inverse of STARTS) and FINISHED-BY (the inverse of FINISHES).

There is also a set of axioms that define the behavior of these predicates. For the sake of completeness, the IN predicate used above gets the definition:

- IN(X, Y) $\Leftrightarrow$ (DURING(X, Y) $\vee$ STARTS(X, Y) $\vee$ FINISHES(X, Y))

The work of Bruce (1972) represents very early work on computational temporal processing. The author uses seven possible types of temporal relation: *before* (the disjunction of Allen's BEFORE and MEETS), *after* (the disjunction of Allen's AFTER and IS-MET), *same-time* (Allen's EQUALS), *during* (the disjunction of Allen's STARTS, DURING, and FINISHES), *overlaps* (Allen's OVERLAPS), *contains* (the disjunction of Allen's IS-STARTED, CONTAINS and IS-FINISHED) and *overlapped* (Allen's OVERLAPPED-BY).

Further work on the appropriate temporal ontology and set of temporal relations for analyzing temporal phenomena in natural language include that of Galton (1990). Allen's basic relations are all between temporal intervals. Instants play a secondary role in his theory. Galton (1990) claims that continuous change (e.g. a ball falling onto a table) cannot be adequately handled by Allen's theory because of the lack of instants and proposes a series of revisions to it that diversify the temporal ontology so that it includes both intervals and instants.

2.3 Early Computational Approaches

Computational linguistics has been interested in developing systems that incorporate much of what has been presented in this chapter for a few decades now.

Bruce (1972) presents a formal model of the structure underlying temporal references in natural language. It uses the set of temporal relations described above in Section 2.2.3. This model is implemented in a dialog system, *Chronos*. This system can answer questions that require temporal reasoning, provided the facts necessary to answer those questions have been previously entered by the user. An example dialog between a user and *Chronos* is in Figure 2.3.

Passonneau (1988) develops a system for English that associates each clause with a meaning representation that incorporates information about tense and aspect. The representation associates one or more time intervals to each situation described in the input text. The system tries to identify the aspectual type of each situation, and the intervals it uses in the meaning representations are described by properties that reflect aspectual distinctions. This idea is inspired in the previous work of Dowty (1979) and Allen (1984), among others.

```
(THE AMERICAN WAR FOR INDEPENDENCE BEGAN IN 1775)
(INFORMATION ACCEPTED)

(THE AMERICAN WAR FOR INDEPENDENCE ENDED IN 1781)
(INFORMATION ACCEPTED)

(DOES THE AMERICAN WAR FOR INDEPENDENCE COINCIDE WITH
THE TIME FROM 1775 TO 1781 *)
YES

(THE ARTICLES OF CONFEDERATION PERIOD WAS FROM 1777 TO 1789)
(INFORMATION ACCEPTED)

(WHEN DID THE ARTICLES OF CONFEDERATION PERIOD BEGIN *)
(IN 1777)

(WHEN DID THE ARTICLES OF CONFEDERATION PERIOD END *)
(IN 1789)

(HOW LONG WAS THE ARTICLES OF CONFEDERATION PERIOD *)
(12 YEARS)

(HOW LONG WAS THE AMERICAN WAR FOR INDEPENDENCE *)
(6 YEARS)

(WAS THE AMERICAN WAR FOR INDEPENDENCE BEFORE 1800 *)
YES

(HOW LONG WAS IT BETWEEN THE AMERICAN WAR FOR
INDEPENDENCE AND THE WAR OF 1812 *)
(31 YEARS)
```

Figure 2.3: Dialog between a user and Chronos (Bruce, 1972)

Van Eynde (1994) presents a typed feature structure (Carpenter, 1992) specification of tense and aspect. This work uses DRT semantics, mentioned above in Section 2.2.2, and the grammatical framework of Head-driven Phrase Structure Grammar (Pollard & Sag, 1987, 1994; Sag *et al.*, 2003), which is well-suited for computational implementations. The work of Van Eynde (1994) was developed as part of an effort to implement computational grammars for various languages. It constitutes an early effort to incorporate temporal semantics inspired by Reichenbach (1947) as well as aspectual constraints in the meaning representations produced by computational grammars.

2.4 TERN 2004

The first work on annotating temporal expressions was in MUC-6 (MUC-6, 1995) and MUC-7 (MUC-7, 1998). These Message Understanding Conferences were concerned with several tasks that are relevant to information extraction systems. These evaluations defined the TIMEX tags used to annotate **time expressions**.

A **temporal expression** (or **time expression**, or **timex**) is a natural language expression that refers to a date, a time or, more generally, any span of time. Some examples are expressions like “July 17, 1999”, “12:00”, “the summer of 69”, “yesterday”, “last week”, “the next millennium”. Expressions that denote durations, such as “one hour”, “two weeks”, are also considered temporal expressions.

The first evaluation campaign of temporal information systems was the 2004 Temporal Expression Recognition and Normalization evaluation —TERN 2004 (Ferro *et al.*, 2004)—,[1] part of the Automatic Content Extraction (ACE) program (Strassel *et al.*, 2008).

TERN 2004 was specifically concerned with the automated identification and normalization of temporal expressions in English and Chinese text. The goal was to mark up raw text with XML tags around temporal expressions (**recognition**), and determine the value of attributes for these XML elements that encode a standardized representation of times and dates (**normalization**). Figure 2.4 shows an example of the data used in the TERN evaluation, with annotations. The time expression

[1] http://timex2.mitre.org

<TIMEX2 val="1998-W49" mod="" set="" non_specific="" anchor_val="" anchor_dir="" comment=""> *Two weeks ago* </TIMEX2>*, Renada Daniel Patterson's only kidney, donated by her father, began to fail, prompting a swirling debate when he offered to give her his remaining one. But on* <TIMEX2 val="1998-12-18" mod="" set="" non_specific="" anchor_val="" anchor_dir="" comment=""> *Friday* </TIMEX2>*, even as the medical ethics committee at the University of California-San Francisco Medical Center was discussing whether to allow a transplant that would make David Patterson a dialysis patient for the rest of his life, his* <TIMEX2 val="P16Y" mod="" set="" non_specific="" anchor_val="1998" anchor_dir="ENDING" comment=""> *16-year-old* </TIMEX2> *daughter and her mother announced that the debate was moot: Renada had been healed by God.*

Figure 2.4: Sample of the data annotated for TERN 2004

Friday is annotated with 1998-12-18 as the value of its val attribute. This indicates the particular date that it refers to.

The normalization of temporal expressions had begun with DARPA's Translingual Information Detection, Extraction, and Summarization (TIDES) research program (Ferro *et al.*, 2001, 2004), where the TIMEX2 specification was developed. The TIMEX2 tags were also employed in TERN 2004.

The challenge behind the recognition and normalization of time expressions is that these expressions can take a variety of linguistic forms, some more vague than others. Additionally, some expressions may be anchored in other expressions, and there are several ways in which a temporal **anchor** is chosen for a timex. Some timexes, such as "today", "three years ago", and "next week", are deictic and anchored to the time of speech (which when processing a document can be identified by the **document's creation time**). Others, such as "two months earlier" and "the next week", are anaphoric and anchored to a salient time in discourse, typically expressed by a previous timex. They may be partially so—e.g. in the expression "May 3" the anchor only provides the year. A timex may also contain its own anchor: "three days after May 3", whose anchor is the embedded timex "May 3".

The best performing system in TERN 2004 was a rule-based system (Negri & Marseglia, 2004). It employed a part-of-speech tagger and then used regular expressions to find temporal expressions. Normalization was also performed by hand-made rules. It achieved a 92% F-measure for recognition but lower values for normalization (F-measures between 68% and 87% for the different attributes).

TERN 2004 also made available data with annotations pertaining to temporal expressions. The data of the TERN 2004 evaluation consist of a training set and a test set. The training corpus is made up of 511 documents of newswire and broadcast news transcripts, with 5326 temporal expressions. The test corpus contains 192 similar documents, with 1828 temporal expressions.

More recently, Ahn *et al.* (2007) used a set of machine learning classifiers in conjunction with a syntactic parser and achieved results comparable to those of TERN 2004 with the same data set.

2.5 TimeML and TimeBank

An XML-based annotation format for representing temporal information called **TimeML** (Pustejovsky *et al.*, 2003a) has matured and emerged in recent years as the *de facto* standard for the temporal annotation of natural language text.

Like the annotation scheme used for TERN 2004, TimeML allows for the annotation of temporal expressions. The format used in TimeML to annotate the normalized value of time expressions (i.e. to describe resolved dates and times) follows an ISO standard (ISO 8601). Additionally, TimeML defines annotations for other elements that are relevant to time. Among others, terms that denote events[1] are annotated, and so are temporal relations holding between various elements.

Several TimeML corpora have been created over the years. TimeBank (Pustejovsky *et al.*, 2003b) is an English corpus that is manually annotated with TimeML. There is an extension of the TimeBank, called the OTC corpus, obtained by combining it with the AQUAINT corpus. The TimeBank corpus consists of 186 newswire articles (with around 65,000 words). AQUAINT features 73 documents (with around 40,000 words). Both corpora are annotated with TimeML, and available at http://www.timeml.org/site/timebank/timebank.html.

[1] Note that the linguistics literature employs the term *event* to refer to situations that are not states. In the literature on temporal processing, *events* include both stative and non-stative situations. Therefore, TimeML seeks to annotate all situations that are temporally bound (i.e. it still excludes some stative descriptions when they correspond to generic statements such as *Lions are mammals*).

Setzer (2001) and Setzer & Gaizauskas (2000a,b, 2001) describe an early effort to manually annotate temporal expressions, events and temporal relations. This work served as the basis for TimeML.

The inventory of temporal relations covered in TimeML was inspired in the inventory of relations of Allen (1984), mentioned above in Section 2.2.3. This inventory also contains thirteen relations but a few are different (and several other relations have different names). Nevertheless, most work focusing on temporal relation classification has used an even smaller set of temporal relations, namely the reduced set of TempEval (see Section 2.6). This is partly in order to make the problem of automated temporal relation classification easier to be handled by systems trained on data annotated with TimeML.

The TimeBank was manually annotated with the help of software, that among other things, checks the annotations for consistency and automatically computes the temporal closure of the manually annotated temporal relations (Verhagen, 2005; Verhagen *et al.*, 2005). The inferred relations can then be added to the annotations. This is inspired by Setzer (2001) and Katz & Arosio (2001), who both add a closure component to a temporal annotation environment. The reasoning components is based on Allen's algebra. Because temporal annotation is a hard task for humans, research has continued on how to aid human annotation of temporal phenomena (e.g. Pustejovsky & Stubbs (2011); Xue & Zhou (2010); Zhou & Xue (2011)).

2.6 TempEval (TempEval 2007)

A slightly simplified and slightly altered version of TimeBank was used in a track of the evaluation competition Semeval-1/Senseval-4 called TempEval (Verhagen *et al.*, 2007). This was the first time that Senseval included a track for temporal information processing.

TempEval presented three challenges, related to **events** and temporal entities. A description of these tasks can be found on the TempEval web page (http://www.timeml.org/tempeval). We repeat it here:

- **Task A**

  For a restricted set of event terms, identify temporal relations between events and all time expressions appearing in the same sentence.

- **Task B**
  For a restricted set of event terms, identify temporal relations between events and the Document Creation Time (DCT).

- **Task C**
  Identify the temporal relations between contiguous pairs of matrix verbs.

The *restricted set of event terms* mentioned here is the set of event terms occurring at least 20 times in TimeBank.

We will refer to these three tasks throughout the text of this dissertation. To make reading easier, we will use more descriptive names, respectively: Task A Event-Timex (since it is about temporal relations between events and dates or times given by time expressions, or timexes), Task B Event-DocTime (as it is about temporal relations between events and the document creation time) and Task C Event-Event (about temporal relations between two events). We will still sometimes use the shorter names, e.g. in some tables.

Figure 2.5 shows a sample of the annotated data used in TempEval. The value of the relType attribute of the TLINK elements is what the systems competing in these tasks had to determine, all other information was given, including the relevant event terms (inside the EVENT tags) and temporal expressions (tagged with TIMEX3) and the Document Creation Time (the TIMEX3 with the value CREATION_TIME for the attribute functionInDocument).

For all these tasks, the possible values of the attribute relType are AFTER, BEFORE, OVERLAP, OVERLAP-OR-AFTER (the disjunction of TempEval's OVERLAP and AFTER), BEFORE-OR-OVERLAP (the disjunction of TempEval's BEFORE and OVERLAP) and VAGUE (the disjunction of TempEval's BEFORE, OVERLAP and AFTER). The last three values are used in those cases where the human annotators could not decide for a more specific value. The correspondence between these temporal relations and those of Allen (1983, 1984) and Bruce (1972), mentioned above in Section 2.2.3, is presented in Figure 2.2.

In this competition there were symbolic systems, resorting to hand-made rules, supervised machine learning approaches and hybrid systems. Some systems based their classification on relatively shallow information, others used more sophisticated natural language processing.

```
<?xml version="1.0"?>
<TempEval>
ABC<TIMEX3 tid="t52" type="DATE" value="1998-01-14" temporalFunction="false"
functionInDocument="CREATION_TIME">19980114</TIMEX3>.1830.0611
NEWS STORY
<s>In Washington <TIMEX3 tid="t53" type="DATE" value="1998-01-14"
temporalFunction="true" functionInDocument="NONE"
anchorTimeID="t52">today</TIMEX3>, the Federal Aviation Administration
<EVENT eid="e1" class="OCCURRENCE" stem="release" aspect="NONE" tense="PAST"
polarity="POS" pos="VERB">released</EVENT> air traffic control tapes from
<TIMEX3 tid="t54" type="TIME" value="1998-XX-XXTNI" temporalFunction="true"
functionInDocument="NONE" anchorTimeID="t52">the night</TIMEX3> the TWA
Flight eight hundred <EVENT eid="e2" class="OCCURRENCE" stem="go"
aspect="NONE" tense="PAST" polarity="POS" pos="VERB">went</EVENT>
down.</s>
...
<TLINK lid="l1" relType="BEFORE" eventID="e2" relatedToTime="t53" task="A"/>
<TLINK lid="l2" relType="OVERLAP" eventID="e2" relatedToTime="t54" task="A"/>
<TLINK lid="l4" relType="BEFORE" eventID="e2" relatedToTime="t52" task="B"/>
...
</TempEval>
```

Figure 2.5: Sample of the data annotated for TempEval, corresponding to the fragment:
ABC.19980114.1830.0611
NEWS STORY
In Washington today, the Federal Aviation Administration released air traffic control tapes from the night the TWA Flight eight hundred went down.

Allen (1983, 1984)	Bruce (1972)	TempEval
BEFORE MEETS	before	BEFORE
OVERLAPS	overlaps	OVERLAP
STARTS DURING FINISHES	during	
EQUALS	same-time	
OVERLAPPED-BY	overlapped	
IS-STARTED CONTAINS IS-FINISHED	contains	
IS-MET AFTER	after	AFTER

Table 2.2: Correspondence between the temporal relation inventories of Allen (1983, 1984), Bruce (1972), and TempEval.

Table 2.3 presents the relevant results, taken from Verhagen *et al.* (2009). It presents the *strict scores*. *Relaxed scores* were also used, giving partial credit to mismatches involving the disjunctive classes VAGUE, BEFORE-OR-OVERLAP, OVERLAP-OR-AFTER.

All systems achieved relatively poor results, and most performed quite similarly within the same task. Nevertheless, most systems performed considerably above the majority baseline, specially for Task B Event-DocTime. The best performing system was WVALI (Puşcaşu, 2007). It employed hand-made rules and heuristics operating on the output of a syntactic parser, more specifically a dependency parser, making use of a module with world knowledge axioms.

2.7 TempEval-2 (TempEval 2010)

TempEval-2[1] (Pustejovsky & Verhagen, 2009; Verhagen *et al.*, 2010) featured six tasks. The first two were concerned with the identification and normalization of

[1] http://www.timeml.org/tempeval2

	Score (%)		
System	Task A	Task B	Task C
CU-TMP	61	75	54
LCC-TE	58	73	**55**
NAIST	61	75	49
USFD	59	73	54
WVALI	**62**	**80**	54
XRCE-T	34	66	42
Average	56	74	51
Baseline	57	56	47

Table 2.3: Results of TempEval (accuracy). The baseline is to always assign the majority class. Highest scores in boldface.

temporal expressions (task A) and event terms (task B). Tasks C, D, and E of TempEval-2 are similar to the Task A Event-Timex, Task B Event-DocTime and Task C Event-Event of the first TempEval. The last task of TempEval-2 (task F) consists in determining the temporal relations holding between two events expressed by terms that occur in the same sentence and are syntactically related.

TempEval-2 covered other languages besides English. Although data sets were prepared for several languages (English, Italian, Spanish, Chinese, Korean and French), only English and Spanish were addressed by participants attending the competition.

Tables 2.4 and 2.5 show the results obtained in TempEval-2 for English and Spanish respectively, for tasks C through F.

The comparison between the results of TempEval-2 and those of the first TempEval competition is somewhat disappointing. The bests results of TempEval-2 are slightly better than the best ones in TempEval, but the baselines are also higher in TempEval-2 for tasks D and E. The other task that they have in common (task C of TempEval-2 and Task A Event-Timex of TempEval) was arguably easier in TempEval-2. In TempEval-2 this task was only concerned with temporal relations between an event and a time given by words and phrases that are syntactically related in the sentence where they occur (e.g. the time expression is part of a modifier of the event term). In the first TempEval, this task covered more cases, namely cases where the two elements were not directly related in the syntactic structure of the sentence.

	Score (%)			
System	Task C	Task D	Task E	Task F
JU_CSE	63	80	56	56
NCSU-indi	63	68	48	**66**
NCSU-joint	62	21	51	25
TIPSem	55	**82**	55	59
TIPSem-B	55	81	55	60
TRIOS	**65**	79	56	60
TRIPS	63	76	**58**	59
USDF2	63	-	45	-
Average	61	70	53	55
Baseline	55	59	49	30

Table 2.4: Results of TempEval-2 (accuracy) for English. The baseline is to always assign the majority class. Highest scores in boldface.

	Score (%)	
System	Task C	Task D
TIPSem	**81**	**59**
TIPSem-B	**81**	**59**
Baseline	81	46

Table 2.5: Results of TempEval-2 for Spanish (accuracy). The baseline is to assign the majority class always. Highest scores in boldface.

The best systems of TempEval-2 employed various approaches. The TRIPS and TRIOS systems (UzZaman & Allen, 2010) used a combination of parsing and machine learning methods such as Conditional Random Fields (CRF; Lafferty *et al.* (2001)) and Markov Logic Networks (Richardson & Domingos, 2006). TIPSem (Llorens *et al.*, 2010a) also used CRFs trained using several kinds of features, including features extracted from the output of a syntactic parser, namely that of Charniak & Johnson (2005) for English. Like UzZaman & Allen (2010), the NCSU systems (Ha *et al.*, 2010) employed Markov Logic using features taken from different natural language processing tools. Ha *et al.* (2010) gave a bigger emphasis to features that capture lexical relations between the event terms involved (such as similarity relations between *producing* and *creating* events, antonymy relations between the terms *open* and *close*, etc.).

Boguraev & Ando (2006) is a reflection on some limitations of the TimeBank, in particular its short size and some inconsistent annotations. This is of note, since the data used in the TempEval challenges are largely based on the TimeBank.

Some of their remarks are important. For instance, TimeBank is one of the outcomes of the TERQAS effort (Temporal and Event Recognition for QA Systems), from which the TimeML annotation guidelines emerged. TimeBank was not originally intended as a resource to support the training and evaluation of computational systems, but rather the result of an annotation exercise intended to develop TimeML and test the TimeML annotation guidelines. As the authors write, “it was never the subject of rigorous considerations of scope, coverage, size, consistency, double-annotation, and inter-annotator agreement”. Therefore, some level of noise in the data is expected.

Additionally, there is a serious problem of data sparseness. This is clear in the case of the disjunctive temporal relation types also in the data of TempEval: the classes VAGUE, OVERLAP-OR-AFTER and BEFORE-OR-OVERLAP are never assigned by any machine learned classifier induced with these data. According to the authors, the problem is not exclusive of this TimeML attribute.

2.8 Other Corpora and Competitions

In recent years, a number of corpora with temporal annotations have been developed for several languages, inspired by the English TimeBank and the TimeML specification. There have been efforts to build TimeBanks for Chinese (Cheng *et al.*, 2008), French (Bittar *et al.*, 2011), Italian (Caselli *et al.*, 2011), Korean (Im *et al.*, 2009), Romanian (Forăscu & Tufiş, 2012).

For Portuguese, the second HAREM[1] was an evaluation competition for named entity recognition that included a track for temporal expressions. The temporal annotation of the data "was largely inspired in the recent work on TimeML" (Hagège *et al.*, 2008a). The best performing system in this track was rule-based (Hagège *et al.*, 2008b). The best results are around a 75% F-measure for both recognition and normalization of temporal expressions.

The authors of the WikiWars corpus (Mazur & Dale, 2010) acknowledge that the previously deployed corpora with annotations for time expressions (the TERN 2004 corpus and the TimeBank) consist of small documents with news stories, and "this impacts on the number, range and variety of temporal expressions they contain." They further add that existing research on the interpretation of temporal expressions (Ahn *et al.*, 2005; Baldwin, 2002; Mazur & Dale, 2008) suggests that "many temporal expressions in documents (. . .) can be interpreted fairly simply as being relative to a reference date that is typically the document creation date", but "this phenomenon does not carry over to longer, more narrative-style documents that describe extended sequences of events" (recall the discussion about anchors in Section 2.4). The authors create a corpus with annotations for time expressions that is intended to address this shortfall. WikiWars consists of 22 documents from the English Wikipedia that describe the historical course of wars.

2.9 Further Approaches

The development of TimeML allowed the creation of annotated natural language data where the focus is on temporal relations rather than temporal expressions. However, there had been previous efforts to compute temporal relations from text.

[1] http://www.linguateca.pt/HAREM

Filatova & Hovy (2001) develop a system that assigns a calendar date and time to each clause in an input news story, even when no explicit information about time is present in that clause (apart from verb tense). Schilder (1997) and Schilder & Habel (2001) try to relate verbs and some nouns to times when there is a syntactic relation between the event-denoting term and the time-denoting expression.

TimeML, the TimeBank and the two TempEval challenges have been very influential in the area. A lot of recent work has used the TimeBank and the data sets made available in the two TempEval challenges.

Mani *et al.* (2006) use machine learning methods to learn classifiers of temporal relations from the OTC corpus, annotated with TimeML (see Section 2.5). The distinctive idea of this work is that they use automated reasoning to oversample the data. Since each train or test instance represents a temporal relation, there is the possibility of increasing the training data by computing the temporal closure of the given relations. Even though this is an interesting idea, the authors recognized in subsequent work that there were methodological problems in this work which invalidate the results (Mani *et al.*, 2007).

Verhagen & Pustejovsky (2008) present a system that automatically annotates raw text with TimeML, including annotations for events, time expressions and temporal relations.

Denis & Muller (2010) compare the TimeML array of temporal relation types (before, overlap, etc.; see Section 2.6) with the inventory of temporal relations of Allen (1984) and Bruce (1972), both described in Section 2.2.3. These three algebras encode temporal relations at different levels of granularity and have different inferential properties. Through various experiments on the TimeBank/AQUAINT corpus, they conclude that "although the TempEval relation set leads to the best classification accuracy performance, it is too vague to be used for enforcing consistency", the other two sets of relations being harder to learn, but more useful for the purpose of ensuring global consistency.

Pan *et al.* (2006, 2011) annotate events for estimated bounds on their duration, and show that machine learning techniques, applied to this data, yield coarse-grained event duration information. Gusev *et al.* (2011) also infer the duration of events. In addition to the supervised methods experimented with by Pan *et al.* (2006), they also try an unsupervised approach, namely using web queries. Chambers *et al.*

(2007) trained machine learning classifiers on the TimeBank, namely Naive Bayes classifiers. They were concerned with temporal relations between pairs of events, that could be in the same sentence or not. So their system's goal intersects Task C Event-Event of the first TempEval and Task E of TempEval-2 (relations between events in different sentences). Their algorithm operates on two stages. In the first stage, they try to learn some properties of the events in the temporal relation, such as tense, grammatical aspect and aspectual class. Here they use some morphossyntactic features as well as features based on information provided by WordNet (Fellbaum, 1998), a lexical database encoding word senses and relations between them. In the second stage, they classify the temporal relation between those events. They use as classifier features the information obtained in the first stage, as well as other kinds of features based on the syntactic structure of the sentences where the events are mentioned. Llorens *et al.* (2010b), similarly to Llorens *et al.* (2010a), explore the contribution of semantic role labeling to temporal information processing.

Since the advent of TimeBank and the TempEval competitions, machine learning methods have become dominant in addressing the problem of extracting the temporal ordering of what is described in a text. One major limitation of machine learning methods is that they are typically used to classify temporal relations in isolation, and therefore it is not guaranteed that the resulting ordering is globally consistent. Yoshikawa *et al.* (2009) and Ling & Weld (2010) overcome this limitation using Markov logic networks, which learn probabilities attached to first-order formulas. Some of the participants of the second TempEval used a similar approach (Ha *et al.*, 2010; UzZaman & Allen, 2010). Denis & Muller (2011) cast the problem of learning temporal orderings from texts as a constraint optimization problem. They search for a solution using Integer Linear Programming (ILP), similarly to Bramsen *et al.* (2006), mentioned below, and Chambers & Jurafsky (2008a). Because ILP is costly (it is NP-hard), the latter two only consider *before* and *after* relations. Denis & Muller (2011) manage to use all kinds of temporal relations by encoding the original temporal relations between time intervals as temporal relations between instants (the endpoints of those intervals), which reduces the number of variables and constraints involved. Lee (2010) similarly cast the problem of classifying a relation between two time intervals as the problem of finding four relations between four instants (the endpoints of the two original time intervals). His issue is that he wants to use the original inventory of temporal relations in the TimeBank (as many as Allen's) instead of the reduced set of relations in the data sets of the two TempEval challenges. Each of these thirteen relations between time intervals can be described by four temporal relations between their endpoints, drawing from an inventory of only three types of relation between instants: *before*, *equals* and *after*.

The logical properties of temporal relations indeed make temporal information processing stand out from many of the other natural language processing tasks. UzZaman & Allen (2011) propose a new way to evaluate temporal information processing systems. Instead of the usual precision and recall metrics used in the two TempEval competitions, they argue that it is better to compute the temporal closure of the reference annotations and confront the result with a system's output. This is because a system may identify temporal relations that are not part of the reference annotations but nevertheless are logical consequences of the ones that are in fact annotated.

Mirroshandel *et al.* (2011) apply active learning to the problem of temporal relation classification. Traditional approaches rely on large amounts of existing annotated data. With active learning, the learner has control over choosing the instances that will constitute the training set. A typical active learning algorithm begins with a small number of annotated data, and selects one or more informative instances from a large set of unlabeled instances. Active learning strategies aim to efficiently select the most informative samples for labeling.

Some recent work concerned with time has however used ad-hoc data sets. Li *et al.* (2005) try to recognize temporal relations between events mentioned in Chinese text, making use of hand-coded rules. Bramsen *et al.* (2006), mentioned above, order *temporal segments*. A temporal segment is composed of one or more sentences that occur contiguously in a text. They use Integer Linear Programming to compute the best solution based on hand coded constraints, such as the transitivity of temporal precedence. The resulting temporal graphs are guaranteed to obey specific well-formedness conditions (e.g. no cycles). Chambers & Jurafsky (2008b) induce *narrative event chains* from raw newswire text. A narrative event chain is a partially ordered set of events related by a common protagonist. Therefore, they need to order events mentioned in text. To this end they employ an algorithm inspired in that of Chambers *et al.* (2007), mentioned above, but employing support vector machines instead of Bayesian inference.

Lapata & Lascarides (2006) report on a method that bypasses the need for manual annotation and allows for the temporal ordering of events described in the same sentence. They train a model on sentences where explicit markers of temporal relations, such as the words *before* and *after* occur, which is able to generalize to other sentences where the temporal relation holding between the various situations being described is not marked explicitly.

In recent years, the topic of temporal expression recognition and normalization has not been abandoned, however. As mentioned above in Section 2.8, WikiWars is a recent corpus where time expressions are annotated. Other recent work on this topic is Han & Lavie (2004); Zhao *et al.* (2010), etc. Kolomiyets *et al.* (2011) investigate the portability of time expression recognition to non-newswire domains. Their idea is to generate additional training examples by substituting temporal expression words with potential synonyms, taken from the WordNetand other resources.

Linguistics has had very little to say about time expressions. Most of the linguistics literature about temporal phenomena focuses instead on tense and aspect and related matters. Similarly, linguistic frameworks with heavy computational use, such as Head-driven Phrase Structure Grammar (Pollard & Sag, 1987, 1994; Sag *et al.*, 2003) and Lexical Functional Grammar (Kaplan & Bresnan, 1982), have largely ignored time expressions. This is understandable due to the fact that these frameworks were developed in order to describe the syntax of natural languages and are far more powerful than what is needed to process time expressions. Although time expressions show some recursion (cf. *the day after the day after tomorrow*), simple formalisms are able to describe the vast majority of time expressions occurring in every day text. Indeed, the state of the art in the recognition and normalization of time expressions employs regular expressions (Negri & Marseglia, 2004), which are less powerful than the commonly used linguistic formalisms, based on context-free grammars. Dale & Mazur (2006), however, work on temporal expressions using Head-driven Phrase Structure Grammar (Pollard & Sag, 1987, 1994; Sag *et al.*, 2003), a linguistic framework with a long tradition in computational applications. The authors develop a feature structure representation of temporal expressions, distinguishing *points* (dates and times) from *durations*. This structure uses one feature for each field in a normalized timex. For instance, the point 2006-05-13T15:00:00Z (where Z indicates the GMT zone) gets the feature structure representation:

$$\begin{bmatrix} \textit{point} & \\ \text{TIMEANDDATE} & \begin{bmatrix} \text{TIME} & \begin{bmatrix} \text{HOUR} & \mathit{15} \\ \text{MINS} & \mathit{00} \\ \text{SECS} & \mathit{00} \end{bmatrix} \\ \text{DATE} & \begin{bmatrix} \text{DAY} & \begin{bmatrix} \text{DAYNAME} & D4 \\ \text{DAYNUM} & \mathit{13} \end{bmatrix} \\ \text{MONTH} & \mathit{5} \\ \text{YEAR} & \mathit{2006} \end{bmatrix} \end{bmatrix} \\ \text{ZONE} & z \end{bmatrix}$$

The authors distinguish between the *local semantics* and the *in-document semantics* of temporal expressions. The local semantics of a timex is a representation that includes only what is expressed in the temporal expression, completely ignoring the surrounding context. Its in-document semantics is its normalized value. To determine it, it may be necessary to look at other parts of the document in which it occurs, and some reasoning may also be required.

Therefore, the local semantics of a timex is essentially an underspecified version of its global semantics. For instance, the local semantics of an expression like *May 13* is the partially specified structured:

$$\begin{bmatrix} \textit{point} & \\ \text{TIMEANDDATE} & \begin{bmatrix} \text{DATE} & \begin{bmatrix} \text{DAY} & \begin{bmatrix} \text{DAYNUM} & \mathit{13} \end{bmatrix} \\ \text{MONTH} & \mathit{05} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

They assume a granularity ordering over what they call the defining attributes in a temporal representation:

$$\text{year} > \text{month} > \text{daynum} > \text{hour} > \text{minute} > \text{second}$$

Timex normalization (i.e. deriving its *in-document semantics*) is then a matter of ensuring that there is a value for every defining attribute that is of greater granularity than the smallest granularity attribute present in a partially specified representation.

In the case of this example, this granularity rule says that it is necessary to determine the value for YEAR, but not for HOUR, MINS or SECS. The granularity rule ensures that the expanded feature structure is structurally identical to an ISO 8601 expression.

As a final note, almost all the work in the area assumes that the creation time of documents is known, as it is often trivial to determine. Chambers (2012) points out that it may not be the case for many documents found on the Web. As such, he proposes a way to infer it based on the temporal expressions found in the document itself: a phrase such as “since 1999” is a strong indication that the document is more recent than 1999. This is a relatively new task in the natural language processing community: automatic document dating.

2.10 Summary

In this chapter, we briefly described some of the most important work in the field of temporal information processing.

We started by introducing some fundamental concepts and views about the way time is mentioned in natural language. Reichenbach (1947) describes the various verb tenses of English by considering three salient times—the speech time (or utterance time), the event time and the reference time—and temporal relations between the speech time and the reference time and between the reference time and the event time. Vendler (1957) and Dowty (1979) work on aspectual type: situations can have different temporal structure: some (like *John was ill yesterday*) are homogeneous, holding in every subinterval of the interval in which they are reported to be true; others have a natural endpoint (as in *John ate a whole cake yesterday*), etc. The work of Prior (1957, 1967, 1969) developed a calculus to reason about situations bound in time. Allen (1983, 1984) posits a comprehensive set of temporal relations between intervals and rules that describe which inferences are possible from sets of these relations.

After that, we focused on computational work, mentioning several challenges that have been put forth recently, as well as the data sets that they have used and the solutions that have been found using these data sets. The Message Understanding Conferences (MUC-6, 1995; MUC-7, 1998) eventually took an interest in time expressions as part of named entity recognition tasks. This sort of task gained importance on its own, motivating the Temporal Expression Recognition and Normalization (TERN) challenge in 2004 (Ferro *et al.*, 2004). Since then, an interest has developed in more detailed annotations of time and the automated extraction of more phenomena related to time from text. The TimeML specification (Pustejovsky *et al.*, 2003a) has matured, corpora such as the TimeBank (Pustejovsky *et al.*, 2003b) have surfaced, and competitions like the two TempEval challenges (Pustejovsky & Verhagen, 2009; Verhagen *et al.*, 2007, 2010) have been conducted. In all of these, the focus has shifted to temporal relations between events and times or dates. A lot of research has been conducted based on them, too. It has been dominated by machine learning approaches and focused mostly on English. Work on the temporal processing of other languages has started, with the appearance of annotated data sets for Chinese (Cheng *et al.*, 2008), French (Bittar *et al.*, 2011), Korean (Im *et al.*, 2009), etc.

Chapter 3

Data

This chapter describes the data sets used for the development and testing of the solutions put forth in the remainder of this dissertation. In this work, we are interested in working with the Portuguese language. This language lacks the data sets with annotations about temporal information that are available for other languages and support the state of the art in temporal information processing described in Chapter 2. One of the significant contributions of the research work reported in this dissertation is the creation of TimeBankPT. To develop this resource, the adopted solution was to translate an existing resource (namely the English data set used in the first TempEval) to Portuguese and adapt it.

3.1 Outline

This chapter proceeds as follows. In Section 3.2 we motivate this choice of adapting an existing resource. Then, in Section 3.3 we briefly explain the XML tags used in the TempEval data. In Section 3.4, we describe how the original English data set was annotated. This resource has received some criticism, and that is presented in Section 3.5. Section 3.6 describes the adaptation process, and Section 3.7 reports on an approach to automatically detect errors in the corpus thus created, which are then manually corrected. Section 3.8 provides a quantitative comparison between the original English resource and the TimeBankPT data set. Section 3.9 tries to assess whether TimeBankPT, being a small corpus, contains enough data to support the tasks that it is intended to support. In Section 3.10 we mention a complication resulting from the fact that Portuguese is currently undergoing a spelling reform. We conclude this chapter with a small summary in Section 3.11.

3.2 Approach

It is important to be able to evaluate the performance of any tool, including those related to temporal information processing, in a way that is comparable to the results that can be found in the literature for similar tasks. To that end, data sets for Portuguese similar to those used in the literature are necessary.

It must also be mentioned that experimenting with different languages is important. As Chapter 4 will show, some improvements that can be obtained over the state of the art require approaches that are increasingly more language dependent.

In this work we are interested in working with the Portuguese language. For this language, there is the data used for the second HAREM evaluation. However, as far as temporal information is concerned, it only contains annotations for temporal expressions. The more interesting and harder problems of temporal information extraction—namely the extraction of temporal relations—cannot be explored with these data, as there are no annotated temporal relations.

The other two sources of data easiest to obtain are the ones used in TERN 2004 and the two TempEval evaluations. The TERN 2004 data are also only annotated for temporal expressions, whereas the data used in the TempEval challenges contain further annotations for event terms and relations between event terms and temporal expressions. However, both TERN 2004 and TempEval data are for other languages. In order to use them, it is necessary to adapt them to Portuguese. Another possibility is to annotate Portuguese text from scratch.

The chosen approach was to adapt the data of the first TempEval to Portuguese. It must be noted that the data sets of TempEval-2 largely overlap those of the first TempEval, so there is not much of a point in adapting both sets.

This option has several advantages over fully annotating Portuguese text from scratch. The data for TempEval were annotated by more than one person and then checked for consistency. By adapting it, one obtains comparable data without the need to have access to multiple annotators.

Additionally, the annotators used a special web interface to annotate it. The TimeML annotations contain many references to EVENT elements and TIMEX3 elements, as can be seen in Figure 2.5 (in the previous Chapter), whose content is repeated here in Figure 3.1. As such, using a special interface to annotate the data avoids XML coding mistakes. So, ideally, an annotation workbench would have been needed, or at least a tool to check annotation consistency, if we had pursued the option of developing the data set from scratch.

Data adaptation should also be faster than full annotation. A substantial part of the annotations cross over from the English corpus to the Portuguese one unchanged. For instance, as long as one is careful in trying to maintain a 1-to-1 relation between the EVENT and the TIMEX3 elements in the English and the Portuguese texts, the original TLINK elements can be used just like they are in the original data.

Another advantage is that the data obtained are comparable to the original English data used in TempEval, and results of the work presented in this dissertation will also be more comparable to the results for English that can be found in the literature, as many of them are based on the TimeBank data or the TempEval data (they are roughly the same data).

This Portuguese corpus resulting from adapting the English data used in the first TempEval is called TimeBankPT and is available at http://nlx.di.fc.ul.pt/~fcosta/TimeBankPT.

3.3 TimeML

The TimeML annotation guidelines can be found in Pustejovsky *et al.* (2003a, 2005); Saurí & Pustejovsky (2009); Saurí *et al.* (2006, 2009). In this section, we present a brief description of the TimeML annotations employed in the TempEval data for English.

Figure 3.1 contains an excerpt of a document from the original TempEval corpus. Event terms are tagged with EVENT tags, temporal expressions are inside TIMEX3 elements, and temporal relations between temporal entities are represented with TLINK elements.

```
<?xml version="1.0"?>
<TempEval>

ABC<TIMEX3 tid="t52" type="DATE" value="1998-01-14"
temporalFunction="false"
functionInDocument="CREATION_TIME">19980114</TIMEX3>.1830.0611
NEWS STORY

<s>In Washington <TIMEX3 tid="t53" type="DATE" value="1998-01-14"
temporalFunction="true" functionInDocument="NONE"
anchorTimeID="t52">today</TIMEX3>, the Federal Aviation
Administration <EVENT eid="e1" class="OCCURRENCE" stem="release"
aspect="NONE" tense="PAST" polarity="POS" pos="VERB">released
</EVENT> air traffic control tapes from <TIMEX3 tid="t54"
type="TIME" value="1998-XX-XXTNI" temporalFunction="true"
functionInDocument="NONE" anchorTimeID="t52">the night
</TIMEX3> the TWA Flight eight hundred <EVENT eid="e2"
class="OCCURRENCE" stem="go"
aspect="NONE" tense="PAST" polarity="POS" pos="VERB">went</EVENT>
down.</s>
...
<TLINK lid="l1" relType="BEFORE" eventID="e2" relatedToTime="t53"
task="A"/>
<TLINK lid="l2" relType="OVERLAP" eventID="e2"
relatedToTime="t54" task="A"/>
<TLINK lid="l4" relType="BEFORE" eventID="e2" relatedToTime="t52"
task="B"/>
...
</TempEval>
```

Figure 3.1: Sample of the data annotated for TempEval, corresponding to the fragment:
ABC.19980114.1830.0611
NEWS STORY
In Washington today, the Federal Aviation Administration released air traffic control tapes from the night the TWA Flight eight hundred went down.

3.3.1 Events

The TimeML annotation guidelines define an event as “a cover term for situations that *happen*, *occur*, *hold* or *take place*”, adding that they “can be punctual” (20a) or “last for a period of time” (20b), and they include “those predicates describing *states* or *circumstances* in which something obtains or holds true” (20c). That is, they cover all situation types, not just events in the narrow sense (that excludes states, cf. Section 2.2.2).

(20) a. A fresh flow of lava, gas, and debris **erupted** there Saturday.

b. “We’re **expecting** a major eruption,” he said in a telephone interview early today.

c. Israel has been scrambling to buy more masks abroad, after a **shortage** of several hundred thousand gas masks.

Events may be expressed by means of verbs (21a), nouns (21b), adjectives (21c), prepositions (21d), etc. In predicative contexts, such as in (21c) and (21d) below, the copula (the forms *was* and *been* in these two sentences) is not annotated but rather its complement (the adjective *under-development* or the preposition *in* in these examples).

(21) a. A fresh flow of lava, gas, and debris **erupted** there Saturday.

b. Israel will ask the United States to delay a military **strike** against Iraq until the Jewish state is fully prepared for a possible Iraqi **attack**.

c. France was **under-developed** in the eighteenth century, and Germany at the beginning of the nineteenth.

d. No woman has been **in** charge of the mission until now.

The relevant attributes of EVENT elements are the following:

- eid: an identifier for the event.
- stem: the event term’s lemma, i.e. its dictionary form.
- pos: the term’s part-of-speech, with the values VERB, NOUN, ADJECTIVE, PREP or OTHER.

- polarity: this attribute takes the value NEG if the event term is in a negative syntactic context, and POS otherwise.

- tense: the grammatical tense if the event term is a verb, or NONE otherwise.

- aspect: the grammatical aspect—NONE (e.g. *he runs*), PROGRESSIVE (e.g. *he is running*), PERFECTIVE (e.g. *he has run*) or PERFECTIVE_PROGRESSIVE (e.g. *he has been running*).

- class: this attribute encodes several levels of information. It distinguishes states (values STATE and I_STATE) from events that are not states (values OCCURRENCE and I_ACTION). It thus represents a binary distinction related to aspectual class (see Section 2.2.2). Additionally, it states whether the word associated with this event takes a clause as its complement (values I_STATE and I_ACTION) or not (values STATE and OCCURRENCE).[1] Some examples of I_STATE terms are *expect*, *predict*. The class of STATE terms contains event terms like *have*, *owe*, *standstill*. Examples of OCCURRENCE terms are *earn*, *close*, *drop*. The I_ACTION terms include *avoid*, *try*, *estimate*.

  Three special subtypes of I_ACTIONs are also marked in this attribute, by means of three dedicated values: REPORTING, for terms like *say*, *inform* or *announce*; PERCEPTION, for terms like *see* or *hear*; and ASPECTUAL, with examples such as *begin*, *stop* or *continue*.

3.3.2 Temporal Expressions

Temporal expressions are terms or expressions that refer to calendar dates, clock times or periods of time. They are usually adverbs (*yesterday*) or noun phrases (*next year*; *the current month*).

Temporal expressions are marked up with TIMEX3 tags and the following attributes:

- tid: an identifier for the timex.

[1] Here, the I_ prefix stands for *intensional*.

- type: the type attribute has the values DATE if the timex refers to a calendar date, TIME if it describes a time of the day, DURATION if it denotes a duration, or SET, used for sets of times. Some examples:
  - DATE: *Friday, October 1, 1999*; *the second of December*; *yesterday*; *last summer*; *next week.*
  - TIME: *ten minutes to three*; *five to eight*; *9 a.m. Friday, October 1, 1999.*
  - DURATION: *2 months*; *3 hours.*
  - SET: *every Thursday*; *twice a week.*
- value: the timex's value encodes a normalized representation of this temporal entity, in the ISO 8601 format. This representation can take one of three forms:
  - Most dates and times are expressed as a string matching the regular expression dddd(-dd(-dd(Tdd(:dd(:dd(.ddd)?)?)?)?)?)?, where d indicates a digit or the character X, which is used to fill in unknown values. Its meaning is *year-month-day*T*hour:minute:second.millisecond.* T is used to separate the date from the time. There are more possibilities to encode dates and times. For instance, seasons of the year or parts of the day can be used: 1990-SU is *the summer of 1990*, and *tomorrow night* might get the value 1990-10-10TNI.
  - Durations are coded as a string matching the pattern P(d+u)*(Td+u)*, where d indicates a digit and u indicates a unit (*Y* for years, etc..). P is a prefix used to indicate that what follows is a duration. T is once again the time separator (P2M is *two months* and PT2M is *2 minutes*).
  - One of the vague descriptions: PAST_REF, PRESENT_REF, FUTURE_REF. Examples of time expressions with such values are *now*, *the past*, *the future.*

  Some examples follow (for many of them it is crucial to know the document's creation time):

  - "I was sick <TIMEX3 value="1999-07-14">*yesterday*</TIMEX3>"

  - “After an emergency meeting in <TIMEX3 value="1994-11">*November* </TIMEX3>, relations began to improve”
  - “A Chinese gymnast was paralyzed in the Goodwill Games <TIMEX3 value="1998-SU">*last summer*</TIMEX3>”
  - “The U.N. Secretary-General departs <TIMEX3 value="1999-W28-WE"> *this weekend*</TIMEX3> for Baghdad”
  - <TIMEX3 value="1999-10-01T09">*9 a.m. Friday, October 1, 1999* </TIMEX3>
  - “The sponsor arrived at <TIMEX3 value="1999-07-15T14:50">*ten minutes to 3*</TIMEX3>”
  - “NATO may be changing a military destiny <TIMEX3 value="PAST_REF">*once*</TIMEX3> based on geography to a defense of common values”
  - “The gestation period in humans is <TIMEX3 value="P9M">*nine months* </TIMEX3>”
  - “The video is only <TIMEX3 value="PT30M">*half an hour*</TIMEX3> long”
  - “She is part of the most visible and influential presence that women have had in the <TIMEX3 value="P52Y">*52-year*</TIMEX3> history of the United Nations”

- mod: the mod attribute is optional, and it encodes somewhat vague information that cannot be represented by the ISO 8601 specification. It is used for expressions like *early this year*, which are annotated with mod="START". For instance:

  - “The restaurant opened <TIMEX3 value="1996" mod="APPROX">*about three years ago*</TIMEX3>”
  - “who served briefly in Congress <TIMEX3 value="1989" mod="BEFORE"> *more than a decade ago*</TIMEX3>”
  - “There is certain to be excitement at <TIMEX3> value="2000" mod= "START">*the dawn of 2000*</TIMEX3>”

  - <TIMEX3 value="PT5M" mod="MORE_THAN">*more than 5 minutes* </TIMEX3>

- freq and quant: these are used to describe timexes that have been annotated to have SET as their type:
  - "Our fund has had positive net sales <TIMEX3 type="SET" quant= "EVERY" value="P1M">*every month*</TIMEX3> for the last three years"

- functionInDocument, temporalFunction and anchorTimeID: as can be seen in Figure 3.2 there are other attributes for timexes that encode whether it is the document's creation time (functionInDocument) and whether its value can be determined from the expression alone or rather depends on the value of another temporal expression (temporalFunction and anchorTimeID):
  - "For the six months ended <TIMEX3 tid="t149">*Aug. 12*</TIMEX3>, Provigo posted net income of C $6.5 million, or eight Canadian cents a share, compared with C $18.1 million, or 21 Canadian cents a share, <TIMEX3 value="1988-08-12" temporalFunction="true" anchorTimeID="t149">*a year earlier*</TIMEX3>"

3.3.3 Temporal Relations

The TLINK elements encode temporal relations, and their attributes are:

- eventID: this attribute holds a reference to the first argument of the relation.

- relatedToTime and relatedToEvent: the second argument is given by the attribute relatedToTime (if it is a time, date or duration) or relatedToEvent (if it is another event).

- relType: the type of the temporal relation:
  - OVERLAP

    "In *the last twenty four hours*$_{t2}$, the value of the Indonesian stock market has *fallen*$_{e5}$ by twelve percent."

    <TLINK relType="OVERLAP" eventID="e5" relatedToTime="t2"/>

- BEFORE

  "In space, some say female pilots were *held*$_{e35}$ up until *now*$_{t5}$ by the lack of piloting opportunities for them in the military."

  <TLINK relType="BEFORE" eventID="e35" relatedToTime="t5"/>

- AFTER

  "And at the brokerage houses, after *ten years*$_{t9}$ of boom, they're *talking*$_{e48}$ about layoffs."

  <TLINK relType="AFTER" eventID="e48" relatedToTime="t9"/>

- BEFORE-OR-OVERLAP

  "Nigeria state radio says thousands of people *began*$_{e12}$ gathering in the capital Abuja early Tuesday for the *two day*$_{t4}$ rally supporting General Sani Abacha's candidacy."

  <TLINK relType="BEFORE-OR-OVERLAP" eventID="e12" relatedToTime="t4"/>

- OVERLAP-OR-AFTER

  "In a joint statement with Tourism Minister Andrew Thomson, it said two new flights would *leave*$_{e13}$ Bombay on Monday and Tuesday nights from *March 30*$_{t9}$, with the third departing each Thursday from August 6."

  <TIMEX3 relType="OVERLAP-OR-AFTER" eventID="e13" relatedToTime="t9"/>

- VAGUE

  "Cilcorp said the business to be *acquired*$_{e12}$ had revenue of $76 million for the year ended *March 31*$_{t32}$"

  <TIMEX3 relType="VAGUE" eventID="e12" relatedToTime="t32"/>

- lid: an identifier for the relation.

- task: the TempEval task to which this temporal relation pertains (the value A for Task A Event-Timex, B for Task B Event-DocTime, or C for Task C Event-Event in the case of the first TempEval).

Task	Accuracy (%)	Kappa		
		Average	Lowest	Highest
Task A Event-Timex	69	0.54	0.28	0.70
Task B Event-DocTime	74	0.54	0.27	0.76
Task C Event-Event	65	0.47	0.18	0.63

Table 3.1: Inter-annotator agreement on the TempEval tasks (lowest and highest refer to annotator pairings). Adapted from Verhagen *et al.* (2009)

3.4 The Annotation Process

The English TempEval data were created with manual annotation aided by a web-based interface that would automatize part of the process (Verhagen, 2005). Each document in this corpus was annotated by two participants, and overall there were seven annotators. Afterward, three experienced annotators decided the cases where there was disagreement between the two original annotations.

An important piece of information is the inter-annotator agreement, because it sheds light on how hard this task is for humans. Verhagen *et al.* (2009) cite pair-wise kappa scores (Carletta, 1996; Cohen, 1960) for inter-annotator agreement that are for no pair higher than 0.76, which is lower than the 0.8 mark that is often considered reliable.[1] This suggests that the tasks are either ill-defined or that they are hard (or subjective) and that therefore we should not be expecting perfect results from an automated solution, as that would be performing substantially better than humans. The average agreement scores per task are shown in Table 3.1.

Indeed, the same work reports that 19% of the disagreements involved the VAGUE class, an indication that many instances were genuinely difficult to annotate. However, 9% of disagreements were between AFTER and BEFORE, which suggests that the underlying reason for a considerable number of the divergent decisions was simply confusion about which argument was which in the relation.

[1] The kappa coefficient is defined in terms of the observed inter-annotator agreement, $P(A)$, and the agreement expected by chance, $P(E)$:

$$\kappa = \frac{P(A) - P(E)}{1 - P(E)} \tag{3.1}$$

	All systems correct		No system correct	
Task	Count	%	Count	%
Task A Event-Timex	24	14	33	20
Task B Event-DocTime	160	45	36	11
Task C Event-Event	35	14	40	16

Table 3.2: Easy and hard test instances in TempEval. Adapted from Lee & Katz (2009).

3.5 System Performance in TempEval

Lee & Katz (2009) present an error analysis of the systems that took part in the TempEval competition.

Some of the test items were much harder to correctly classify than others. For Task A Event-Timex and Task C Event-Event, 14% of the data was classified correctly by all systems, but, depending on the task, up to 20% of the test data was classified incorrectly by all of them. Task B Event-DocTime seemed easier, as 45% of all data was correctly processed by all systems, and only 11% could not be correctly classified by a single system. Table 3.2 is an overview of these numbers.

One observation is that results are better when the events are denoted by verbs, because the attributes tense and aspect help the decision. For those events that are given by nouns, the results are much worse in general (they get the value NONE for these attributes).

They also conclude that the poor performance is in part due to data sparseness. For instance, the disjunctive classes (VAGUE, BEFORE-OR-OVERLAP and OVERLAP-OR-AFTER) have very bad results, because of very few training instances with these classes.

3.6 TimeBankPT

As mentioned at the beginning of this chapter, the English data used in TempEval were adapted to Portuguese.

All TimeML markup in the TempEval data was first removed and the resulting text was input to the Google Translator Toolkit.[1] This tool combines machine translation with a translation memory. All the data were translated to Portuguese in this semi-automated way, by manually correcting the suggested translations.

After that, there were three collections of documents (the original TimeML data, the English unannotated data and the Portuguese unannotated data) aligned by paragraphs (the line breaks from the original collection were unchanged in the other collections). This way, for each paragraph in the Portuguese data, all the corresponding TimeML tags in the original English paragraph are known.

We tried using support software, namely GIZA++ (Och & Ney, 2003), to perform word alignment on the unannotated texts, which would have enabled us to transpose the TimeML annotations automatically. However, word alignment algorithms do not have 100% accuracy, so the results would have to be checked manually. Therefore, this idea was abandoned, and instead we simply placed the different TimeML markup in the correct positions manually. A small script was developed to place all relevant TimeML markup at the end of each paragraph in the Portuguese text, and then each tag was manually repositioned to the appropriate place in that paragraph. It is of note that the TLINK elements always occur at the end of each document, each in a separate line: therefore they do not need to be repositioned. They are just copied over unchanged.

The resulting data were checked automatically for possible errors and then manually corrected: we checked their conformance to a DTD as well as more specific constraints (for instance, the stem value for verbs must end with an *r* in Portuguese).

The creation of TimeBankPT and the corpus itself are also described in Costa & Branco (2010, 2012d).

3.6.1 Annotation Decisions

When porting the TimeML annotations from English to Portuguese, a few decisions had to be made. For illustration purposes, Figure 3.2 contains the Portuguese equivalent of the extract presented in Figure 3.1.

For TIMEX3 elements, the issue is that, when the temporal expression to be annotated is a prepositional phrase, the preposition should not be inside the TIMEX3

[1] http://translate.google.com/toolkit

```
<?xml version="1.0" encoding="UTF-8"?>
<TempEval>

ABC<TIMEX3 tid="t52" type="DATE" value="1998-01-14"
temporalFunction="false" functionInDocument="CREATION_TIME">
19980114</TIMEX3>.1830.1611
REPORTAGEM

<s>Em Washington, <TIMEX3 tid="t53" type="DATE"
value="1998-01-14" temporalFunction="true"
functionInDocument="NONE" anchorTimeID="t52">hoje</TIMEX3>, a
Federal Aviation Administration <EVENT eid="e1"
class="OCCURRENCE" stem="publicar" aspect="NONE" tense="PPI"
polarity="POS" pos="VERB">publicou</EVENT> gravações do controlo
de tráfego aéreo da <TIMEX3 tid="t54" type="TIME"
value="1998-XX-XXTNI" temporalFunction="true"
functionInDocument="NONE" anchorTimeID="t52">noite</TIMEX3>
em que o voo TWA800 <EVENT eid="e2" class="OCCURRENCE"
stem="cair" aspect="NONE" tense="PPI" polarity="POS" pos="VERB">
caiu</EVENT>.</s>
...
<TLINK lid="l1" relType="BEFORE" eventID="e2" relatedToTime="t53"
task="A"/>
<TLINK lid="l2" relType="OVERLAP" eventID="e2"
relatedToTime="t54" task="A"/>
<TLINK lid="l4" relType="BEFORE" eventID="e2" relatedToTime="t52"
task="B"/>
...
</TempEval>
```

Figure 3.2: Sample of TimeBankPT, corresponding to the fragment:
ABC.19980114.1830.0611
REPORTAGEM
Em Washington, hoje, a Federal Aviation Administration publicou gravações do controlo de tráfego aéreo da noite em que o voo TWA800 caiu.

tags according to the TimeML specification. In the case of Portuguese, this raises the question of whether to leave contractions of prepositions with determiners outside these tags (in the English data the preposition is outside and the determiner is inside).[1]

We chose to leave them outside, as can be seen in that Figure. In this example, the prepositional phrase *from the night*/*da noite* is annotated with the English noun phrase *the night* inside the TIMEX3 element, but the Portuguese version only contains the noun *noite* inside those tags. By contrast, the HAREM data included prepositions (and also contractions) inside the elements marking temporal expressions. In any case, since the list of Portuguese prepositions and contractions is finite and quite small, it is straightforward to automatically check whether the token immediately preceding a TIMEX3 is a preposition or a contraction, and even to transform the data automatically in order to include or exclude these elements from TIMEX3s.

The attributes of TIMEX3 elements carry over to the Portuguese corpus unchanged.

In the case of EVENT elements, some of the attributes are adapted. The value of the attribute stem is obviously different in Portuguese. The attributes aspect and tense have a different set of possible values in the Portuguese data, simply because the morphology of the two languages is different. In the example in Figure 3.1 the value PPI for the attribute tense stands for *pretérito perfeito do indicativo*. We chose to include mood information in the tense attribute because the different tenses of the indicative and the subjunctive moods do not line up perfectly as there are more tenses for the indicative than for the subjunctive. Appendix I lists all possible values of the tense attribute. For the aspect attribute, which encodes grammatical aspect, we only use the values NONE and PROGRESSIVE, leaving out the values PERFECTIVE and PERFECTIVE_PROGRESSIVE, as the Portuguese expressions formed in ways similar to the English perfect (*have* combined with a past participle) do not carry the same semantic value (Portner, 2003), and as such were not annotated as having perfect aspect.

[1] The fact that prepositions are placed outside of temporal expressions may seem odd at first, but this is because in the original TimeBank, from which the TempEval data were derived, they are annotated with SIGNAL tags. The TempEval data do not contain SIGNAL elements, however.

The TLINK elements are taken verbatim from the original documents.

3.7 Automated Error Mining

It is possible to automatically detect errors in temporal annotation. For instance, if an event A is annotated as temporally preceding another event B, and B is annotated as preceding C, A must precede C as well, because temporal precedence is a transitive relation. If we then find an annotation according to which C precedes A, we have a temporal loop, and something is wrong. We ran a temporal reasoning system on the adapted data, which enabled us to detect this kind of error.

The original TempEval data had been similarly checked for consistency Verhagen (2005). However, our reasoning component performs one extra step, that allowed us to identify more possible annotation errors: before applying any temporal reasoning rules, it first orders annotated temporal expressions according to their normalized value (e.g. the date 1989-09-29 is ordered as preceding 1989-10-02). That is, we exploit the TIMEX3 annotations in order to enrich the set of temporal relations that we work with, and more specifically we make use of the value attribute of TIMEX3 elements. In this way, we end up having a much larger set of initial temporal relations than the set of those that are explicitly annotated. All temporal relations that are explicitly annotated are binary and involve at least one event. Our approach further adds a large number of temporal relations between dates or times.

The corpus distribution contains a file where each error that was discovered with the help of temporal reasoning is described. This file serves as documentation about the changes introduced during the adaptation process, but from these descriptions it is also easy to identify the corresponding data in the original English corpus.

The inference procedure allowed for the detection of around 80 problems in the corpus (affecting both the train and the test sets), that were then manually corrected. These corrections result in some differences between TimeBankPT and the original TempEval English data. Since they affect the type of the annotated temporal relations, they cause differences in the distribution of temporal relations. In Section 3.8, we quantify the effect of these corrections on the data, by comparing the distribution of temporal relations in TimeBankPT with that in the English TempEval data set.

Several authors have used reasoning as a means to aid temporal annotation. Katz & Arosio (2001) used a temporal reasoning system to compare the temporal annotations of two annotators. In a similar spirit, Setzer & Gaizauskas (2001) first compute the deductive closure of annotated temporal relations so that they can then assess annotator agreement with standard precision and recall measures.

As mentioned above, Verhagen (2005) uses temporal closure as a means to aid TimeML annotation. He reports that closing a set of manually annotated temporal relations more than quadruples the number of temporal relations in TimeBank (see Section 2.5), a corpus that is the source of the data used for the TempEval challenges.

A considerable amount of work in the area of temporal information processing has used reasoning components in the proposed solutions. One recent example is the work of Ha *et al.* (2010), a participant of the second TempEval, and there are several others.

3.7.1 Ordering of Dates and Times

As already mentioned, temporal expressions are ordered according to their normalized value. For instance, the date 2000-01-03 is ordered as preceding the date 2010-03-04. Since all temporal expressions are normalized in the annotated data, we order temporal expressions before applying any temporal reasoning. This increases the number of temporal relations we start with, and the potential number of relations we end up with after applying temporal reasoning.

To this end, we used Joda-Time 2.0 (http://joda-time.sourceforge.net). Each normalized date or time is converted to an interval.

In many cases it is possible to specify the start and end points of this interval. For instance, the date 2000-01-03 is represented internally by an interval with its start point at 2000-01-03T00:00:00.000 and ending at 2000-01-03T23:59:59.999. Many different kinds of normalized expressions require many rules. For instance, an expression like *last Winter* could be annotated in the data as 2010-WI, and dedicated rules are used to get its start and end points.

Some time expressions are normalized as PRESENT_REF (e.g. *now*), PAST_REF (*the past*) or FUTURE_REF (*the future*). These cases are not represented by any Joda-Time object. Instead we need to account for them in a special way. They can be temporally ordered among themselves (e.g. PRESENT_REF precedes FUTURE_REF), but generally not with other temporal expressions. We further stipulate that PRESENT_REF includes each document's creation time (which therefore precedes FUTURE_REF, etc.). So, in additional to the representation of times and dates as time intervals, we employ a layer of specifically designed rules.

Chambers & Jurafsky (2008a) also order dates with hand-crafted rules before applying reasoning to increase the number of explicit temporal relations. Their work is, however, more limited: they only order dates (we also order times); when doing so, they only look at the year, month and day of the month (the normalized value of temporal expressions can be represented by resorting to other fields, such as the season of the year, which we explore). In addition, our work uses a richer set of temporal relations (we allow for inclusion relations between dates/times) and a richer set of reasoning rules.

3.7.2 Deduction Procedure

The rules implemented in our reasoning component are:

- Temporal precedence is transitive, irreflexive and antisymmetric;
- Temporal overlap is reflexive and symmetric;
- If A does not precede B, then either B precedes A or A and B overlap;
- If A overlaps B and B precedes C, then C does not precede A.

Because we also consider temporal relations between times and dates, we also deal with temporal inclusion, a type of temporal relation that is not part of the annotations used in the TempEval data but that is still useful for reasoning. We make use of the following additional rules, dealing with temporal inclusion:

- Temporal inclusion is transitive, reflexive and antisymmetric;
- If A includes B, then A and B overlap;
- If A includes B and C overlaps B, then C overlaps A;
- If A includes B and C precedes A, then C precedes B;

- If A includes B and A precedes C, then B precedes C;
- If A includes B and C precedes B, then either C precedes A or A and C overlap (A cannot precede C).
- If A includes B and B precedes C, then either A precedes C or A and C overlap (C cannot precede A).

3.8 Comparison with the English Data Set

One interesting question is how different the English and the Portuguese data sets are. Obviously, the results obtained using the adapted data cannot be strictly comparable to the results that have been published based on the TempEval English data, since there is still the language difference. It is interesting nevertheless to have an idea of how similar or different the two data sets are.

3.8.1 Size, Annotations and Class Distribution

The original English data for TempEval are organized in two data sets: one for training and development and another one for evaluation. The full data are organized in 182 documents (162 documents in the training data and another 20 in the test data). Each document is a news report from television broadcasts, newswire or newspapers. A large amount of the documents (123 in the training set and 12 in the test data) are taken from several issues of the Wall Street Journal dating from 1989. These texts are usually smaller than the other ones, and contain a large amount of jargon and stock market data.

Table 3.3 compares the original English corpus and TimeBankPT. In this table, an additional word count is presented for Portuguese, obtained by counting the words identified as such by a part-of-speech tagger (see Section 4.3.1) that treats punctuation marks as separate word tokens and also expands contractions into their composing elements.

The major difference between the two data sets is the number of words, which is due to language differences, with Portuguese being more verbose than English. The remaining, small differences between the English and Portuguese data sets are

	English		Portuguese	
	Train	Test	Train	Test
Sentences	2,236	376	2,281	351
Words (according to whitespace)	52,740	8,107	60,782	8,920
Words (according to tagger)			68,351	9,829
Annotated events	6,799	1,103	6,790	1,097
Annotated temporal expressions	1,244	165	1,244	165
Annotated temporal relations				
Task A Event-Timex	1,490	169	1,490	169
Task B Event-DocTime	2,556	331	2,556	331
Task C Event-Event	1,744	258	1,735	258
Total	5,790	758	5,781	758
Words / events	7.76	7.35	8.95	8.13
Words / temporal expressions	42.4	49.13	48.86	54.06

Table 3.3: Counts for the English TempEval data and TimeBankPT

the result of the small corrections to the data resulting from the automated error mining process described above in Section 3.7.

Table 3.4 shows the class distributions for the three TempEval tasks, both for the English data used in TempEval and for TimeBankPT, in full detail. As can be seen from that table, the differences are very small.

3.8.2 Classifier Performance

Another way to compare the two sets is to check the performance of easily reproducible approaches to the problems of TempEval on each data set.

One participant of TempEval was the USFD system (Hepple *et al.*, 2007). The USFD system implemented a straightforward solution: it simply trained classifiers with Weka (Witten & Frank, 2005), coding as attributes information that was readily available in the data and did not require any natural language processing. For all tasks, the attribute relType of TLINK elements is unknown and must be discovered, but all other information is given. So all other information that was annotated could be used as classifier features.

The authors' objectives were to see "whether a 'lite' approach of this kind could yield reasonable performance, before pursuing possibilities that relied on 'deeper'

		Task A		Task B		Task C	
Set	Class	EN	PT	EN	PT	EN	PT
Train	BEFORE	19%	19%	62%	62%	25%	25%
	AFTER	25%	25%	14%	14%	18%	17%
	OVERLAP	50%	49%	19%	19%	42%	42%
	BEFORE-OR-OVERLAP	2%	2%	2%	2%	4%	4%
	OVERLAP-OR-AFTER	2%	2%	1%	1%	3%	3%
	VAGUE	2%	2%	2%	1%	9%	9%
Test	BEFORE	12%	11%	56%	56%	23%	23%
	AFTER	18%	18%	15%	15%	16%	16%
	OVERLAP	57%	59%	24%	25%	47%	47%
	BEFORE-OR-OVERLAP	1%	2%	2%	3%	5%	5%
	OVERLAP-OR-AFTER	3%	3%	1%	0%	3%	3%
	VAGUE	8%	7%	2%	2%	6%	6%

Table 3.4: Class distributions for the three tasks, the two data sets in each corpus (train and test) and the two corpora (English, EN, and Portuguese, PT).

NLP analysis methods", "which of the features would contribute positively to system performance" and "if any [machine learning] approach was better suited to the TempEval tasks than any other". In spite of its simplicity, they obtained results quite close to the best systems, specially for Task A Event-Timex and Task C Event-Event.

Replicating their experiments on the adapted data helps further comparisons between the two data sets. This section describes these results.

These authors experimented with the set of features shown in Table 3.5. They started with the full set of features and removed them one by one whenever that improved classifier performance. The features that remained at the end are marked with a check mark (✓) in that table.

In this table, the features with a name starting with event- are based on the attributes of TimeML EVENT elements with the same name. The ones with a name beginning with timex3- are taken from TIMEX3 elements. The order- attributes are computed by simple string manipulation of the TimeML annotated documents: order-event-first encodes whether the event appears in the document before the times; order-event-between whether there is an annotated event term in the text between the two entities; order-timex-between is similar, but considers temporal expressions; and order-adjacent is true if and only if both order-event-between and order-timex-between

	Task		
Attribute	Task A	Task B	Task C
event-aspect	✓	✓	✓
event-polarity	✓	✓	×
event-pos	✓	✓	✓
event-stem	✓	×	×
event-string	×	×	×
event-class	×	✓	✓
event-tense	×	✓	✓
order-adjacent	✓	N/A	N/A
order-event-first	✓	N/A	N/A
order-event-between	×	N/A	N/A
order-timex-between	×	N/A	N/A
timex3-mod	✓	×	N/A
timex3-type	✓	×	N/A

Table 3.5: Features used by Hepple *et al.* (2007) in TempEval.

are false (even if some textual material actually occurs between the two annotated elements). The last feature is the class attribute.

They tested several classifier algorithms, using the chosen set of features. Their results are presented in Table 3.6. The machine learning algorithms they tested are:

- rules.DecisionTable is a decision table classifier (Kohavi, 1995).
- rules.JRip is a propositional rule learner implementing the RIPPER algorithm of Cohen (1995).
- lazy.KStar is a nearest neighbor classifier that uses an entropy-based similarity function (Cleary & Trigg, 1995).
- bayes.NaiveBayes is a Bayesian classifier (John & Langley, 1995).
- functions.SMO is an algorithm to train support vector machines (Platt, 1998).

Each classifier was tested using ten-fold cross-validation on the training data. The best classifier for each task was then used for evaluation. The results they obtained in TempEval can be seen in the *English* column of Table 3.8.

We reproduced this experiment using TimeBankPT. This yields the results in Table 3.7 for the training data. Here the same attributes were used as the ones

	Score (%)		
Algorithm	Task A	Task B	Task C
lazy.KStar	**58.2**	76.7	54.0
rules.DecisionTable	53.3	**79.0**	52.9
functions.SMO	55.1	78.1	**55.5**
rules.JRip	50.7	78.6	53.4
bayes.NaiveBayes	56.3	76.2	50.7
Baseline	49.8	62.1	42.0

Table 3.6: Weka classifiers used by Hepple *et al.* (2007), and their performance on the TempEval training data, with cross-validation. The best result for each task is in boldface and was used for evaluation.

	Score (%)		
Algorithm	Task A	Task B	Task C
lazy.KStar	56.0	77.7	54.4
rules.DecisionTable	49.9	78.5	50.2
functions.SMO	**56.6**	**79.3**	**56.5**
rules.JRip	51.6	77.4	51.4
bayes.NaiveBayes	56.1	78.4	53.9
Baseline	49.3	62.2	41.8

Table 3.7: Weka classifiers on the Portuguese training data, with cross-validation.

reported by Hepple *et al.* (2007) and presented in Table 3.5. Table 3.8 shows the results on the test data, under the column *Portuguese*. The results for Portuguese are in the second column. The algorithms used are the ones used by Hepple *et al.* (2007), and the same algorithms and feature combinations are used for Portuguese. These classifier and feature combinations are optimized for English, but they serve our purpose of comparing the two data sets.

The results in Table 3.8 show that, despite language differences and the additional corrections performed on the Portuguese data, the results on the two data sets are nevertheless quite comparable. From these results we conclude that the development of the Portuguese data set by adapting the English one was not lossy.

The most salient difference, when it comes to classifier performance, is for Task B Event-DocTime, with a 4% difference between the English data and the Portuguese data. We inspected the models produced by classification algorithms that output

		Score (%)	
Task	Algorithm	English	Portuguese
Task A Event-Timex	lazy.KStar	59	58
	Baseline	57	59
Task B Event-DocTime	rules.DecisionTable	73	77
	Baseline	56	56
Task C Event-Event	functions.SMO	54	54
	Baseline	47	47

Table 3.8: Weka classifiers on the test data

human readable models. We looked at the results of the RIPPER algorithm, presented above, and decision trees (Quinlan, 1993), which were not used by Hepple *et al.* (2007) but were also tried by us. For Task B Event-DocTime, we see that verb tense is the most important feature used by them. Because verb tense is language specific, we hypothesize that it is the differences in the tense system of the two languages that are behind the differences in the results for Task B Event-DocTime (i.e. they are due to language differences).

The other tasks do not seem to be as sensitive to tense. It makes sense that it is precisely Task B Event-DocTime that is affected the most by it, as this task is about temporal relations holding between events and the document's creation time, and verb tense is primarily an indicator of the temporal relation between the event denoted by the verb and the speech time.

In Table 3.8 we also present the majority class baselines for each task. The differences in the baselines between TimeBankPT and the TempEval corpus of English are due to the corrections to the data resulting from the automated error mining procedure described in Section 3.7

3.9 The Size of TimeBankPT

A corpus of approximately 70,000 words is small for many natural language processing tasks. In order to check whether the size of TimeBankPT is adequate for the tasks that it is meant to address (automatic temporal relation classification), one can measure the effect of the size of the data on classifier performance.

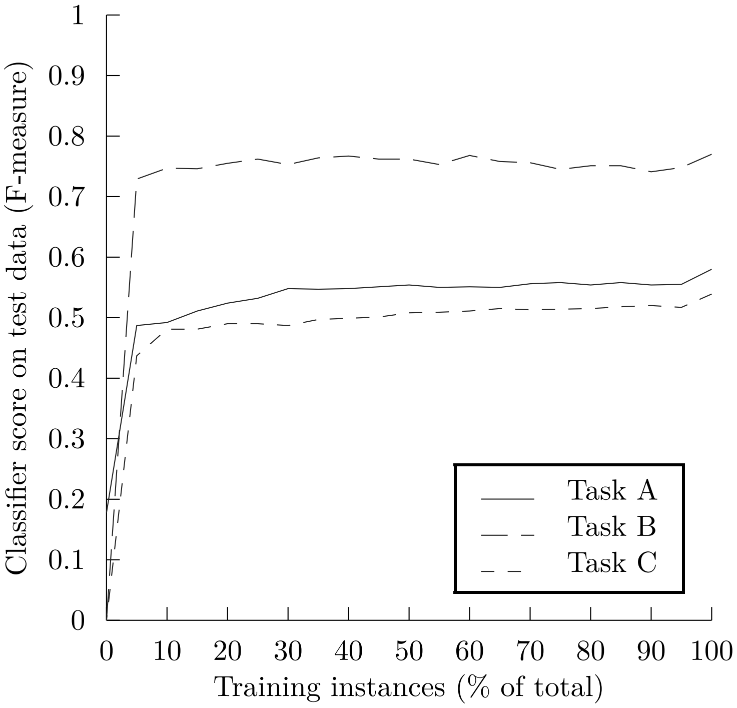


Figure 3.3: Classifier performance by size of training data

Figure 3.3 shows the performance of classifiers similar to the ones in Table 3.8 but trained with subsets of the training data. They were evaluated on the whole test set.

The machine learning algorithms employed to get the values shown there are the same as the ones in Table 3.8. The models were produced using the same feature set, too. Each value used to plot that graph is the average of ten samplings of the training data that differ only in as much as they use different seeds for the random number generator involved in the sampling process.

The performance of the classifiers for the three sorts of temporal relations appears quite stable across many sizes of training data. Classifier performance does go up with more training data, but it does so very slowly. Therefore, more data would likely not increase classifier performance very quickly.

Figure 3.4 shows similar data, this time using subsets of the test data. That is, the classifiers trained with the full training set were tested with subsets of the test data of different sizes. Each data point is also the average of ten runs that used the same amount of test data but different seeds to the random number generator used

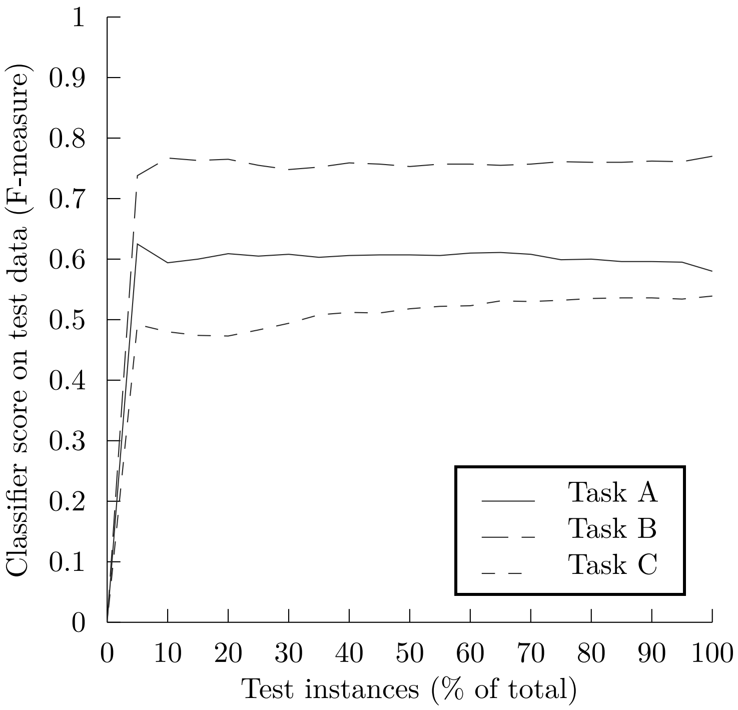


Figure 3.4: Classifier performance by size of test data

to sample the data. Once again, it can be seen that the curves are rather stable after an initial range of very short test data sizes, where, precisely because of the small size of the test data, the curves are a bit erratic and variation is high (not visible in that graph). This problem is more obvious in Figure 3.4 than in Figure 3.3 because the test data set is considerably smaller than the train data set (see Table 3.3).

From these two results we conclude that it appears that increasing the size of the corpus would not rapidly increase classifier performance.

3.10 A Note on Spelling

The spelling of the Portuguese language is currently going through a reform. The new spelling (Houaiss, 1991) is known as the 1990 spelling agreement but its coming into effect is quite recent.[1] It unifies the two official orthographies that existed for

[1] An official document with the spelling agreement can be found at http://www.dre.pt/pdf1s/1991/08/193A00/43704388.pdf.

Portuguese: the Brazilian spelling, followed by Brazil, and the European spelling, followed by the remaining Portuguese speaking countries.

The new orthography has already been ratified in five countries (Brazil, Cape Verde, East Timor, Guinea-Bissau, Portugal and São Tomé and Príncipe). Only two countries where Portuguese is official (Angola and Mozambique) have yet to ratify it. In 2009, several countries, including Brazil and Portugal, initiated a transitional period in which the old spelling is still acceptable, in parallel with the new one.

The most noticeable change to the spelling is, from the Brazilian point of view, the deletion of diacritic marks in some words. In many cases the European spelling did not use them already. So for instance, *ideia* ("idea") is now written like that by all speakers, whereas the old Brazilian spelling is *idéia*, and similarly for the word *frequente* ("frequent"), with the older spelling *freqüente*. The most striking change to the European orthography is the removal of silent consonants (consonants that were written solely because of etymology but had no phonological basis), that had already been abandoned in the Brazilian spelling. One example is the word *ótimo* ("great"), which has the old European spelling *óptimo*, with a silent *p*.

TimeBankPT features the unified orthography, so that the corpus remains useful for future research on the long run. This decision has, however, negative short term consequences, as the typical existing natural language processing tools, developed for the old spellings, may not have been updated yet. Error rates may be higher currently when processing data with the new spelling, as some frequent words are now out-of-vocabulary (because they have a different spelling) for the natural language processing tools not yet updated.

This is precisely what happened with some experiments reported in Chapter 4, that require the processing of TimeBankPT. In some cases the tools that were employed to process the data made errors that they would not have made if we had used the old European spelling, for which the tools were developed. We did not quantify the amount of error that the new spelling introduced in these tools because these errors appeared to be quite infrequent, and they were corrected manually.

3.11 Summary

In this chapter, we presented the data set we developed to be used to experiment with temporal information processing solutions for Portuguese, as reported in the following chapter. This corpus is TimeBankPT, which was developed by adapting the English data set used in the first TempEval to the Portuguese language. We described the temporal annotations that are used in the TempEval data and in TimeBankPT. We mentioned some shortcomings of the original resource—low inter-annotator agreement, some difficult instances in the test data and few training instances for some of the classes—, which should be kept in mind when interpreting results from tools or solutions that resort to the data of TempEval—and consequently the results based on TimeBankPT as well.

We explained how this adaptation was carried out, and we presented an effort to automatically detect annotation errors, based on the logical properties of the temporal relations being annotated. Finally, we provided an assessment of the differences between the original English corpus and TimeBankPT, as well as a discussion on the size of TimeBankPT.

Chapter 4

Classification of Temporal Relations

One of the major goals of this dissertation is to improve on the problem of temporal relation classification. This chapter is devoted to reporting on the results obtained with that goal in mind. The hypothesis is that many different kinds of information are needed to successfully classify the temporal relations implicit in a text: linguistic, but also pragmatic and logical. As Derczynski & Gaizauskas (2010) put it,

> Recent improvements (. . .) still yield marginal improvements (. . .). It seems that to break through this performance “wall”, we need to continue to innovate with and discuss temporal relation labeling, using information and knowledge from many sources to build practical high-performance systems.

In this chapter, we thus explore several different kinds of information with the purpose of improving the task of classifying temporal relations.

A note on the terminology employed throughout this text, for the sake of its readability, is in order. What we call event terms are terms, or words, that denote events; time expressions (or temporal expressions or timexes) are expressions, or phrases, that denote times, dates or durations (or times, for short). Temporal relations relate events and times; but what we find in text is event terms and time expressions. When there is no risk of confusion, we will use the term *time* to refer to anything that can be denoted by a time expression: a time (*5 p.m.*), a date (*June 20*), a duration (*two hours*), a set of times or dates (*every Friday*). Additionally, we will sometimes use *event* and *event term* interchangeably, to refer both to events and to event terms. We will also use *temporal expression*, *time expression*, *timex* and *time* interchangeably. This is to avoid long descriptions such as *the event denoted by the event term that occurs in the text after the time expression that denotes the time that...* (instead we just say *the event that occurs after the time that...*).

4.1 Outline

The general approach followed in this chapter is to develop new classifier features, with the purpose of improving the solutions to the problem of automatic temporal relation classification.

In order to evaluate these new classifier features, classifiers that incorporate these features are trained and evaluated, and then they are compared with baselines. Simple classifiers, that use relatively shallow features, serve as baselines for comparison. They are presented in Section 4.2 below. The data used for the training and evaluation is TimeBankPT, presented in the previous chapter.

The section that immediately follows that one (Section 4.3) presents several natural language processing tools that are used with the solutions developed here. The many features under testing are then described in Section 4.4. As mentioned, they encode different levels of information that are plausibly relevant to the task of temporal relation classification. Section 4.5 then explains how these features are selected to be part of the final solutions and evaluates these final classifiers, comparing them to the baselines. Finally, Section 4.6 is a short summary of this chapter.

4.2 Baselines

Different types of information are tried in the solutions for the automatic classification of temporal relations.

Several classifier features are tested, and many of them are new. These are presented in Section 4.4. In order to evaluate these classifier features, simple classifiers are employed as baselines. These baselines use a minimal set of relatively shallow features. They are presented in this section.

The baselines consist of machine learned classifiers similar to the ones used by Hepple *et al.* (2007). This was one of the participating systems of the first TempEval. It used machine learning algorithms implemented in Weka (Witten & Frank, 1999). Here we follow the same approach and test several of the algorithms implemented in Weka. These baseline classifiers are also similar to the classifiers used in Section 3.8.2 to compare TimeBankPT to the original English corpus that it is based on, with a few differences made explicit here. We present results for the same algorithms as Hepple *et al.* (2007) used in the first TempEval, and additionally for a decision trees algorithm, Weka's trees.J48. This last one was chosen because it is fast to train and produces human readable models, which is useful during development and for error mining.

The algorithms that were employed are:

- rules.DecisionTable is a decision table classifier;
- trees.J48 is Weka's implementation of the C4.5 algorithm to generate decision trees;
- rules.JRip is a propositional rule learner implementing the RIPPER algorithm;
- lazy.KStar is a nearest neighbor classifier that uses an entropy-based similarity function;
- bayes.NaiveBayes is a Bayesian classifier;
- functions.SMO is an algorithm to train support vector machines.

The default parameters are used for all of these algorithms, both in these baselines and in the final classifiers.

At this point, it should be mentioned again that the tasks of TempEval are to determine the type of temporal relations. Each train or test instance thus corresponds to a temporal relation, i.e. a TLINK element in the TimeML annotations (see Figures 2.5 and 3.2). The classification problem is to determine the value of the attribute relType of TimeML TLINK elements. These temporal relations relate an event (referred by the eventID attribute of TLINK elements) to another temporal entity, that can be a time (pointed to by the relatedToTime attribute), in the case of Task A Event-Timex and Task B Event-DocTime, or, in the case of Task C Event-Event, another event (given by the attribute relatedToEvent).

For the features that are employed in the baseline classifiers we also took inspiration in the approach of Hepple *et al.* (2007) and our approach in Section 3.8.2. The same features described there are used in these baselines as well. These are good features for baselines since they are easily computed from the annotated data.

The event- features correspond to attributes of EVENT elements, with the exception of the event-string feature, which takes as value the character data inside the corresponding TimeML EVENT element. In a similar spirit, the timex3- features are taken from the attributes of TIMEX3 elements with the same name. The *order* features are the attributes computed from the document's textual content. The feature order-event-first encodes whether in the text the event term precedes the time expression it is related to by the temporal relation to classify. The classifier feature order-event-between describes whether any other event is mentioned in the text between the two expressions for the entities that are in the temporal relation, and similarly order-timex3-between is about whether there is an intervening temporal expression. Finally, order-adjacent is true if and only if both order-timex3-between and order-event-between are false (even if other words occur between the expressions denoting the two entities in the temporal relation).

One difference between the baseline models and the models described in Section 3.8.2 is that the final sets of features employed in the classifiers used in Section 3.8.2 are the same as the ones used by Hepple *et al.* (2007) for English: since the point was to compare classifier performance on the two data sets, the same features were used. That is, the feature sets employed are the ones reported by Hepple *et al.* (2007) and optimized for English. The feature sets in these baselines are, in turn, optimized for Portuguese.

More specifically, we tried all possible combinations of these features. The resulting classifiers are evaluated using 10-fold cross-validation on the training data. The the best classifier was kept as the baseline, for the rest of the work reported in this chapter. This operation is performed for each algorithm. Table 4.1 shows

	Task		
Attribute	Task A	Task B	Task C
event-class	d×rkns	×jrk×s	djrkns
event-stem	×jrk××	××r×n×	××××××
event-aspect	××rk×s	××××n×	××rkn×
event-tense	××rkn×	djrkns	djrkns
event-polarity	d×××ns	××××n×	××r×n×
event-pos	××r××s	×××k××	×j××ns
event-string	××××ns	××××××	××××××
order-adjacent	××××n×	N/A	N/A
order-event-first	djrkns	N/A	N/A
order-event-between	djrkns	N/A	N/A
order-timex3-between	×jrk×s	N/A	N/A
timex3-mod	××r×ns	×××k××	N/A
timex3-type	d×rk×s	××rk××	N/A

Table 4.1: Feature combinations used in the baseline classifiers. Features inspired by the ones used by Hepple *et al.* (2007) in TempEval. Key: d means the feature is used with DecisionTable; j, with J48; r with JRip; k, with KStar; n, with NaiveBayes and s with SMO.

the sets of features that yield these best results and are employed in the baseline classifiers.

Table 4.2 presents the evaluation results for the best feature combination and for each task and algorithm, using 10-fold cross-validation. data.

The results in Table 4.2 are better than the results in Table 3.7, in Section 3.8.2, because in the former case feature selection was performed with the Portuguese data, whereas in the latter the combination of features used was the same as the one used for English by Hepple *et al.* (2007), although the initial set of features is identical.

These are the classifiers that will be used for the comparison with the additional features to be tried. As mentioned before in Chapter 3, the data used are organized in a training set and an evaluation set. The training part is around 60,000 words long, the test data containing around 9,000 words. When tested on the held-out test data, these six classifiers present the scores in Table 4.3. These scores will also be compared at the end.

These baselines are easily reproducible: they are based on freely available software, and the features that are employed are easily computed from the annotated

	Task		
Classifier	Task A	Task B	Task C
DecisionTable	55.5	79.3	52.2
J48	57.1	79.7	55.6
JRip	53.8	78.8	52.7
KStar	58.3	79.5	56.8
NaiveBayes	57.5	80.2	54.2
SMO	57.2	79.8	57.0
Majority class baseline	49.4	62.4	41.8

Table 4.2: Performance of the baseline classifiers on the training data, using 10-fold cross-validation on the training data

data, with no need to run any natural language processing tools whatsoever (or any other tool).

A few comments on the selected features are in order. Task A Event-Timex seems to benefit from some of the order- features considerably, as they are present in the optimal feature set of every classifier for this task. Task A Event-Timex is about temporal relations between events and times mentioned in the same sentence. When they are mentioned close enough in the text, it is often the case that the time expression is syntactically dependent on the event term, in which case the temporal relation is very frequently OVERLAP. In some other cases, these two entities are mentioned in the same sentence very far apart from each other, and the temporal relation between them is more indirect, and it is often not OVERLAP.

For Task B Event-DocTime and Task C Event-Event, verb tense seems to be a very important classifier feature. For Task B Event-DocTime, it is the only feature that is present in the best feature combinations of all algorithms. This is expected, since this task is about relating an event with the document creation time, and verb tense locates the event denoted by a verb relative to the speech time, which is the same as the document creation time. For Task C Event-Event, the information carried in the class attribute of EVENT elements, encoded in the event-class feature, is also useful. Task C Event-Event is about temporal relations between the main events of two consecutive sentences. The feature class distinguishes, among other things, between states and other types of situations (see Section 3.3.1 and Section 2.2.2). It is often claimed in the literature that, in narratives with a simple linear structure

	Task		
Classifier	Task A	Task B	Task C
DecisionTable	53.3	77.0	50.0
J48	57.4	77.0	52.7
JRip	61.0	73.7	52.3
KStar	53.9	73.4	53.1
NaiveBayes	50.3	75.5	53.1
SMO	55.6	76.4	53.9
Majority class baseline	59.2	56.2	47.3

Table 4.3: Performance of the baseline classifiers on the test data

and comprising sentences in the past tense, non-stative event sentences move the action forward in time, while the state sentences do not; instead they describe how things are at the time of the last-mentioned event (Hinrichs, 1986; Kamp & Rohrer, 1983; Lascarides & Asher, 1993; Partee, 1984). For this reason, a state appearing as the second event is expected to go with overlap relations more than a non state.

4.3 Natural Language Processing Tools Used

Several natural language tools are necessary to extract information conveyed by the features employed in this work to explore the problem of temporal relation classification. They include a morphological analyzer and a part-of-speech tagger, a constituency parser and a dependency parser. These tools are described in this section. For the sake of reproducibility, a copy of TimeBankPT annotated with these tools is available at http://nlx.di.fc.ul.pt/~fcosta/thesis.

4.3.1 Morphological Analysis

LX-Suite (Barreto *et al.*, 2006; Branco & Silva, 2006; Silva, 2007) splits a text into paragraphs and sentences, splits sentences into words and then annotates each word with its lemma (i.e. its dictionary form), part-of-speech (whether it is a noun, verb, adjective, etc.), and inflectional morphology (gender and number for nouns and adjectives, person, number and tense for verbs, etc.). It additionally recognizes multi-word names.

Figure 4.1 shows the morphological annotation produced at this stage for an example sentence occurring in a document input to the system. In that figure, the topmost box contains the raw text. The middle box shows the direct output of LX-Suite. The box at the bottom contains the output of LX-Suite, converted to an XML format in such a way that the removal of all XML tags results in the original, unannotated text. This is convenient for alignment purposes, as explained below.

This last representation is the one that is used in subsequent phases. Sentences are enclosed in s tags. Words are associated with w elements and annotated with: their dictionary form (the lemma attribute), their part-of-speech (pos), and their inflectional morphology (morph). There is also a numeric identifier, useful for further processing and debugging (the id attribute).

In the output of LX-Suite, punctuation marks are represented as separate word tokens, and contractions are split up into their composing elements. For instance, the contracted forms *do* and *da* in Figure 4.1 are separated in *de* “of” and *o* or *a* “the.” The parts-of-speech annotated in that figure are: preposition (PREP), name (PNM), punctuation (PNT), adverb (ADV), definite article (DA), verb (V), common noun (CN), adjective (ADJ), relative pronoun (REL).

This tool is not completely error free (notice the name TWA800 in Figure 4.1 annotated as an adjective), but the error rates are very low and state-of-the-art for this sort of tool. For instance, the part-of-speech tagger has an accuracy of 96.87% (Branco & Silva, 2006).

Because two sources of annotations are often needed in combination—the original TimeML annotations and the annotations provided by natural language tools such as the just mentioned LX-Suite—it is necessary to combine the two groups of annotations.

The challenge here is that one cannot simply send the annotated data to LX-Suite, as it has no way of knowing what is an annotation and what is linguistic material. Additionally, LX-Suite changes the input text when it splits sentences into words: by separating punctuation and splitting contractions, the number of word tokens, as defined by whitespace, is different between its input and its output.

Therefore, the linguistic material in the two annotated formats need to be aligned somehow. The approach used is to convert the LX-Suite output, shown in the middle box in Figure 4.1, into an XML format like the one in the bottom box

Em Washington, hoje, a Federal Aviation Administration publicou gravações do controlo de tráfego aéreo da noite em que o voo TWA800 caiu.

```
<s> Em/PREP[O] Washington/PNM[B-LOC] ,*//PNT[O] hoje/ADV[O] ,*//PNT[O]
a/DA#fs[O] Federal/PNM[B-ORG] Aviation/PNM[I-ORG] Administration/PNM[I-ORG]
publicou/PUBLICAR/V#ppi-3s[O] gravações/GRAVAÇÃO/CN#fp[O] de_/PREP[O]
o/DA#ms[O] controlo/CONTROLO/CN#ms[O] de/PREP[O]
tráfego/TRÁFEGO/CN#ms[O] aéreo/AÉREO/ADJ#ms[O] de_/PREP[O] a/DA#fs[O]
noite/NOITE/CN#fs[O] em/PREP[O] que/REL[O] o/DA#ms[O] voo/VOO/CN#ms[O]
TWA800/TWA800/ADJ#ms[O] caiu/CAIR/V#ppi-3s[O] .*//PNT[O] </s>
```

```
<s><w id="3" lemma="Em" pos="PREP">Em</w> <w id="4" lemma="Washington"
pos="PNM">Washington</w><w id="5" lemma="," pos="PNT">,</w> <w id="6"
lemma="hoje" pos="ADV">hoje</w><w id="7" lemma="," pos="PNT">,</w> <w
id="8" lemma="a" pos="DA" morph="fs">a</w> <w id="9" lemma="Federal"
pos="PNM">Federal</w> <w id="10" lemma="Aviation" pos="PNM">Aviation</w>
<w id="11" lemma="Administration" pos="PNM">Administration</w> <w id="13"
lemma="PUBLICAR" pos="V" morph="ppi-3s">publicou</w> <w id="14"
lemma="GRAVAÇÃO" pos="CN" morph="fp">gravações</w> <c><w id="16"
lemma="de" pos="PREP" surface="de"/><w id="17" lemma="o" pos="DA" morph="ms"
surface="o"/><cs>do</cs></c> <w id="19" lemma="CONTROLO" pos="CN"
morph="ms">controlo</w> <w id="20" lemma="de" pos="PREP">de</w> <w id="21"
lemma="TRÁFEGO" pos="CN" morph="ms">tráfego</w> <w id="22" lemma="AÉREO"
pos="ADJ" morph="ms">aéreo</w> <c><w id="24" lemma="de" pos="PREP"
surface="de"/><w id="25" lemma="a" pos="DA" morph="fs"
surface="a"/><cs>da</cs></c> <w id="27" lemma="NOITE" pos="CN"
morph="fs">noite</w> <w id="28" lemma="em" pos="PREP">em</w> <w id="29"
lemma="que" pos="REL">que</w> <w id="30" lemma="o" pos="DA"
morph="ms">o</w> <w id="31" lemma="VOO" pos="CN" morph="ms">voo</w> <w
id="32" lemma="TWA800" pos="ADJ" morph="ms">TWA800</w> <w id="34"
lemma="CAIR" pos="V" morph="ppi-3s">caiu</w><w id="35" lemma="."
pos="PNT">.</w></s>
```

Figure 4.1: Morphological annotation of raw input. The sentence translates to English as *In Washington today, the Federal Aviation Administration released air traffic control tapes from the night the TWA Flight eight hundred went down.*

of that figure. This format has the property that if one removes all the XML tags, the original text is obtained. For alignment purposes with the TempEval annotations, this characteristic is important because TimeML also has this property. As a result, alignment can be performed by looking at character positions, ignoring the annotations.

This is how the two kinds of annotations are combined. Figure 4.2 shows the TimeML annotation for the sentence in Figure 4.1 (top box) and the result of combining it with the automatic morphological annotation (bottom box).

4.3.2 LX-Parser and LX-DepParser

In some of the work reported below, syntactic information is used. This information is derived from two parsers: LX-Parser (Silva *et al.*, 2010), a constituency parser, and LX-DepParser (Reis, 2010), a dependency parser.

Figure 4.3 shows a syntactic tree produced by LX-Parser for a sentence that is a shorter version of this chapter's working example, so it can fit the page. The actual output format of LX-Parser is a bracketed representation of a tree, as shown in Figure 4.4.

LX-Parser is based on the Stanford parser of Klein & Manning (2003). It was trained for Portuguese with mostly news articles. Under the Parseval metric it achieves an F-measure of 88% (value obtained through 10-fold cross-evaluation).

LX-DepParser produces dependency graphs for input sentences. An example can be seen in Figure 4.5. Once again, the parser's output is actually textual. More specifically, it follows the CoNNL format. It is organized in columns and rows, with each row representing a word, and each column a specific piece of information relating to that word. A slightly abridged example, where columns irrelevant to the present discussion were eliminated, can be seen in Figure 4.6. There, the leftmost column contains a numeric identifier for a word. The second column shows the surface form of the word as it occurs in the text. The third, fourth and fifth columns contain properties of the word identified by LX-Suite: respectively its lemma, part-of-speech and inflection tag. The last two columns describe the dependency graph. The sixth column contains the identifier of the word that the current word depends on, and the last column shows the name of the dependency relation. The main verb

```
<s>Em Washington, <TIMEX3 tid="t53" type="DATE" value="1998-01-14"
temporalFunction="true" functionInDocument="NONE"
anchorTimeID="t52">hoje</TIMEX3>, a Federal Aviation Administration <EVENT
eid="e1" class="OCCURRENCE" stem="publicar" aspect="NONE" tense="PPI"
polarity="POS" pos="VERB">publicou</EVENT> gravações do controlo de tráfego
aéreo da <TIMEX3 tid="t54" type="TIME" value="1998-XX-XXTNI"
temporalFunction="true" functionInDocument="NONE"
anchorTimeID="t52">noite</TIMEX3> em que o voo TWA800 <EVENT eid="e2"
class="OCCURRENCE" stem="cair" aspect="NONE" tense="PPI" polarity="POS"
pos="VERB">caiu</EVENT>.</s>
```

```
<s><w pos="PREP">Em</w> <w pos="PNM">Washington</w><w
pos="PNT">,</w> <TIMEX3 tid="t53" type="DATE" value="1998-01-14"
temporalFunction="true" functionInDocument="NONE" anchorTimeID="t52"><w
pos="ADV">hoje</w></TIMEX3><w pos="PNT">,</w> <w pos="DA"
morph="fs">a</w> <w pos="PNM">Federal</w> <w pos="PNM">Aviation</w> <w
pos="PNM">Administration</w> <EVENT eid="e1" class="OCCURRENCE"
stem="publicar" aspect="NONE" tense="PPI" polarity="POS" pos="VERB"><w pos="V"
lemma="PUBLICAR" morph="ppi-3s">publicou</w></EVENT> <w pos="CN"
lemma="GRAVAÇÃO" morph="fp">gravações</w> <c><w pos="PREP"
surface="de"/><w pos="DA" morph="ms" surface="o"/><cs>do</cs></c> <w
pos="CN" lemma="CONTROLO" morph="ms">controlo</w> <w pos="PREP">de</w>
<w pos="CN" lemma="TRÁFEGO" morph="ms">tráfego</w> <w pos="ADJ"
lemma="AÉREO" morph="ms">aéreo</w> <c><w pos="PREP" surface="de"/><w
pos="DA" morph="fs" surface="a"/><cs>da</cs></c> <TIMEX3 tid="t54" type="TIME"
value="1998-XX-XXTNI" temporalFunction="true" functionInDocument="NONE"
anchorTimeID="t52"><w pos="CN" lemma="NOITE" morph="fs">noite</w></TIMEX3>
<w pos="PREP">em</w> <w pos="REL">que</w> <w pos="DA"
morph="ms">o</w> <w pos="CN" lemma="VOO" morph="ms">voo</w> <w pos="ADJ"
lemma="TWA800" morph="ms">TWA800</w> <EVENT eid="e2" class="OCCURRENCE"
stem="cair" aspect="NONE" tense="PPI" polarity="POS" pos="VERB"><w pos="V"
lemma="CAIR" morph="ppi-3s">caiu</w></EVENT><w pos="PNT">.</w></s>
```

Figure 4.2: TimeML and morphological annotation

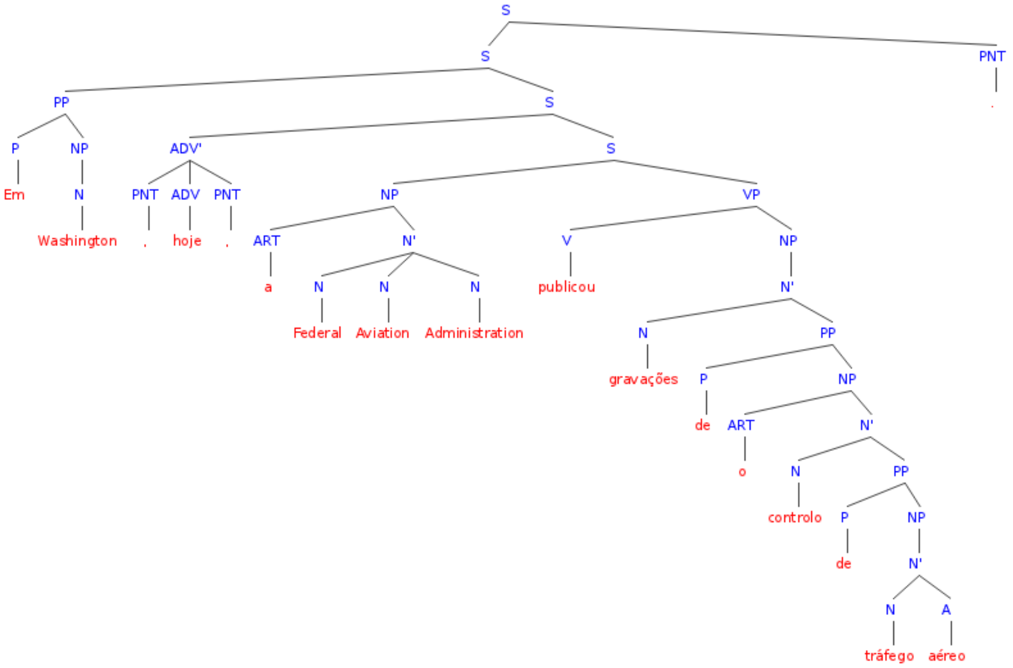


Figure 4.3: Example parse tree produced by LX-Parser. The sentence translates to English as *In Washington today, the Federal Aviation Administration released air traffic control tapes.*

```
(S (S (PP (P    (Em))
          (NP   (N    (Washington))))
     (S   (ADV' (PNT  (,))
                (ADV  (hoje))
                (PNT  (,)))
          (S    (NP   (ART  (a))
                      (N'   (N  (Federal))
                            (N  (Aviation))
                            (N  (Administration))))
                (VP   (V     (publicou))
                      (NP   (N' (N   (gravações))
                                (PP (P    (de))
                                    (NP  (ART (o))
                                         (N'    (N  (controlo))
                                                (PP (P   (de))
                                                    (NP (N' (N (tráfego)))
                                                            (A (aéreo))))))))))))))
```

Figure 4.4: Example output of LX-Parser corresponding to the tree in Figure 4.3. The sentence translates to English as *In Washington today, the Federal Aviation Administration released air traffic control tapes.*

is represented as depending on word 0 (which does not exist), with the dependency relation being ROOT.

LX-DepParser was developed based on the MSTParser (McDonald *et al.*, 2005) and trained on the same corpus as LX-Parser. Its accuracy is 86.8%.

As can be seen from these examples, the two parsers sometimes produce results that say different things. For instance, the dependency representation corresponding to the syntactic tree for this sentence would have the word *hoje* "today" depending on the main verb form *publicou* "released", since the structure in Figure 4.3 is meant to indicate that that adverbial is a modifier of a syntactic constituent headed by this verb. Instead, the dependency parser wrongly says it is a modifier of the preposition phrase *em Washington* "in Washington." Therefore, if syntactic information is to be explored in the context of temporal relation classification (or any other problem), the choice of parser can produce different results.

The representations produced by these parsers are aligned with the word tokens coming from LX-Suite, so no additional alignment efforts are required.

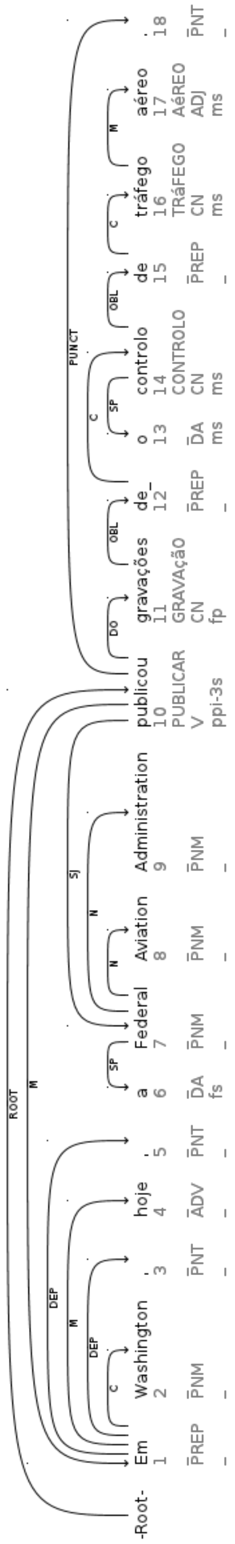


Figure 4.5: Example dependency graph produced by LX-DepParser

```
1  Em             _          PREP _      10 M
2  Washington     _          PNM  _      1  C
3  ,              _          PNT  _      1  DEP
4  hoje           _          ADV  _      1  M
5  ,              _          PNT  _      1  DEP
6  a              _          DA   fs     7  SP
7  Federal        _          PNM  _      10 SJ
8  Aviation       _          PNM  _      7  N
9  Administration _          PNM  _      7  N
10 publicou       PUBLICAR   V    ppi-3s 0  ROOT
11 gravações      GRAVAÇÃO   CN   fp     10 DO
12 de             _          PREP _      11 OBL
13 o              _          DA   ms     14 SP
14 controlo       CONTROLO   CN   ms     12 C
15 de             _          PREP _      14 OBL
16 tráfego        TRÁFEGO    CN   ms     15 C
17 aéreo          AÉREO      ADJ  ms     16 M
18 .              _          PNT  _      10 PUNCT
```

Figure 4.6: Abridged output of LX-DepParser

4.4 Classifier Features

The main ideia in this Chapter is that different levels of information are required for temporal relation classification. This Section describes various kinds of additional features deemed to be interesting for this problem. These new features range from features taken from shallow processing tools (Section 4.4.1) to features obtained with natural language tools that provide richer information (Section 4.4.5) or other kinds of linguistic information that are difficult to obtain from tools but can be approximated by compiling values for them off-line (Section 4.4.2). Some also resort to extra-linguistic information, such as logic (Section 4.4.4) and knowledge about the world (Section 4.4.3)

4.4.1 Shallow Processing

One interesting question is to what extent information coming from part-of-speech taggers and morphological analyzers can help the problem at hand. That is, before trying more elaborate solutions, we want to assess the impact of shallow natural language technology. Shallow tools do not provide structural information, like parsers do, but they are typically more efficient and accurate.

It must be noted here that the TimeML annotations already encode some information about part-of-speech and morphology. In particular, the attributes stem, pos, tense and aspect of TimeML EVENT elements carry the information that such tools provide. Namely, the attribute stem carries the lemma, or dictionary form, of the tagged event, the pos attribute its part-of-speech, the tense attribute its tense (or the value none if the event term is not a verb), and its aspect is in the attribute aspect.

However, the words in the text that do not denote events are not similarly tagged. In some cases this sort of information might help the classification. For instance, since many prepositions and conjunctions carry temporal information (e.g. words like *before*, *after*, etc.), their presence can be a useful heuristic. To get at this we might use a part-of-speech tagger to extract prepositions and conjunctions from the vicinity of the temporal expressions and events that enter the temporal relations that are to be classified.

Additionally, there are other pieces of information provided by the shallow processing tools that are not available in the annotations.

This section explores this sort of simple approach. To this end, we implement several attributes that take advantage of the information provided by these shallow tools. They are briefly described next.

Subject Agreement In TimeML, co-referent event terms are annotated as denoting temporally overlapping events. For this reason, event co-reference resolution should help the classification of temporal relations, specially Task C Event-Event. Indeed, the two tasks have been considered to be related in the literature (Bejan & Harabagiu, 2010). For instance, Quine (1985) and Davidson (1985) consider that two events are identical if they happen in the same place and at the same time and have the same participants.

Therefore, Task C Event-Event may benefit from features that are heuristics to detect co-referent events. Subject-verb agreement, as manifested in verbal inflection, can be a cue to identify co-referent event terms when they are verbs. If two verbs describe the same event, their subject will also be co-referent. Because of that, the two subjects will likely share person and number features. That is not necessary—for instance, a singular noun phrase headed by a collective noun can be co-referent with a plural noun phrase—but the expectation is that it is frequently so.

Since most events are denoted by verbs, this cue may affect many instances.

For Task C Event-Event, an additional feature called events-equal-subject-agreement checks whether the inflectional features with respect to subject agreement of the terms for the two elements in the temporal relation are identical.

The conjunction closest to the event (in the same sentence) A number of the temporal relations for Task A Event-Timex relate events and times that are denoted by expressions that are not syntactically related. Instead, the temporal expression is modifying some other event term. The temporal relation to classify can thus only be determined by considering two other temporal relations: the one between this temporal expression and the event that it modifies; and the one between the two events. One way to identify this relation between the two events is to look for conjunctions in the sentence, as these often signal specific temporal relations between the situations described in the two clauses that they join.

We use a feature whose value is the surface form of the conjunction that is closest to the event that is the first argument of the temporal relation to classify. Distance here is the number of words between the two words (the conjunction and the event term). In order to find the conjunction closest to the event term, we look for conjunctions on both sides of the event term until one is found or a sentence boundary is hit.

The following example illustrates why such a feature may be useful:

(22) Mas de qualquer forma as empresas estrangeiras compraram apenas um pequeno número de empresas japonesas **este ano**, enquanto que as empresas japonesas **adquiriram** centenas de companhias estrangeiras.
But by all accounts foreign companies have bought only a relative handful of Japanese companies ***this year****, while Japanese companies have* ***acquired*** *hundreds of foreign companies.*

In this example *enquanto que* "while" signals an overlap temporal relation between the events in the two clauses (the buying event and the acquiring event). This can be used as a cue for the temporal overlap relation between the entities denoted by the expressions highlighted in boldface.

Lemmas of words in the temporal expressions The events and times ordered in the temporal relations for Task A Event-Timex can be denoted by words and expressions that are not syntactically related, as we have just seen. These cases are usually difficult to classify. Sometimes, however, the time expression and the event term provide enough information to make a strong guess at what the temporal relation is. Consider the example, taken from the training data:

(23) “Não vamos **passar** de um certo nível”, disse David N. McCammon, vice-presidente financeiro da Ford, numa conferência de imprensa **ontem**, em Dearborn, no Michigan.
*“We will not **go** over a certain level,” said David N. McCammon, Ford’s vice president for finance, at a news conference **yesterday** in Dearborn, Mich.*

The highlighted verb occurs in a future construction (*vamos passar*/ will not go), and the highlighted temporal expression *ontem* “yesterday” refers to a past date. The date can thus only precede the event, and this is indeed what is annotated. Since tense is already a classifier feature, we add another feature that can capture facts such as that of the temporal expression referring to past or future times.

This feature’s value is based on the temporal expression that denotes the time related by the temporal relation to classify. The value is computed like this: (i) each word in the temporal expression is replaced by its lemma, according to the morphological analyzer; (ii) each of them is removed unless it is equal (ignoring case) to one of a small list of lemmas. The lemmas in this list are of words that have some temporal content and are often seen in the temporal expressions seen in the training data. This list includes *ainda* “still”, *amanhã* “tomorrow”, *anterior* “previous”, *anteriormente* “previously”, *atual* “current”, *breve* “soon”, *brevemente* “soon”, *cada* “each”, *corrente* “current”, *this* “este”, *futuro* “future”, *haver* “ago”, *hoje* “today”, *já* “already” , *passado* “past”, *posterior* “following”, *presente* “present”, *próximo* “next”, *seguinte* “next”, *todo* “every”, *recent* “recent”, *recentemente* “recently”, *último* “last”.

This feature enables the classification algorithms to learn that a time expression with *ontem* "yesterday" or *passado* "past" should refer to a time that precedes present and future events, or that a time expression with *amanhã* "tomorrow" should denote a date that follows a past or present event.

The preposition preceding the temporal expression Temporal expressions, as annotated by the TimeML specifications, are often noun phrases. In many cases they are the complement of a preposition, which in some cases conveys temporal meaning. For instance:

(24) E nas grandes corretoras, após **dez anos** de crescimento, **fala**-se de demissões.
*And at the big brokerage houses, after **ten years** of boom, they're **talking** about layoffs.*

In this example the preposition *após* "after" immediately before the temporal expression is a strong indicator of the temporal relation between the event and the time period described by the highlighted words.

A classifier feature timex3-preposition is employed which has as its value the preposition that immediately precedes the temporal expression in the text, or the value NONE if that word is not a preposition or the time expressions appears at the beginning of a sentence.

4.4.2 Aspectual Type

There are several reasons to think aspectual type, as presented in Section 2.2.2 is relevant to temporal information processing, as aspect and tense are deeply related.

Motivation First, these distinctions are related to how long events last: culminations are punctual, whereas states can be very prolonged in time. States are thus more likely to temporally overlap other temporal entities than culminations, for instance.

Second, there are grammatical consequences on how events are anchored in time. Consider the following examples, from Ritchie (1979) and Moens & Steedman (1988), already mentioned in Section 2.2.2:

(25) a. When they built the 59^{th} Street bridge, they used the best materials.

b. When they built that bridge, I was still a young lad.

The situation of building the bridge is a culminated processed, composed by the process of actively building a bridge followed by the culmination of the bridge being finished. In sentence (25a), the event described in the main clause (that of using the best materials) is a process, but in sentence (25b) it is a state (the state of being a young lad). Even though the two clauses in each sentence are connected by *when*, the temporal relations holding between the events of each clause are different. On the one hand, in sentence (25a) the event of using the best materials (a process) overlaps with the process of actively building the bridge and precedes the culmination of finishing the bridge. On the other hand, in sentence (25b) the event of being a young lad (which is a state) overlaps with both the process of actively building the bridge and the culmination of the bridge being built. This difference is arguably caused by the different aspectual types of the main events of each sentence.

As another example, states overlap with temporal location adverbials, as in (26a), while culminations are included in them, as in (26b).

(26) a. He was happy last Monday.

b. He reached the top of Mount Everest last Monday.

In other cases, differences in aspectual type can disambiguate ambiguous linguistic material. For instance, the preposition *in* is ambiguous as it can be used to locate events in the future but also to measure the duration of culminated processes; it is thus ambiguous with culminated processes, as in *He will read the book in three days* but not with other aspectual types, as in *He will be living there in three days.*

A factor related to aspectual class, that is not trivial to account for, is the phenomenon of aspectual shift, or aspectual coercion (de Swart, 1998a, 2000; Moens & Steedman, 1988). Many linguistic contexts pose constraints on aspectual type. This does not mean, however, that clashes of aspectual type cause ungrammaticality. What often happens is that phrases associated with an incompatible aspectual type get their type changed in order to be of the required type, causing a change in meaning.

For instance, the progressive construction combines with processes. When it combines with e.g. a culminated process, the culmination is stripped off from this culminated process, which is thus converted into a process. The result is that a sentence like (27a), with a progressive construction, does not say that the bridge was finished (the event has no culmination), whereas one such as (27b) does say this (the event has a culmination).

(27) a. They were building that bridge.
b. They built that bridge.

Aspectual type is not a property of just words, but phrases as well. For example, while the progressive construction just mentioned combines with processes, the resulting phrase behaves as a state (cf. the sentence *When they built the 59th Street bridge, they were using the best materials* and what was mentioned above about *when* clauses).

Limitation Naturally, we would like to evaluate the impact of this kind of information on these tasks. The TimeML annotations already include an attribute class for EVENTs that encodes some aspectual information, distinguishing between stative (annotated with the value STATE) and non-stative events (value OCCURRENCE). This attribute is relevant to the classification problem at hand, i.e. it is a useful feature for machine learned classifiers for the TempEval tasks, as shown above in Table 4.1 (although this class attribute encodes other kinds of information as well, as described in Section 3.3.1). However, aspectual distinctions can be more fine-grained than a mere binary distinction, and so far no system has explored this sort of information to help improve the solutions to temporal relation classification. Here, we assume the four-way distinction presented in Section 2.2.2, between states, processes, culminated processes and culminations.

Ideally, a feature would be available to these classifiers, encoding the aspectual type of the event in the temporal relation. This feature would have four possible values, reflecting these four aspectual types. This information is not present in the TimeML annotations, so it must be extracted from another source. No existing tool for Portuguese provides it, either. Our approach is to extract this information in an unsupervised way, as that is the fastest way to obtain this information.

Strategy Aspectual type is hard to annotate. This is partly because of what was just mentioned: it is not a property of just words, but rather phrases, and different phrases with the same head word can have different aspectual types; however annotation schemes like TimeML annotate the head word as denoting events, not full phrases or clauses.

For this reason, our strategy is to obtain aspectual type information from unannotated data. Because these data are gradient—an event-denoting word can be associated with different aspectual types, depending on word sense and on syntactic context—we do not aim to extract categorical information, but rather numeric values for each event term that reflect associations to aspectual types. These may be seen as values that are indicative of the frequencies in which an event term denotes a state, or a process, etc.

In order to extract these indicators, we resort to a methodology sometimes referred to as Google Hits: large amounts of queries are sent to a web search engine (not necessarily Google), and the number of search results (the number of web pages that match the query) is recorded and taken as a measure of the frequency of the queried expression.

This methodology is not perfect, since multiple occurrences of the queried expression in the same web page are not reflected in the hit count, and in many cases the hit counts reported by search engines are just estimates and might not be very accurate. Additionally, carelessly formulated queries can match expressions that are syntactically and semantically very different from what was intended. In any case, it has the advantages of being based on a very large amount of data and not requiring any manual annotation, which can introduce errors.

The Web as a Very Large Corpus Hearst (1992) is one of the earliest studies where specific textual patterns are used to extract lexico-semantic information from very large corpora. The author's goal was to extract hyponymy relations between words. With the same goal, Kozareva *et al.* (2008) apply similar textual patterns to the web.

The web has been used as a corpus by many other authors with the purpose of extracting syntactic or semantic properties of words or relations between them, e.g. Ravichandran & Hovy (2002), Etzioni *et al.* (2004), etc. Some of this work is specially relevant to the problem of temporal information processing. VerbOcean (Chklovski & Pantel, 2004) is a database of web mined relations between verbs. Among other kinds of relations, it includes typical precedence relations, e.g. *sleeping* happens before *waking up*. This type of information has in fact been used by some of the participating systems of TempEval-2 (Ha *et al.*, 2010), with good results.

More generally, there is a large body of work focusing on lexical acquisition from corpora. Just as an example, Mayol *et al.* (2005) learn subcategorization frames of verbs from large amounts of data. Relevant to our work is that of Siegel & McKeown (2000). The authors guess the aspectual type of verbs by searching for specific patterns in a one million word corpus that has been syntactically parsed. They extract several linguistic indicators and combine them with machine learning algorithms. The indicators that they extract are naturally different from ours, since they have access to syntactic structure and we do not, but our data are based on a much larger corpus.

Textual Patterns as Indicators of Aspectual Type Because of aspectual shift phenomena (see Section 4.4.2), full syntactic parsing is necessary in order to determine the aspectual type of a natural language expression. However, this aspectual type can be approximated by frequencies: it is natural to expect that e.g. stative verbs occur more frequently in stative contexts than non-stative verbs, even if there may be errors in determining these contexts if syntactic parsing is not a possibility.

If one uses Google Hits, syntactic information is not accessible. In return for its impreciseness, the Google Hits methodology has the advantage of producing results based on a very large body of data.

We try this approach focusing exclusively on verbs, even though events can be denoted by words belonging to other parts-of-speech. This limitation is linked to the fact that the textual patterns that are used to search for specific aspectual contexts are sensitive to part-of-speech (i.e. what may work for a verb may not work equally well for a noun).

Extracting the Aspectual Indicators We extracted the 4,000 most common verbs from a 180 million word corpus of Portuguese newspaper text, CETEM-

Público.[1] This corpus contains approximately 180 million words of text taken from the newspaper *O Público*.

Because this corpus is not annotated, we used LX-Suite (see Section 4.3.1), a part-of-speech tagger and morphological analyzer, to detect verbs and to obtain their dictionary form. We then used a verbal inflection tool (Branco *et al.*, 2009) to generate the specific verb forms that are used in the queries. They are mostly third person singular forms of several different tenses.

The indicators that we used are ratios of Google Hits. They compare two queries.

Several indicators were tested. We provide examples with the verb *fazer* "do" for the queries being compared by each indicator. The name of each indicator reflects the aspectual type being tested, i.e. states should present high values for State Indicators 1 and 2, processes should show high values for Process Indicators 1–4, etc. The indicators are:

- State Indicator 1 (the classifier feature event-indicator-st1) is about imperfective and perfective past forms of verbs. It compares the number of hits a for an imperfective form *fazia* "did" to the number of hits b for a perfective form *fez* "did": $\frac{a}{a+b}$. Assuming the imperfective past constrains the entire clause to be a state, and the perfective past constrains it to be telic, the higher this value the more frequently the verb appears in stative clauses in a past tense.[2]

- State Indicator 2 (event-indicator-st2) is about the co-occurrence with *acaba de* "has just finished". It compares the number of hits a for *acaba de fazer* "has just finished doing" to the number of hits b for *fazer* "to do": $\frac{b}{a+b}$. In Portuguese, this construction does not seem to be felicitous with states.

- Process Indicator 1 (event-indicator-pc1) is about past progressive forms and simple past forms (both imperfective). It compares the number of hits a

[1] http://www.linguateca.pt/CETEMPublico

[2] We expect this frequency to be indicative of states because states can appear in the imperfective past tense with their interpretation unchanged, whereas non-stative events have their interpretation shifted to a stative one in that context (e.g. they get a habitual reading). In order to refer to an event occurring in the past with an on-going interpretation, non-stative verbs require the progressive construction to be used in Portuguese, whereas states do not. Therefore, states should occur more freely in the simple imperfective past.

for *fazia* "did" to the number of hits b for *estava a fazer* "was doing": $\frac{b}{a+b}$. Assuming the progressive construction is a function from processes to states (see Section 4.4.2), the higher this value, the more likely the verb can occur with the interpretation of a process.

- Process Indicator 2 (event-indicator-pc2) is about past progressive forms vs. simple past forms (perfective). It compares the number of hits a for *fez* "did" to the number of hits b for *esteve a fazer* "was doing": $\frac{b}{a+b}$. Similarly to the previous indicator, this one tests the frequency of a verb appearing in a context typical of processes.

- Process Indicator 3 (event-indicator-pc3) is about the occurrence of *for* Adverbials. It compares the number of hits a for *fez* "did" to the number of hits b for *fez durante muito tempo* "did for a long time": $\frac{b}{a+b}$. This number is also intended to be an indication of how frequent a verb can be used with the interpretation of a process. Note that Portuguese allows modifiers to occur freely between a verb and its complements, so this test should work for transitive verbs (or any other subcategorization frame involving complements), not just intransitive ones.

- Process Indicator 4 (event-indicator-pc4) is about the co-occurrence of a verb with *parar de* "to stop". It compares the number of hits a for *parou de fazer* "stopped doing" to the number of hits b for *fazer* "to do": $\frac{a}{a+b}$. Just like the English verbs *stop* and *finish* are sensitive to the aspectual type of their complement, so is the Portuguese verb *parar*, which selects for processes.

- Atelicity Indicator 1 (event-indicator-at1) is about comparing *in* and *for* adverbials. It compares the number of hits a for *fez num instante* "did in an instant" to the number of hits b for *fez durante muito tempo* "did for a long time": $\frac{b}{a+b}$. Processes can be modified by *for* adverbials, whereas culminated processes are modified by *in* adverbials. This indicator tests the occurrence of a verb in contexts that require these aspectual types.

- Atelicity Indicator 2 (event-indicator-at2) is about comparing *for* Adverbials with *suddenly*. It compares the number of hits a for *fez de repente* "did suddenly" to the number of hits b for *fez durante muito tempo* "did for a long time": $\frac{b}{a+b}$. *De repente* "suddenly" seems to modify culminations, so this indicator compares process readings with culmination readings.

- Culmination Indicator 1 (event-indicator-cm1) is about differentiating culminations and culminated processes. It compares the number of hits a for *fez de repente* "did suddenly" to the number of hits b for *fez num instante* "did in an instant": $\frac{a}{a+b}$.

For each of the 4,000 verbs, the necessary queries required by these indicators were generated and then sent to a search engine. The queries were enclosed in quotes, so as to guarantee exact matches. The number of hits was recorded for each query.

We had some problems with outliers for a few rather infrequent verbs. These could show very extreme values in the ratios supporting some indicators. In order to minimize their impact, we homogenized the 100 highest values that were found for each indicator. More specifically, each one of the highest 100 values that each indicator shows was replaced by its 100^{th} highest value. The bottom 100 values were similarly changed. This way the top 99 values and the bottom 99 values are discarded and replaced by the 100^{th} highest value and the 100^{th} lowest value respectively.

Each indicator ranges between 0 and 1 in theory. In practice, we seldom find values close to the extremes, as this would imply that some queries would have close to 0 hits, which does not occur very often (after all, we intentionally used queries for which we would expect large hit counts, as these are more likely to be representative of true language use). For this reason, each indicator is scaled so that its minimum (actual) value is 0 and its maximum (actual) value is 1.

The Aspectual Indicators as Classifier Features Each of these indicators is incorporated as a classifier feature, whose value is taken from the indicator. For the State Indicator 1 of the event in the temporal relation to classify, we use the feature event-indicator-st1. Similarly, for the remaining indicators we use the classifier features: event-indicator-st2, event-indicator-pc1, event-indicator-pc2, event-indicator-pc3, event-indicator-pc4, event-indicator-at1, event-indicator-at2, event--indicator-cm1. For Task C Event-Event, we also employ features that encode the aspectual indicators for the second event in the relation.

4.4.3 Knowledge about the World

This section explores information about the world that can be useful to temporal relation classification. In particular, some temporal relations between events are more expected than others just due to lexical semantics. The isolated meaning of the words involved may provide temporal clues that are worth exploring.

Temporal Direction Consider the two following examples, from the point of view of classifying the temporal relations between the events and times that are highlighted in boldface in each sentence (i.e. Task A Event-Timex):

(28) a. Os analistas **previam** [que em **1990** a BellSouth visse lucros na casa dos 3,90 dólares por ação.]
*Analysts were **predicting** [that in **1990** BellSouth would see earnings in the range of $3.90 a share.]*

b. A N.V. DSM **informou** [que o lucro líquido no **terceiro trimestre** subiu 63%.]
*N.V. DSM **said** [net income in **the third quarter** jumped 63%].*

In the case of the example in (28a), the fact that the temporal expression occurs in the complement of this verb (enclosed in square brackets) is a good indication that the event precedes the date, because of what the verb means: predictions are made before what is predicted happens, and since what is predicted is the 1990 BellSouth earnings, the *predict* event should have occurred earlier than 1990 (or at least earlier than the time at which these earning are announced).

In (28b), with the verb *informar* “inform, say”, we find the inverse temporal relation. Here, reporting events are expected to temporally follow the reported events. In this sentence, there is an annotated temporal relation between the event denoted by the term *informou* “said” and the time expression *o terceiro trimestre* “the third quarter.” This time expression locates the time of the event described in the embedded clause (inside brackets) in the timeline. The annotated temporal relation is thus dependent on the temporal relation between the two events. Since that is a temporal relation between a reporting event and a reported event, the expectation is that the reporting event temporally follows the reported event.

The idea is to record this sort of information in a feature for the classifiers. Although the classifiers do not know what the complement of the verb is, they do now that in these two examples the verb precedes the temporal expression (because of the classifier feature order-event-first presented above in Section 4.2), which can be regarded as a hint that the temporal expression occurs inside the complement of the verb (which is the case in these two examples), as complements follow their heads in a language like Portuguese.

We call this sort of temporal information between a word and its complement "temporal direction", for lack of a better expression.

In order to obtain this information, all event lemmas present in the training data were extracted and a mapping was manually created between them and the expected temporal relation with its complement. For many of these words the associated value is NONE, since they impose no temporal constraint with respect to the material mentioned in their complement. The other possible values are AFTER and BEFORE. A few examples:

- *acusar* "accuse, charge" AFTER
- *atrasar* "stall, delay" BEFORE
- *organizar* "organize" BEFORE
- *prever* "predict" BEFORE
- *relatar* "report,post" AFTER
- *tentar* "try, seek, attempt" BEFORE

This feature thus records knowledge about the world. According to the examples provided, events of *accusing* follow the events that someone is accused of doing, events of *delaying* precede delayed events, events of *organizing* precede organized events, *reporting* events follow reported events, and *trying* events precede tried events. Appendix III shows the full list of manually annotated lemmas of the event words found in the training data.

This annotated information is not evaluated independently, and some error is likely. Rather, a classifier feature is employed, with these values, related to the lemma of the event that is the first argument of the temporal relation to be classified, in the hope that it will be useful despite its imperfections.

It must be mentioned that this manual annotation was performed without looking at contexts where the words occur, but rather by just taking into account what one would expect to see in the data, based on the word. The justification for that choice is so that the resulting mapping does not overfit the training data.

It must be once again stressed that the classifiers do not know that e.g. in (28b) *lucros* "earnings" is the syntactic complement of the verb. They only know that the event denoted by the verb probably precedes the event denoted by whatever event is mentioned in the verb's complement. Other features can, however, provide clues for this syntactic relation, like the feature order-event-first already mentioned, although this is just a hint. This is another limitation of this feature.

Even though this feature records expected temporal relations between events, it should be useful for Task A Event-Timex, as a means to classify temporal relations in those cases where the time in that relation is given by an expression that is not a syntactic dependent of the word denoting the event in the temporal relation, but rather modifies another event denoting word or phrase in the appropriate syntactic relation with the other event term, as the example in (28a) above. Based on classifier performance, this feature does seem to be somewhat useful for the problem of temporal relation classification (Section 4.5.2).

4.4.4 Temporal Deduction

The problem of temporally ordering events and times is constrained by the logical properties of temporal relations. For instance, temporal precedence is a strict partial order. Therefore, it is natural to incorporate logical information in the solutions to the problem of ordering events and time intervals. Perhaps surprisingly, this idea has not been explored by many authors in the context of temporal relation classification. Part of the reason might be that most systems participating in the TempEval competitions were based on general machine learning algorithms, most of which do not naturally combine with logical constraints.

```
<TIMEX3 tid="t190" type="TIME" value="1998-02-06T22:19:00"
functionInDocument="CREATION_TIME">02/06/1998 22:19:00</TIMEX3>
<s>WASHINGTON _ The economy<EVENT eid="e1">created</EVENT>jobs at a
surprisingly robust pace in <TIMEX3 tid="t191" type="DATE"
value="1998-01">January</TIMEX3>, the government<EVENT
eid="e4">reported</EVENT> on <TIMEX3 tid="t193" type="DATE"
value="1998-02-06">Friday</TIMEX3>, evidence that America's economic stamina
has <EVENT eid="e6">withstood</EVENT> any <EVENT
eid="e7">disruptions</EVENT> <EVENT eid="e224">caused</EVENT> so far by
the financial <EVENT eid="e228">tumult</EVENT> in Asia.</s>
<TLINK lid="l1" relType="OVERLAP" eventID="e4" relatedToTime="t193" task="A"/>
<TLINK lid="l2" relType="AFTER" eventID="e4" relatedToTime="t191" task="A"/>
<TLINK lid="l26" relType="BEFORE" eventID="e4" relatedToTime="t190" task="B"/>
```

Figure 4.7: Example (simplified) temporal annotations for the fragment: *WASHINGTON _ The economy created jobs at a surprisingly robust pace in January, the government reported on Friday, evidence that America's economic stamina has withstood any disruptions caused so far by the financial tumult in Asia.*

Motivation The motivation for using logical information as a means to help solving this problem can be illustrated with the sample text in Figure 4.7, taken from the TempEval training data.

There, we can see that the date 1998-02-06, denoted by the expression *Friday*, includes the document's creation time, which is 1998-02-06T22:19:00. We know this from comparing the normalized value of these two expressions, annotated in the value attribute of TIMEX3 elements. From the annotated temporal relation with the lid l26 (the last one in the figure) we also know that the event identified with e4, denoted by the form *reported*, precedes the document's creation time.

From these two facts one can conclude that this event either precedes the time denoted by *Friday* or they overlap; this time cannot however precede this event. That is, the possible relation type for the relation represented with the TLINK named l1 is constrained—it cannot be AFTER.

What this means is that, in this example, solving Task B Event-DocTime can, at least partially, solve Task A Event-Timex. The information obtained by solving Task B Event-DocTime can be utilized in order to improve the solutions for Task A Event-Timex. Similar examples can be found for other combinations of tasks where these dependencies can be seen.

Feature Design The problem of temporal relation classification benefits from temporal reasoning, because, when a system annotates raw text, it may split the annotation process in several steps, corresponding to the different TempEval tasks. In this scenario, the information annotated in previous steps can be used. For instance, if a system has already classified the temporal relations between the events in a text and its creation time (Task B Event-DocTime, which is also the easiest), this information can then be used to help classify the remaining temporal relations.

Our approach here is to incorporate in these classifiers new features for the three tasks. These new features are meant to help predict the class feature by computing the temporal closure of a set of initial temporal relations. This initial set of temporal relations is composed of relations coming from two sources:

- Temporal relations between pairs of dates or times corresponding to annotated temporal expressions. Because the annotations for time expressions contain a normalized representation of them, it is possible to order them symbolically. That is, they are ordered according to the value attribute of the corresponding TIMEX3 element.

- The temporal relations annotated for the other tasks.

For the sake of experimentation, all combinations of tasks were tried:

- Predict Task A Event-Timex after temporally closing the relations annotated for Task B Event-DocTime and Task C Event-Event (and the temporal relations between the times mentioned in the document);

- Similarly, predict Task B Event-DocTime, based on the temporal relations for Task A Event-Timex and Task C Event-Event and those between dates and times;

- Predict Task C Event-Event after temporally closing the relations annotated for Task A Event-Timex and Task B Event-DocTime and those between dates and times.

After some experimentation, the best strategy seems to consist in first solving Task B Event-DocTime, taking into account the temporal relations between the dates and times in the document, then solving Task A Event-Timex (considering the answers for Task B Event-DocTime as well as the temporal relations between times) and finally addressing Task C Event-Event (taking advantage of the previously identified temporal relations for the other tasks and those between times).

Ordering times and dates As a first step, the times and dates mentioned in a document are ordered according to their normalized value. For instance, the date 2000-01-03 is ordered as preceding the date 2010-03-04. We used Joda-Time 2.0 (http://joda-time.sourceforge.net), which already implements some of the functionality that is required. Since all temporal expressions are normalized in the annotated data, we order temporal expressions before applying any temporal reasoning. This increases the number of temporal relations we start with, and the potential number of relations we end up with after reasoning.

First, each annotated date or time is converted into a time interval. In many cases it is possible to specify the start and end points of this interval. For instance, the date 2000-01-03 is represented internally by an interval with its start point at 2000-01-03T00:00:00.000 and ending at 2000-01-03T23:59:59.999. Many different kinds of normalized expressions require many rules. For instance, an expression like *last Winter* could be annotated in the data as 2010-WI, and dedicated rules are used to get its start and end points.

Some time expressions are normalized as PRESENT_REF (e.g. *now*), PAST_REF (*the past*) or FUTURE_REF (*the future*). These cases are not represented by any Joda-Time object. Instead we need to account for them in a special way. They can be temporally ordered among themselves (e.g. PRESENT_REF precedes FUTURE_REF), but not with other temporal expressions. We further stipulate that PRESENT_REF includes each document's creation time (which therefore precedes FUTURE_REF, etc.). So, in additional to the representation of times and dates as time intervals, we employ a layer of *ad-hoc* rules.

A full account of the rules implemented in order to order times and dates is provided in Appendix IV.

Temporal Reasoning Rules The rules implemented in the reasoning component are:

- Temporal precedence is transitive, irreflexive and antisymmetric;
- Temporal overlap is reflexive and symmetric;
- If A does not precede B, then either B precedes A or A and B overlap;
- If A overlaps B and B precedes C, then C does not precede A.

Because we also consider temporal relations between times and dates, we also deal with temporal inclusion, a type of temporal relation that is not part of the annotations used in the TempEval data, but that is still useful for reasoning. We make use of the following additional rules, dealing with temporal inclusion:

- Temporal inclusion is transitive, reflexive and antisymmetric;
- If A includes B, then A and B overlap;
- If A includes B and C overlaps B, then C overlaps A;
- If A includes B and C precedes A, then C precedes B;
- If A includes B and A precedes C, then B precedes C;
- If A includes B and C precedes B, then either C precedes A or A and C overlap (A cannot precede C);
- If A includes B and B precedes C, then either A precedes C or A and C overlap (C cannot precede A).

Features Used The values for these features reflect the possible values of the class feature (i.e. the temporal relation being classified), after applying temporal reasoning to these two sets of relations. The possible values for these classifier features are the six class values (BEFORE, AFTER, OVERLAP, BEFORE-OR-OVERLAP, OVERLAP-OR-AFTER and VAGUE). It must be noted that the values BEFORE-OR--OVERLAP or OVERLAP-OR-AFTER are output when none of the three more specific values (BEFORE, OVERLAP and AFTER) can be identified by the temporal reasoner but one of them can be excluded (i.e. OVERLAP-OR-AFTER is used when BEFORE can be excluded). Similarly, VAGUE is output when no constraint can be identified from the initial set of temporal relations. These underspecified values do not necessarily correspond to the cases when the annotated data contain these values (those are the cases when the human annotators could not agree on a more specific value). It often is the case that the human annotation is more specific, as humans have access to further information.

The classifier features employed are:

- The classifier feature B-for-A takes as values the possible type of the temporal relations for Task A Event-Timex based on the temporal closure of the relations annotated for Task B Event-DocTime and the temporal relations between the times mentioned in the document.

- The feature AB-for-C similarly tries to classify the temporal relations for Task C Event-Event after temporally closing the relations annotated for Task A Event-Timex and Task B Event-DocTime and those between times.

The usefulness of these classifier features is limited in that they have very good precision but low recall, as temporal reasoning is unable to restrict the possible type of temporal relation for many instances. In fact, trying to predict the type of Task A Event-Timex temporal relations on the basis of the temporal relations annotated for Task C Event-Event and those between the times mentioned in the document produces the VAGUE value for all instances in the training data. This is also the case when using Task C Event-Event to predict Task B Event-DocTime. But even these two features, B-for-A and AB-for-C, show a very high number of training instances with the value VAGUE: 79% and 93% respectively.

For this reason, another set of features is used that, instead of trying to predict the class value directly, can provide useful heuristics to the classifiers. These are:

- For Task B Event-DocTime, the feature timexes-majority-for-B looks at all annotated temporal expressions in the same sentence as the event being related to the document creation time (DCT) and takes as value the majority temporal relation between those temporal expressions and the DCT, based on their annotated value attributes;

- Also for Task B Event-DocTime, the feature timexes-closest-for-B has as its value the temporal relation between the DCT and the time expression closest to the event being ordered with the DCT;

- For Task A Event-Timex, the feature closure-vague-B-for-A takes as its value a "vague" (this vagueness is explained below) temporal relation based on the relations annotated for Task B Event-DocTime and the temporal relations between the times/dates mentioned in the document;

- For Task C Event-Event, the feature closure-vague-AB-for-C has as its value a "vague" temporal relation for Task C Event-Event based on the relations annotated for Task A Event-Timex and Task B Event-DocTime and the temporal relations between the times/dates mentioned in the document.

These temporal relations that we call vague are useful when the reasoning component does not identify a precise temporal relation between the two relevant entities in the temporal relation (due to insufficient information). In these cases, it may be interesting to known that e.g. both of them temporally overlap a third one, as this may provide some evidence to the classifiers that they are likely to overlap. This sort of information is what these vague features encode. Their possible values are:

- BOTH-FOLLOW-A-THIRD-ONE: a third entity precedes the two entities;

- BOTH-OVERLAP-A-THIRD-ONE: a third entity overlaps both entities;

- BOTH-PRECEDE-A-THIRD-ONE: a third entity follows the two entities;

- BOTH-OVERLAP-AND-FOLLOW-A-THIRD-ONE: a third entity overlaps both entities and a third entity follows both entities;

- UNCONNECTED: the two entities are not even connected in the temporal graph for the document (this is a graphic whose nodes correspond to events and times and whose edges correspond to overlap and precedence relations);

- UNRELATED: none of the above.

Related Work The work of Allen (1983) and Allen (1984), presented in Chapter 2, paved the way for a vast literature on automated temporal reasoning. Since then, research on this topic has been concerned with problems such as computational efficiency and completeness (e.g. Vilain *et al.* (1990) or Tsang (1987)).

As mentioned in Section 2.9, temporal reasoning has been used in several recent efforts related to temporal information annotation (Katz & Arosio, 2001; Setzer & Gaizauskas, 2001; Verhagen, 2005) and processing (Bramsen *et al.*, 2006; Chambers & Jurafsky, 2008a; Denis & Muller, 2011; Ha *et al.*, 2010; Ling & Weld, 2010; Mani *et al.*, 2006; UzZaman & Allen, 2010; Yoshikawa *et al.*, 2009). Additionally, one participant of the first TempEval used “world-knowledge axioms” as part of a symbolic solution to this challenge (Puşcaşu, 2007). This world-knowledge component includes rules for reasoning about time. These approaches are similar to what we explore here in that they use knowledge about one TempEval task to help solve the other tasks. However, these studies do not report on the full set of logical constraints used or explore little information (e.g. the transitivity of temporal precedence only). Our work does not have these shortcomings. Closest to our work is that of Tatu & Srikanth (2008). The authors employ information about Task B Event-DocTime and temporal reasoning as a source of classifier features for Task C Event-Event only. This is more limited than our approach: we also explore the other tasks as sources of knowledge, besides Task B Event-DocTime, and we also experiment with solutions for the other tasks, not just Task C Event-Event.

4.4.5 Parsing

Syntactic information can be used to constrain the possible temporal relations between the various entities mentioned in a text. The sentences making up the documents in the data to process can be analyzed by parsers like those referred in Section 4.3.2. The parsers produce representations that include information about the way in which words are combined in these sentences. This can be used by handmade rules to detect the temporal relations to be classified, or at least to constrain the possible values. This information can then be incorporated in a classifier feature. This section describes an approach to implement this idea.

Motivation Task A Event-Timex is about classifying temporal relations between entities denoted by words and expressions occurring in the same sentence. In many cases these words or phrases are not directly related syntactically. Rather, they can be arbitrarily distant, since the only criterion for including them in Task A Event-Timex is that they occur in the same sentence. This was changed in the second TempEval, where the corresponding task did not consider such cases, but in the data for the first TempEval—and accordingly in TimeBankPT—many temporal relations annotated for this task correspond to these more difficult cases where the two entities in the temporal relation are denoted by words or phrases that are far apart from each other in the sentence where they occur.

As a consequence, for Task A Event-Timex, many temporal relations can only be determined by looking at the syntactic structure of the sentence where the elements denoting the arguments of those relations occur. The example in (29a) and its Portuguese equivalent in (29b), from the training data of TimeBankPT, illustrate this case:

(29) a. Soviet Foreign Ministry spokesman Yuri Gremitskikh said special ambassador Mikhail Sytenko left Tuesday for consultations with the governments of Syria, Jordan, Egypt and other Arab countries.

b. O porta-voz do Ministério dos Negócios Estrangeiros soviético Yuri Gremitskikh disse que o embaixador especial Mikhail Sytenko partiu terça-feira para consultar os governos da Síria, Jordânia, Egito e outros países árabes.

In this sentence, there is an annotated temporal relation between the event denoted by *said* and the date denoted by *Tuesday*, and the event is temporally located at a time that is after the date. If this temporal expression *Tuesday* were a modifier of this verb *said*, the temporal relation would be one of overlap (cf. the case of *left* and *Tuesday* in this sentence).

One way to correctly arrive at this AFTER value for the temporal relation is to take linguistic and logic information into account:

- The event denoted by *left* and the date for *Tuesday* temporally overlap, because *Tuesday* is a modifier of *left*.

- The event denoted by *said* is after the event denoted by *left*, because of the tenses employed and because *left* is the head verb of the clause that is the complement of *said*. In such a syntactic configuration with the verb *say* and when the two verbs involved are in the simple past tense (as in this case), the meaning of the sentence constrains the temporal relation between the two events in this way.[1]

- If the event denoted by *said* is temporally located at a time that follows the time in which the event denoted by *left* is located, and this time of the leaving event temporally overlaps the date denoted by *Tuesday*, it follows that the event denoted by *said* cannot be located at a time that precedes the date denoted by *Tuesday*: either it follows this date, or they overlap.

In order to arrive at such a conclusion, several ingredients are needed:

- Morphological information, so that we know the verb tenses, among other things;

- Syntactic information, i.e. a parser, so that we know that e.g. *Tuesday* modifies *left* and that this verb heads the clause that is the complement of *said*;

- An analysis of time phenomena, i.e. a set of temporal interpretation rules that explicitly state e.g. that the date denoted by a time expression like *Tuesday* overlaps the event time of the verb that this expression modifies, and that the relation between the events denoted by the two verbs is one of temporal precedence, based on their tense and the syntactic configuration in which they occur;

- A reasoning component, so that we can infer additional temporal relations from the temporal relations identified by this temporal interpretation module,

[1] In English the possibility of overlap also exists in such cases (cf. *He **said** he was **sick***). In the Portuguese version of the data, the tense on the embedded verb form *partiu* ("left"), the *pretérito perfeito* (a perfective past), does force a temporal precedence reading. That is, the imperfective past tense *pretérito imperfeito* allows an overlap reading (***Disse** que **estava** doente "He said he was sick"*), but the perfective past does not, even with stative verbs (***Disse** que **esteve** doente* "He said he was sick (before)").

namely that the event denoted by *said* cannot precede the date denoted by *Tuesday*.

With these ingredients, it is possible to predict the temporal relation being classified by looking at the grammatical information present in the sentence where the entities being related occur, or at least constrain the possible types of this temporal relation, as in this example, where we can exclude the BEFORE value.

The morphological information comes from LX-Suite, presented above in Section 4.3.1. In order to get information about syntax, a parser can be used. Here the parsers described in Section 4.3.2, LX-Parser and LX-DepParser, are used. The reasoning component used for the logic features described in Section 4.4.4 is used here as well. What is specifically needed here is an analysis of time phenomena, which can be viewed as a set of rules that use grammatical information as the main means to extract temporal relations from text. We refer to this component here as the TimeDecorator, and it is described below.

Details of the approach We want to test two parsers that produce different kinds of output. LX-Parser is a constituency parser and as such outputs phrase structure representations. LX-DepParser is a dependency parser and delivers dependency graphs.

The TimeDecorator works on the output of the parsers. A common format for the output of the two parsers would be useful, so that the TimeDecorator can employ the same set of rules no matter which parser is used. For this reason, we developed a module that maps the two types of representation into a single type of representation. We refer to these unified representations as grammatical representations.

Note that the idea is not to merge the output of the two parsers. Each parser is used independently, but we want to test both of them. Each will give rise to a separate classifier feature that tries to predict the type of the temporal relations. Each of these two features is computed by taking the output of one of the parsers and applying a set of hand-made interpretation rules to it (i.e., the TimeDecorator). In order to be able to use the same interpretation module, the output of the parsers is independently adapted: when the input text is parsed with LX-Parser, the resulting phrase structure representation is transformed into a grammatical representation, which is then fed to the TimeDecorator; when the input text is parsed with

LX-DepParser, the resulting dependency graph is transformed into a grammatical representation, which is then sent to the same TimeDecorator.

Grammatical representations These grammatical representations are essentially dependency graphs that use a small inventory of dependency relations. These representations also contain the information coming from LX-Suite (see Section 4.3.1).

More specifically, these simplified dependency graphs are restricted to the following types of dependency relations:

- SUBJ relations between a head and the head of its subject;
- COMP/MOD or COMP relations between a head and the head of one of its complements or modifiers
- CONJ relations for coordinations: the relation between the head of the first coordinand and the head of the other coordinands or the relation between the head of the first coordinand and the conjunction used;
- SPEC relations between a head and the head of its specifiers;
- a ROOT relation identifies the main verb of a sentence.

The reason why no distinction is made between complements and modifiers is because it was observed that the parsers make many mistakes when discriminating between the two. When the dependent element is a noun phrase or a complementizer phrase, it is almost always a complement and there are few errors. For these cases the dependency relation COMP is used. When its syntactic category is different, it was observed that the distinction was not reliably made by the parsers, and COMP/MOD is used instead.

The output of LX-DepParser is straightforward to convert into this grammatical representation, as the latter is a simplified version of the former, where some distinctions are neutralized. In order to convert the output of LX-Parser into these representations, it is necessary to:

- Find the head of a phrase, as these dependency relations are between words. For instance, the head of a noun phrase is the noun. If that noun phrase occurs as the subject of a sentence, there must be a SUBJ dependency between that noun and the main verb of that sentence.

- Find syntactic functions based on the phrase structure. For instance, if an S (sentence) node has two daughter nodes and one of them is labeled VP (verb phrase), the other daughter node is the subject of that sentence.

These two operations are performed by a set of handcrafted rules. They are not particularly challenging to implement, and the remainder of this discussion assumes these operations are performed correctly.

Figure 4.8 shows an example of these grammatical representations, ommitting the part-of-speech and morphological information assigned to each word. There, each line refers to a word, whose surface form is inside quotation marks. Each relation label (in capitals) names the relation between the word in its line and the word in the lowest line that is one indentation level above it. So, for instance, there is a SPEC relation between *o* ("the") and *porta-voz* ("spokesman") in this example. Inside parentheses we show the identifiers for events and temporal expressions of the corresponding TimeML annotations (the TimeML annotations are aligned with these representations). In italics we also include the English translation of each word, for clarity (the English translation is not actually implemented). The actual representation is not a textual representation (as shown in this figure), but it rather consists of Java objects (one for each line in this figure) connected in various ways. The figure is just one way of seeing how they are organized. These objects are also connected with other objects that represent morphological annotations and others for the TimeML annotations. It is also worth mentioning that the word order in the original text is not represented in Figure 4.8, although the implementation makes this information available.

Temporal constraints from dependency relations As mentioned above, this module encapsulating the temporal interpretation rules is called the TimeDecorator. The TimeDecorator implements a set of rules that aim to extract temporal relations from the text, taking into account the following kinds of information: (i) these simplified dependency relations obtained from the parsers; (ii) the annotations of LX-Suite (see Section 4.3.1), which tags each word with its part-of-speech, its lemma, and its inflection features; (iii) some lexical information relevant to time, which is explained below; (iv) the TimeML annotations describing events and temporal expressions.

```
ROOT "disse" (e112)                      said
   SUBJ "porta-voz"                      spokesman
      SPEC "o"                           the
      ...
   COMP/MOD "que"                        that
      COMP/MOD "partiu" (e113)           left
         SUBJ "embaixador"               ambassador
            SPEC "o"                     the
            ...
         COMP/MOD "terça-feira" (t114)   Tuesday
         COMP/MOD "para"                 to
            COMP/MOD "consultar" (e116)  consult
               COMP/MOD "governos"       governments
                  SPEC "os"              the
                  ...
```

Figure 4.8: Example grammatical representation for *O porta-voz (. . .) disse que o embaixador (. . .) partiu terça-feira para consultar os governos (. . .)* "The spokesman said that the ambassador left Tuesday to consult the governments."

The basic mechanism underlying the TimeDecorator is to associate temporal indices to words in the grammatical representation and to keep a set of temporal relations between these indices. These indices can be seen as denoting time intervals.

We start from the head word of a sentence (the main verb, identified by the ROOT dependency relation) and assign it a temporal index. After that, for each of its dependents we assign another temporal index. The temporal index can be identical to that of the head word (if for instance the two words denote events or dates that happen at the same time), or it can be a different one, in which case a temporal relation between the two indices can be added to the representation. This choice is determined by the implemented rules. The process of assigning temporal indices to the immediate dependents of a head is recursive. In the end, all words end up with an index.

Identical temporal indices represent the same time interval. When a temporal index is assigned to a dependent word and it is different from the temporal index of the head, a temporal relation is added between that index and the index of the head (i.e. between the two time intervals).

Three types of temporal relations between these indices are used: temporal precedence, temporal overlap, temporal inclusion. Precedence and overlap are disjoint,

```
ROOT "disse" (e112) [t193]                              said
  SUBJ "porta-voz" [t193]                               spokesman
    SPEC "o" [t193]                                     the
    ...
  COMP/MOD "que" [t202]                                 that
    COMP/MOD "partiu" (e113) [t202]                     left
      SUBJ "embaixador" [t202]                          ambassador
        SPEC "o" [t202]                                 the
        ...
      COMP/MOD "terça-feira" (t114) [t207]              Tuesday
      COMP/MOD "para" [t208]                            to
        COMP/MOD "consultar" (e116) [t208]              consult
          COMP/MOD "governos" [t208]                    governments
            SPEC "os" [t208]                            the
            ...
precedes  =  {..., < t193, t202>, ...}
 includes  =  {..., < t207, t202 >, ...}
```

Figure 4.9: Grammatical representation decorated with temporal indices and temporal relations

and inclusion is a special case of overlap.

We show the decorations produced for the sentences in (29), repeated here in (30):

(30) a. Soviet Foreign Ministry spokesman Yuri Gremitskikh said special ambassador Mikhail Sytenko left Tuesday for consultations with the governments of Syria, Jordan, Egypt and other Arab countries

b. O porta-voz do Ministério dos Negócios Estrangeiros soviético Yuri Gremitskikh disse que o embaixador especial Mikhail Sytenko partiu terça-feira para consultar os governos da Síria, Jordânia, Egito e outros países árabes.

Figure 4.9 shows this extended representation, for the same sentence. Inside the square brackets we find the temporal index associated with the word. Finally, below the grammatical representation we can find the temporal relations involving these indices, which are stored extensionally.

For words belonging to temporal expressions, the temporal index can be seen as representing the time or date or duration that that temporal expression refers to.

For event denoting words, it can be seen as the event time.[1] The remaining words are still decorated with these indices, even though they do not bear any direct relation to them. In these later cases, the indices are usually identical to a higher index. The rules that decorate words with temporal indices and add temporal relations between these indices are local, in the sense that they can look at the temporal index of the word on which the current word immediately depends, but they do not look at arbitrarily distant words in the representation. By passing these temporal indices from the head to a dependent even when that dependent has no temporal meaning, it is easier to keep these rules local in this sense. This mechanism is thus similar to the feature percolation employed by several grammatical formalisms, such as HPSG (whose contribution to the problem of temporal information extraction is explored in Chapter 5).

As an example, some of the rules for assigning an index to a subject are:

- If the head is a verb whose lemma belongs to the list *causar* ("cause"), *levar* ("take"), *provocar* ("cause"), *resultar* ("result") and *terminar* ("finish"), the head of the subject is assigned a new index that precedes that of the head. This (together with rules that constrain the index of the complement of these verbs appropriately) is intended to cover cases like the hypothetical example *The accident caused a wave of protests*, where the event denoted by *accident* precedes the events denoted by *protests*.

- If the head is a verb whose lemma is *seguir* ("follow"), the head of subject is given an index that temporally follows the index of the head.

There are quite a few rules such as these implemented in the TimeDecorator. As a rough estimate of its size, the implementation comprises around 2,600 lines of Java code with around 340 `if` statements.

These rules are the result of looking at the training data and implementing rules that would cover the cases found there. We stopped creating rules when no more progress could be measured. At some point adding more rules to cover extra data creates errors elsewhere. In large part this is because of parser errors.

[1] We assume they are always time intervals (never points or instants) and thus they can always appear as either argument of relations like inclusion.

Once all words in a TimeML-annotated document are decorated, temporal closure is performed on the temporal relations between the temporal indices determined by the TimeDecorator.

This mechanism discovers additional temporal relations between the various words and phrases in a document. Because these words are aligned with the annotated events and temporal expressions, the TimeDecorator can be used to discover the temporal relations annotated in TimeML.

There are advantages and disadvantages of this approach when compared with the methods employed thus far. The main advantage is that the TimeDecorator has access to structural information, which is arguably necessary to determine several instances of the temporal relations annotated in this corpus (the motivating example above in (29) is a case in point). The disadvantages are that it is a slow process, it depends on information coming from parsers, which have error rates higher than shallower natural processing tools, like part-of-speech taggers, and its usefulness is limited for Task C Event-Event: parsers process each sentence in isolation.

In order to make this approach usable in Task C Event-Event, before starting the decoration process, we combine the grammatical representations for all sentences in a document (the parsers analyze each sentence in isolation) as though they were all coordinated with *and*, i.e. as though the different sentences were coordinate clauses of a single sentence. The decoration process then begins by assigning a temporal index to the main verb of the first sentence. After that, the head of each of its immediate dependents is also decorated with an index. Because all the sentences in a document are now related by a CONJ dependency relation, the decoration process is able to find temporal relations between elements in different sentences. For instance, when the TimeDecorator finds a coordination of verb-headed elements all in the perfective past tense, it assigns to these verbs different temporal indices and adds a temporal precedence relation between the index of the first verb mentioned and that of the second one. This accounts for examples like (4a), shown in Chapter 1 and repeated here in (31).

(31) *Kim came in. Sue left.*

This example illustrates a true limitation of this approach: in case of multiple possibilities, the TimeDecorator only implements one, based on quantitative data

	Task A		Task B		Task C	
	Train	Test	Train	Test	Train	Test
Constituency parser (LX-Parser)						
Recall	0.42	0.47	0.59	0.58	0.18	0.17
Precision	0.72	0.71	0.81	0.85	0.48	0.46
Dependency parser (LX-DepParser)						
Recall	0.35	0.39	0.51	0.45	0.22	0.21
Precision	0.67	0.65	0.78	0.83	0.41	0.37

Table 4.4: Precision and recall of the TimeDecorator at predicting the type of the temporal relations annotated in TimeBankPT. These results are broken down according to the syntax parser used to produce the original grammatical representations on which the TimeDecorator operates.

from the training data. Because of this, wrong results are possible. This would be the case for the example (4f) in Chapter 1 and repeated here in (32). The same rule would apply here and predict the wrong ordering between the two mentioned events. The TimeDecorator does not have any information about causality, which would be required to handle such an example correctly.

(32) *Kim fell down. Sue pushed him.*

Because of Task B Event-DocTime, we also reserve one temporal index for the document creation time (DCT). When the TimeDecorator finds a verb, it adds a temporal relation between the index for the DCT and the temporal index for that verb, mostly based on the grammatical tense of the verb.

On its own, the TimeDecorator has high precision (when it detects a temporal relation, it is often correct), but low recall (it does not recognize many of the temporal relations). In Table 4.4, these figures are shown, both for the training data, which was consulted for its development, and the unseen test data. It can be seen that the precision of the TimeDecorator for Task A Event-Timex is above 0.70, and it can reach 0.85 for Task B Event-DocTime. For Task C Event-Event the results are, however, quite weak.

The low recall for all tasks is due to the fact that the TimeDecorator looks mostly at grammatical information. Because of this, it leaves many annotated temporal relations undetected, as other sources of information are required to discover them.

The TimeDecorator generalizes well. As the implemented rules are handcrafted, one ends up with general rules instead of rules that overfit the training data. This can be seen by looking at the performance of the TimeDecorator on the unseen test data. It is usually not much worse than its performance on the training data, and in some cases (Task B Event-DocTime) it is even better.

Error Analysis and Limitations As mentioned before, the low recall is due to the limited sources of information used by the TimeDecorator. There are other factors that come into play. For instance, in many cases more inferences are needed. Consider the example in (33a), with the corresponding Portuguese sentence in (33b), taken from the training data.

(33) a. In Washington today, the Federal Aviation Administration released air traffic control tapes from the night the TWA Flight eight hundred went down.

b. Em Washington hoje, a Federal Aviation Administration publicou gravações do controlo de tráfego aéreo da noite em que o voo TWA800 caiu.

In this example, the event denoted by *went* is annotated as preceding the time denoted by *today*. The way for an approach similar to the TimeDecorator to find that out is to consider that this event happens at a time included in the time associated with the mentioned night and then to infer that this night precedes the time denoted by *today*. The annotated temporal relation then follows from these two facts.

The first fact can be discovered by the TimeDecorator if the parser produces a correct analysis for this sentence. It is particularly easy in the Portuguese example (where it is literally *the night in which...*): the noun *noite* “night” is modified by a relative clause headed by the verb form *caiu* “went down” and the fronted relative constituent *em que* “in which” includes the preposition *em* “in”, which can be associated with this inclusion relation between the two indices for these two elements (*noite* and *caiu*).

The problem is finding the second fact—that the mentioned night precedes the day referred to by *today*. This is not extractable from the annotations, because the time expression that refers to this night is annotated with the normalized value 1998-XX-XXTNI, where XX represents missing values for the month and day fields of that date, as they are not possible to derive from the textual content of the document (even by the human annotators). To find this precedence relation, verb tense can be a cue: since *went* occurs in the past tense, the mentioned night must be either a past night or the current night (if the sentence is produced at night, which is a possibility). In the English case, the expressions *today* and *the night...* cannot refer to the same stretch of time as the word *today* does not refer to nights. Since *today* refers to the current day, it must follow a past night. One must additionally assume that if this night is being described so verbosely it is not the current night, but a past night instead, i.e. common sense is required. This problem is harder for Portuguese, where *hoje* can mean *today* or *tonight*. In Portuguese the two expressions can in principle refer to the same period. This problem requires explicitly encoding more knowledge about dates and times and the way they are mentioned by these words, and even some common-sense assumptions, which are outside the scope of the TimeDecorator.

Another limitation is parser error. The implemented rules in the TimeDecorator are sometimes less specific than they could be in order to be more robust to parser errors. An example is treating complements and modifiers in the same way, as mentioned above, since parsers have difficulties in discriminating between the two sometimes. The fact is that the kind of text documents present in TimeBankPT (many of these documents are news articles in the domain of economics) is quite different from the data used to train these parsers, which is mostly open domain news articles and some children's literature (Reis, 2010; Silva *et al.*, 2010). These economics texts contain many numbers and constructions involving numbers (e.g. *went up 5%*) that are not common in other kinds of texts and are a source of many errors.

Hybrid approach The low recall of the TimeDecorator means that the overall number of correctly classified temporal relations ends up being much lower than the baselines presented above. Despite its problem with recall, the TimeDecorator can still be useful because we can use its output as a feature of the classifiers developed in this chapter. Because of the high precision of the TimeDecorator, by hypothesis the extended classifiers can show improvements with this feature.

We try two features: predictor-parser and predictor-dep-parser. The only difference between them is that the first one uses the phrase structure parser, LX-Parser, for the initial grammatical representations of the text to decorate with temporal indices, whereas the second one uses the dependency parser, LX-DepParser (see Section 4.3.2).

These two features try to predict the class value directly. The set of values that they can take contains the three basic types of temporal relations annotated in TimeBankPT—OVERLAP, BEFORE and AFTER—as well as additional values that are returned when it is not possible to temporally order the two entities in the target temporal relation. These additional value are:

- EVENT-OVERLAPS-FUTURE (for Task B Event-DocTime only): the event to order with respect to the DCT overlaps a temporal expression whose value attribute has the value FUTURE_REF (one example is a time expression such as *the future*);

- EVENT-OVERLAPS-PAST (for Task B Event-DocTime only): the event to order with respect to the DCT overlaps a temporal expression whose value attribute has the value PAST_REF (one example is a time expression such as *previously*);

- EVENT-OVERLAPS-PRESENT (for Task B Event-DocTime only): the event to order with respect to the DCT overlaps a temporal expression whose value attribute has the value PRESENT_REF (one example is a time expression such as *currently*);

- BEFORE-OR-OVERLAP: when it is possible to rule out the AFTER value but not BEFORE or OVERLAP;

- OVERLAP-OR-AFTER: when it is possible to rule out the BEFORE value but not OVERLAP or AFTER;

- BOTH-INCLUDED-IN-A-THIRD-ONE: there is a third temporally anchored entity (event or time) mentioned in the document such that both entities in the relation are temporally included in it (this can be an indication that these two are temporally close and with some probability overlap);

- BOTH-OVERLAP-A-THIRD-ONE: there is a third temporally anchored entity (event or time) mentioned in the document that both entities in the relation temporally overlap (this can be an indication that these two are temporally close and with some probability overlap);
- BOTH-PRECEDE-A-THIRD-ONE: there is a third temporally anchored entity (event or time) mentioned in the document that both entities in the relation temporally precede (this can be an indication that these two are temporally close and with some probability overlap);
- BOTH-FOLLOW-A-THIRD-ONE: there is a third temporally anchored entity (event or time) mentioned in the document that both entities in the relation temporally follow (this can be an indication that these two are temporally close and with some probability overlap);
- UNRELATED: in the temporal graph for the document (where the nodes are the temporally orderable entities and the edges are the temporal relations between them), there is no path between the nodes for the two entities in the target relation;
- VAGUE when none of the above apply.

If more than one of these apply, the value higher on this list is returned by the TimeDecorator.

4.4.6 Further Exploiting the TimeML Annotations

The TimeML annotations provide information for many more classifier features than the small set of features explored in the baselines (Section 4.2). There are several ways to take advantage of the annotations:

- the values of some of the attributes can be simplified or encoded in different ways that may prove more advantageous;

- there are more annotated elements than those participating in the temporal relation under classification (for instance, other EVENTs or TIMEX3es occurring in the same sentence) that may contain information relevant for this classification;

- for Task C Event-Event, it is possible to compare the features of the two events in the relation. This information can be taken directly from the attributes of these elements, or computed from them just like many of the new features tried are;

- it is possible to take into account various properties relating to the previous temporal relation.

In this section some additional classifier features are described that take advantage of these ideas.

Simplified Tense The annotation decisions for TimeBankPT included the annotation of the feature tense of EVENTs with information about tense and mood (see Section 3.6.1). This creates many possible values for this feature, which can lead to problems of data sparseness, or just make it harder for a classifier to make use of a feature that simply copies the value of this attribute (this is the case of the classifier feature event-tense presented above in Section 4.2). In particular there is a long tail of rarely occurring values, as shown in Figure I.1 in Appendix I.

This is in sharp contrast with the values of the same attribute tense as it is used in the original English data: there it has four possible values: present, past, future and none.

For this reason, it is interesting to check whether an attribute with fewer values can lead to better learned models. To this end, we experimented with a feature whose value is based on the attribute tense of the annotated EVENTs. In this new feature, event-simplified-tense, the possible tense values are mapped to a smaller set of possible values. More specifically, these are the same four values employed in the English corpus (present, past, future and none) and two additional disjunctive values (present_or_future and past_or_present). The mapping is in Appendix II.

This mapping is based on the baseline model induced by J48 for Task B Event-DocTime, presented above in Section 4.2. By inspecting the learned decision tree, one can see which tenses are associated with events that precede, overlap, or follow the document's creation time and associate them with the values past, present and future respectively.

The distribution of the values of this new attribute is in Figure II.1, in Appendix II.

Features based on attributes of other annotated TimeML elements Many temporal relations for Task A Event-Timex relate an event and a time that are far apart in the sentence. In particular, the temporal expression may be closer to another annotated event. Since this other event is annotated with several kinds of information, these can also be used as classifier features.

One piece of information that immediately seems potentially useful is tense. As the previous sections have discussed, sometimes the temporal relation between an event and a time can be hinted at by discovering other temporal relations. For instance, when the time expression modifies another event term in the sentence, the temporal relation between the event in the temporal relation to guess and the event modified by this temporal expression can be useful. Once again, the focus is on Task A Event-Timex.

Tense may be interesting because comparing the tense of two event terms (when they are verbs) may shed light on the temporal relation between the two events: e.g. if the first is in the past tense and the second is in the future tense, the first one probably precedes the second one.

Since other kinds on information are available in the annotations (part-of-speech, lemma, etc.), these are also tried.

There are other situations where looking at other annotated events can be important. Verb tense can be useful even for Task A Event-Timex. For instance, when considering a time expression that unambiguously refers to a future time (e.g. *tomorrow*), which the classifiers can know about through the features in Section 4.4.1, an event given by a verb form in the past tense probably precedes it in time. This information is not available if the event is given by a noun. In this case, the tense of nearby verbs, or nearby annotated events, may be helpful.

These extra events are searched for in the same sentence in the four following ways, and each way produces a group of classifier features. These four distinct ways are easier to understand with a concrete example. Consider the following sentence, taken from the training data, where words in boldface correspond to annotated events:

(34) Ao **lançar** o seu **desafio** final a Saddam Hussein, Bush **manteve** a intensa **diplomacia** pessoal que **começou** após a **invasão** em agosto passado.
In ***setting*** *out his final* ***challenge*** *to Saddam Hussein, Mr. Bush* ***continued*** *the intensive personal* ***diplomacy*** *he* ***began*** *after the* ***invasion*** *last August.*

The first set of features looks at the event term closest to the temporal expression in the text. For example, the instance representing the temporal relation between the event given by *lançar* "setting" and the date denoted by *agosto passado* "last August", contains features describing the event *invasão* "invasão."

The second set of features consider the event term closest to the event that occurs in the temporal relation. For instance, for the temporal relation between *desafio* "challenge" and the same mentioned date, the features in this group pertain to the event denoted by the term *lançar* "setting."

The other two sets of features look for event terms that intervene between the mentions of the two entities in the temporal relation under consideration, i.e. events mentioned in the text in between the words and expressions referring to these entities (events and times).[1] One of these sets of features is for the case when the event in the temporal relation is mentioned in the text before the time is. The other set of features is for the cases when the time is mentioned before the event in the text. In both cases, the annotated event term that is chosen is the one closest to the temporal expressions, if more than one annotated event term can be found in the text in between the mentions to the two entities in the temporal relation. In this example sentence, the instance for the temporal relation between *lançar* "setting" and *agosto passado* "last August" has additional features about the event given by

[1] The existence of these intervening events is an indication that the time in the temporal relation under classification is denoted by an expression that does not modify the term for the event in this relation (the feature order-event-between, one of the features in the baselines (Section 4.2), which provides the same information, seems to be a very powerful feature for Task A Event-Timex, as discussed below in Section 4.5).

invasão "invasion," because from all the event terms intervening between *lançar* and *agosto passado* (all the other annotated events), the one given by *invasão* is the closest to the time expression, and because the event term *lançar* precedes the time expression *agosto passado* in the text. There is yet another set of features that all take the value NONE, because the time expression does not precede the event term. If there were an event in this sentence mentioned after the time expression, an instance representing a temporal relation between these two entities would have normal values for this last set of features and the value NONE for the previous set of features. This division according to word order is because differences in word order may relate to different syntactic constructions, and by separating these cases we may capture some of that information indirectly.

Each extra classifier feature in each of these four groups of features refers to an annotated attribute of the corresponding EVENT element (the event term's tense, as encoded in the tense attribute; the part-of-speech, taken from the pos attribute, etc.). We avoid using features based on the actual words employed in the text (the surface forms or even the lemmas): because the corpus is relatively small, trying to use classifier features that take as values these strings would likely result in data sparseness issues (the set of their possible values would be very large, and many values would be infrequently represented in the data). In addition there are some boolean features that compare the value of these TimeML attributes with the same attribute of the event in the temporal relation.

Some of the features that appear to be useful for the three temporal relation classification tasks are:

- event-closest-to-timex-pos: this is the value of the pos attribute of the EVENT element that is closest to the temporal expression that is in the temporal relation under consideration. The pos attribute encodes the part-of-speech of the event term. Although many annotated events are verbs, some belong to other parts-of-speech.

- event-closest-to-timex-equal-pos: this is a boolean feature that checks whether the value of the pos attributes of two EVENT elements are identical. This classifier feature compares the pos attribute of the event in the temporal relation and that of the event that is mentioned in the text closest to the time in the temporal relation.

- event-closest-to-event-equal-lemma: this is a boolean feature that checks whether the value of the stem attributes of two EVENT elements are identical. This classifier feature compares the stem attribute of the event in the temporal relation and that of the event that is mentioned in the text closest to it.

- event-closest-to-event-pos: this is the value of the pos attribute of the EVENT element that is closest to the event that is in the temporal relation under consideration.

- event-closest-to-event-equal-pos: this is a boolean feature that checks whether the value of the pos attributes of two EVENT elements are identical. This classifier feature compares the pos attribute of the event in the temporal relation and that of the event that is mentioned in the text closest to it.

- event-closest-to-event-class: this is the value of the class attribute of the EVENT element that is closest to the event that is in the temporal relation under consideration. This class attribute contains several different kinds of information, described in Section 3.3.1.

- event-closest-to-event-equal-class: this is a boolean feature that checks whether the value of the class attributes of two EVENT elements are identical. This classifier feature compares the class attribute of the event in the temporal relation and that of the event that is closest to it in the text.

- event-closest-to-event-equal-tense: this is a boolean feature that checks whether the value of the tense attributes of two EVENT elements are identical. This classifier feature compares the tense attribute of the event in the temporal relation and that of the event that is closest to it in the text.

- event-closest-to-event-simplified-tense: this is the simplified tense of the EVENT element that is closest to the event that is in the temporal relation under consideration. This simplified tense is what is presented above at the beginning of this section.

- event-closest-to-event-equal-simplified-tense: this is a boolean feature that checks whether the simplified tense of two events are identical. This classifier feature compares the simplified tense of the event in the temporal relation and that of the event that is closest to it in the text.

- event-closest-to-event-temporal-direction: this is the value of the temporal direction (Section 4.4.3) for the event closest to the event in the temporal relation to be classified.

- event-intervening-preceding-class: this is the value of the class attribute of the event that is mentioned in between the time and the event in the temporal relation under consideration and is mentioned in the text closest to that time, and this time and event textually precede the event in the temporal relation;

- event-intervening-following-tense: this is the value of the tense attribute of the event that is mentioned in between the time and the event in the temporal relation under consideration and is mentioned in the text closest to that time, and this time and event textually follow the event in the temporal relation;

Comparing the attributes of the two events in Task C Event-Event temporal relations For Task C Event-Event, we also experiment with comparing the different attributes of the two events involved in the temporal relation.

We start with a motivating example. In Section 4.4.1, it was mentioned that co-referring event terms are annotated as denoting overlapping events in the annotations we are using. As such, features that help identify co-referent event terms should be useful for Task C Event-Event. Comparing the subject agreement properties of event terms (when they are verbs) may help detect this, and this information gives rise to a classifier feature described in that section.

Another potentially interesting feature for Task C Event-Event, also with the goal of detecting co-referent event terms is one that compares the lemma of the two event terms for string equality (ignoring font case). The underlying idea is that in some cases multiple mentions of one event are made with words that have the same lemma or even the same surface form.

The following example, taken from the training data, illustrates this point (the English original is shown below):

(35) O dividendo pago sobre as ações ordinárias também se aplica às novas ações, **disse** a companhia. A iniciativa recompensa os acionistas e deve melhorar a liquidez das ações, **disse** a Oneida.
The cash dividend paid on the common stock also will apply to the new shares, the company **said***. The move rewards shareholders and should improve the stock's liquidity, Oneida* **said***.*

In this example, the two highlighted event terms are annotated as representing overlapping events. Arguably, they in fact refer to the same saying event. To capture this, we employ a boolean feature that encodes whether the lemmas of the two terms that denote the events in the temporal relation are identical. This feature is called events-equal-lemma.

Since the lemma is given in the stem attribute of EVENT elements in the TimeML annotations, we may also try comparing the other attributes, and encoding that information in other features for the classifiers. We also try extra features that compare values that are computed from these attributes (and already used as independent features). For instance, the temporal direction of an event is solely based on its annotated stem. Therefore, it is possible to check whether the temporal direction of the two events involved in a Task C temporal relation is identical, too. Indeed, this feature, called events-equal-temporal-direction, seems useful for Task C Event-Event (see Section 4.5).

Features about the previous temporal relation Task B Event-DocTime relates events with the document creation time (DCT). In principle, this temporal relation is somewhat independent of the temporal relation between the previously mentioned event and the DCT in so far as it is possible to intermix clauses and sentences referring to past events with sentences referring to future events and with sentences referring to ongoing events.

However, we conjecture that narratives don't constantly switch between talking about the present, the past, and the future. Instead, events that happen close to each other in time are likely to be frequently mentioned in the same parts of a narrative. We do not really know how strong this effect is, or if there is such an effect, but we can always encode this information in a feature and see if it improves the results.

The feature previous-temporal-relation-type records the type (BEFORE, AFTER, etc.) of the previous temporal relation of the same task.

Once again, we experimented with several additional features, based on the different annotations for the previous instance. Some additional features that proved interesting for the classifiers were:

- previous-instance-event-tense: this feature encodes the tense (i.e. the value of the tense attribute of the TimeML EVENT element) for the event that is the first argument of the previous temporal relation;
- previous-instance-event-simplified-tense: this feature is similar, but encodes the simplified tense for that event instead;
- previous-instance-event-temporal-direction: this feature encodes the temporal direction (Section 4.4.3) of the event that is the first argument of the previous temporal relation.

4.5 Feature Selection and Results

The classifier features presented in Section 4.4 are combined in the following manner. The final set of features is searched using a greedy approach. Even though this process does not guarantee finding the best possible combination, it is not feasible to check all combinations of features in a reasonable time, due to the number of features tested.

For each of the algorithms presented above, a model is trained with the full set of all the classifier features described in the previous sections and then evaluated. Each of these features is then removed, creating new feature sets, each with one less feature. One new model is trained and evaluated for each of the new feature sets. If any of these new models shows a better score, the best one is kept as the best model. We then try to reduce the feature set it uses once again, in the same fashion. This procedure is repeated until no improvement can be found, or all classifier features are removed. For these comparisons, 10-fold cross-validation on the training data is used to obtain the evaluation scores.

This process is conducted for the several different machine learning algorithms mentioned above, and for the different tasks. The results produced by each algorithm

	Task A		Task B		Task C	
Algorithm	Cval	Test	Cval	Test	Cval	Test
DecisionTable	65.8	62.1	79.3	76.7	53.1	50.4
J48	67.7	61.0	81.3	76.7	57.3	**55.0**
JRip	65.7	63.3	79.9	74.9	55.2	53.5
KStar	68.5	62.1	80.4	78.9	56.3	44.6
NaiveBayes	**69.0**	65.0	81.3	78.6	56.9	49.2
SMO	68.3	**66.9**	**82.5**	**79.8**	**58.1**	**55.0**

Table 4.5: Final results for temporal relation classification: results with the optimal combination of features for each algorithm and task, under two evaluation schemes: 10-fold cross-validation on the training data (Cval) and evaluation on the test data of models induced from the full training set (Test).

with the best set of features found in this way is presented in Table 4.5. This table present results for two evaluation schemes. The first is 10-fold cross-validation on the training data. This is the evaluation score that is optimized during the feature selection process just described. The value presented in the table is the one for the best combination of features found. The second score presented in this table is obtained by using the unseen test data to evaluate the model trained on the full training data using the optimal set of features found this way (with cross-validation). Each of these scores is the percentage of correctly classified instances, since this is the evaluation metric used in the two TempEval challenges. The best combination of algorithm and features is highlighted in boldface in this table, for each evaluation regime (cross-validation and train plus test).

As can be seen from Table 4.5, support vector machines (Weka's SMO class) are very powerful and consistently produce the best results or results close to the best one. This is in line with other natural language processing tasks, where this classifier is very popular.

The final sets of features chosen in the best performing solutions are presented in Appendix V. The best classifier for Task A Event-Timex on the test data (SMO) uses the set of features in Figure V.1. Figure V.2 and Figure V.3 list the best set of features found for Task B Event-DocTime and Task C Event-Event, respectively, both also with the SMO algorithm.

	Task		
Classifier	Task A	Task B	Task C
Majority class baseline	49.4	62.4	41.8
Simple features baseline	58.3	80.2	57.0
Best classifier	69.0	82.5	58.1

Table 4.6: Final results compared to baselines: 10-fold cross-validation

	Task		
Classifier	Task A	Task B	Task C
Majority class baseline	59.2	56.2	47.3
Simple features baseline	61.0	77.0	53.9
Best classifier	66.9	79.8	55.0

Table 4.7: Final results compared to baselines: evaluation on unseen test data

4.5.1 Comparison with the Baselines

Table 4.6 compares the classifiers in Table 4.5 with the majority class baseline and the baseline classifiers presented earlier in Table 4.2 in Section 4.2. This pertains to the results obtained with 10-fold cross-validation on the training data. The corresponding evaluation on the test data in in Table 4.7, comparing the baseline performance in Table 4.3 with the final results in Table 4.5.

In these tables, the *majority class baseline* consists in always assigning the class value that occurs most often in the training data, and it had been presented before in Table 4.2. The *simple features baseline* for each task is the best baseline classifier in Table 4.2 for that task (KStar for Task A Event-Timex in the cross-validation scenario and SMO for Task A Event-Timex in the evaluation on the test set, NaiveBayes for Task B Event-DocTime and SMO for Task C Event-Event). These baselines consist in classifiers trained with a small set of simple features, easily computed from the annotations in TimeBankPT. The *best classifier* for each task is the best classifier of Table 4.5 for that task (NaiveBayes for Task A Event-Timex in the cross-validation scenario and SMO for the remaining combinations of task and evaluation scheme).

The difference between each of the two baselines and the classifier with the new features is significant for Task A Event-Timex and Task B Event-DocTime, according to Weka's PairedCorrectedTTester and with a 0.05 level of statistical significance (for Task A Event-Timex, these differences are significant also at the 0.01 level). PairedCorrectedTTester implements *t*-test statistics for the classification results of two or more classifiers. For Task C Event-Event the differences between the majority class baseline and the other two are statistically significant. The difference between the simple features baseline and the classifier employing the new features is not, however. When evaluated on the test set, the same picture emerges.

The drastic and significant improvement that the new features produce on Task A Event-Timex reflects the fact that most of these new features are aimed at precisely this task. Many of them are based on syntactic information (e.g. the features presented in Section 4.4.5) or try to approximate syntactic information (quite a few of the features in Section 4.4.1, Section 4.4.3 and Section 4.4.6), which is relevant for Task A Event-Timex, as this task is about temporal relations between two entities mentioned in the same sentence. The temporal relations for the other tasks are between entities in different sentences. Task B Event-DocTime is between events and the DCT, which in TimeBankPT and in the data for TempEval is mentioned at the beginning of documents, before the sentences with the annotated events and timexes occur. Task C Event-Event is about events mentioned in different sentences. Logic inference (Section 4.4.4) is not very useful for Task C Event-Event. Since this task is about temporal relations between two events and the other two tasks are about relations involving one event and one time, only very few relations for Task C Event-Event can be discovered (e.g. when one event precedes the document creation time and the other one follows it).

4.5.2 Ablation Tests

In order to assess the usefulness of the several attributes employed by these classifiers, they are compared with similar models trained with fewer features. This assessment uses 10-fold cross-validation on the training data. The support vector machines are used for the three tasks, as these classifiers perform best. It is performed in two different conditions.

In the first condition, each of the features is removed at a time from the optimal feature set, and the classifiers obtained are evaluated. This results in a ranking of the various classifier features according to their usefulness. The most useful feature is the one with the greatest impact on classifier performance, i.e. the feature which the lowest scoring classifier is lacking. For the sake of illustration, the left part of Table 4.8 shows this ranking for Task A Event-Timex. There, it can be seen that the feature with the highest impact on classifier performance is the feature predictor-parser. Removing this feature from the optimal set of features results in a classifier with an accuracy that is 2.6 percentage points lower than that of the classifier trained with the optimal feature set.

In the second condition, each feature is removed successively from the optimal feature set, starting from the best feature and finishing with the worst one. At each point, the remaining features are reevaluated. That is, in a first step, the features are ranked in the same way as in the first condition. This feature is then removed from the feature set. This ranking operation is performed again on this reduced feature set and the new best feature is removed. This procedure is conducted recursively until no feature is left. For right-hand side of Table 4.8 shows the result for Task A Event-Timex. As can be seen from there, performance degrades very rapidly when multiple features are successively removed. This condition also produces an ordering of features according to their impact on the classification scores, but one that takes feature interactions into account: the second best feature becomes the best feature left once the very best feature is removed from the original optimal feature set.

Comparing these two orderings can shed some light on interactions between features. Sometimes, the usefulness of one feature depends on the presence of other features.

Task A Event-Timex For Task A Event-Timex, the features revealed as the best ones are presented in Table 4.8, under these two conditions.

The number after each feature describes the impact on the performance of the classifier trained with that feature removed from the feature set. More specifically, it is the difference between the model trained with that feature removed and the classifier using the full feature set. The features that are shown in this table are the five ones whose removal had the most dramatic impact on the classifier scores. The difference in scores is statistically significant only for the feature predictor-parser, for a significance level of 0.05, according to Weka's PairedCorrectedTTester. When

Individual removal of features		Successive removal of features	
Feature	Individual impact	Feature	Cumulative impact
predictor-parser	-2.6	predictor-parser	-2.6
event-intervening-following-tense	-1.9	event-intervening-following-tense	-4.8
closure-B-for-A	-1.5	predictor-dep-parser	-7.2
predictor-dep-parser	-1.1	closure-B-for-A	-9.9
event-temporal-direction	-1.1	timex3-relevant-lemmas	-13.6

Table 4.8: Ablation analysis of the SMO classifier for Task A Event-Timex

removed individually, the other features do not produce statistically significant differences. The best features, according to the second condition, are also in Table 4.8.

From this table, it can be seen that the most informative feature is the one based on the phrase structure parser (Section 4.4.5). After removing this feature, the feature event-intervening-following-tense (Section 4.4.6) is the strongest. This feature records the tense of another event in the sentence, namely the one closest to the temporal expression when both are mentioned after the event in the temporal relation. This feature can be useful for examples such as the one in (29), repeated here in (36).

(36) a. Soviet Foreign Ministry spokesman Yuri Gremitskikh said special ambassador Mikhail Sytenko left Tuesday for consultations with the governments of Syria, Jordan, Egypt and other Arab countries.

b. O porta-voz do Ministério dos Negócios Estrangeiros soviético Yuri Gremitskikh disse que o embaixador especial Mikhail Sytenko partiu terça-feira para consultar os governos da Síria, Jordânia, Egito e outros países árabes.

In this example, there is a temporal relation between *said* and *Tuesday*. The event is after the date. The fact that there is another event, the one denoted by *left*, closer to the time expression, is an indication that the time expression modifies this other event, and thus describes the time when this leaving event happened, rather than the saying event. The fact that *left* follows *said* in the sentence is a cue to the syntactic relation between the two verbs: *left* is the main verb of the complement clause of *said*. This and the tense of the two verbs is a strong indication that the event for *said* is after the event for *left*, even more so in Portuguese, where the two

Individual removal of features		Successive removal of features	
Feature	Individual impact	Feature	Cumulative impact
event-simplified-tense	-2.9	event-simplified-tense	-2.9
previous-temporal-relation-type	-1.6	predictor-parser	-5.1
event-class	-1.4	predictor-dep-parser	-9.1
event-closest-to-event-class	-1.1	event-class	-12.8
event-closest-to-event-simplified-tense	-0.9	event-indicator-st1	-14.8

Table 4.9: Ablation analysis of the SMO classifier for Task B Event-DocTime

perfective past forms only allow this possibility (see Section 5.3.3.4, specifically on this phenomenon).

After that, the other feature based on parsing, predictor-dep-parser (Section 4.4.5), is the best. Temporal deduction also ranks high in both conditions.

Task B Event-DocTime The ablation tests for Task B Event-DocTime are shown in Table 4.9. For Task B, each of the three best features (event-simplified-tense, previous-temporal-relation-type and event-class) produces statistically significant differences when it is individually removed from the complete feature set.

A number of comments can be made about Table 4.9. Tense remains the most informative feature. It seems that the information provided by the temporally decorated grammatical representations (the features predictor-parser and predictor-dep-parser) is somewhat redundant with that provided by tense (the feature event-simplified-tense), as they have a much bigger impact when tense is not available (the two right columns of the table) then when it is (the left columns of the table). This is because, in these automatically produced representations, the temporal relations between a verb and the DCT are mostly based on the grammatical tense of the verb.

The temporal relation between the previously mentioned event and the DTC (the feature previous-temporal-relation-type) is another useful feature, but it is dependent on the tense of the event in the current temporal relation (the feature event-simplified-tense). When this tense is known (first condition) this feature has high impact, but once tense is removed this feature becomes much less useful (second condition). Support vector machines are difficult to inspect by humans, so if

Individual removal of features		Successive removal of features	
Feature	Individual impact	Feature	Cumulative impact
event-simplified-tense 1	-4.6	event-simplified-tense 1	-4.6
event-class 2	-1.7	event-class 1	-5.2
previous-instance-event-tense 1	-1.2	event-temporal-direction 1	-7.8
event-class 1	-1.2	event-class 2	-9.1
predictor-parser	-1.1	event-temporal-direction 2	-9.7

Table 4.10: Ablation analysis of the SMO classifier for Task C Event-Event

one wants to look at the inside of the models induced from these data, other algorithms must be used (although this must be viewed with a grain of salt as different algorithms have different properties about the way in which they can separate the instances into classes). Inspecting a decision tree trained on the training data with the J48 algorithm and just the three best features (according to the left side of Table 4.9) shows that the information about the previous temporal relation is used to disambiguate Task B temporal relations involving present tense verbs, which can enter OVERLAP, BEFORE and AFTER relations with the DCT. The event-class feature (which is part of the initial set of features used in the baselines) is also one of the three features producing statistically significant differences, as mentioned. Inspection of a similar decision tree trained with a reduced set of features as well as the distribution of the values for this feature in the training data reveals that the useful bit of information is when this feature takes the REPORTING value, for verbs like *say*, *announce*, etc. The reason is that the temporal relation with the DCT is practically never AFTER: reporting events are almost never future events. This may be particular to the corpus used (it is made of news articles), or it may also be seen in other types of texts. It is an interesting piece of information about the world that is not captured by the new features developed in our work.

Task C Event-Event The ablation tests for Task C Event-Event are shown in Table 4.10. Since Task C is about temporal relations between two events, the features that describe properties of these events come in pairs. In this table, the number 1 after a feature's name indicates a feature describing the event that is the first argument of the temporal relations, the number 2 is used with features describing the second argument.

Instance number	Feature Vector		Class
1	<TRUE,	TRUE>	A
2	<TRUE,	TRUE>	B
3	<TRUE,	TRUE>	B
4	<TRUE,	FALSE>	B
5	<FALSE,	FALSE>	A
6	<FALSE,	FALSE>	A
7	<FALSE,	FALSE>	A
8	<FALSE,	FALSE>	B
9	<FALSE,	FALSE>	B

Table 4.11: A hypothetical set of instances

For Task C Event-Event, only the best feature produces statistically significant differences when removed from the complete feature set. This is the feature describing the simplified tense of the event that is the first argument of the temporal relation. The second best feature is not about the tense of the second event, but rather its annotated class, which contains some information of aspectual type, namely a binary distinction between states and the remaining types (Section 3.3.1 and Section 4.4.2). As mentioned before in Section 4.2, a state as the second event is expected to go with overlap relations more than a non-stative situation, because states tend to be used to describe the ways things were when the previously mentioned events occurred whereas non-stative situations move a narrative forward. A decision tree trained with just these two features for this task indeed shows this sort of association.

4.5.3 Error Analysis

Consider a simplified problem where we use two Boolean features, the class values are A and B, and the training set is composed of the instances in Table 4.11.

A reasonable classifier trained on this data will assign a new instance with the feature vector <TRUE, FALSE> to class B, because of the training instance no. 4. But in the case of an instance like <TRUE, TRUE> it received conflicting evidence of its class: the instances 1, 2, and 3 have similar feature vectors, but while instance

1 has class A, the other two have the class B. The feature vector <FALSE, FALSE> is affected by the same problem: the instances 5, 6 and 7 exhibit these features and have the class A, the instances 8 and 9 have the class B. If in these cases our reasonable classifier assigns the majority class in each group of instances that share the same feature values, he will assign class B to instance no. 1 and class A to instances 8 and 9. If the classifier does not choose the majority class within each group of instances with the same features, error will be higher, at least on this training data. Therefore, this classifier cannot be 100% accurate when classifying instances that it has seen during training, and such a classifier will likely not be 100% accurate on unseen instances represented with the same set of features. This problem can be solved by adding features that differentiate these instances.

The same problem can be seen with the classifiers and the sets of features used in our work.

Even with all the features employed, for each task there are still some groups of instances such that all instances in that group are identical (their feature vectors are identical) but not all instances in that group have the same class. This means that in any of these tasks, 100% accuracy is impossible with these features.

Using the optimal set of features for the best classifiers for each task (the support vector machines), this amounts to the following numbers. For Task A Event-Timex, there are 7 such groups in the training data, affecting 17 instances. The number of instances that do not exhibit the majority class in their group is 7. Therefore, in the training data at least, error has to be at least 0.5%.

For the other tasks these numbers are higher. For Task B Event-DocTime, there are 16 groups and 35 instances. The number of instances associated with the minority class is 16, or 0.6% of all instances in the training set. In the case of Task C Event-Event, there are 20 such groups, encompassing 64 training instances. 25 instances do not have the same class as the majority class in their group, or 1.4% of the total number of training instances. The test data do not show this problem.

With the simple features baselines (the baseline classifiers that employ a smaller set of features), this problem is much stronger. For Task A Event-Timex, there are 341 instances in the training data affected by it, in 95 groups. Around 10% of the total number of instances belong to one of these groups and do not have the majority class of their group. 10% is also roughly the gain in accuracy on the training data for this task when we go from the simple features baselines to the final classifiers. For Task B Event-DocTime, 459 instances are affected, in 120 groups. Around 7% of the instances are affected and do not have the majority class of their group. For Task C Event-Event, this amounts to just 3%. Task A Event-Timex is the one where our work showed the greatest improvement but also the one where clearly more features were needed to properly distinguish the instances. The very small size of this problem in the final models indicates that further progress may be difficult with the mere introduction of more features.

These classifiers always produce an answer (no instance is left unclassified), but recall and precision measures can still be computed for each class value, and we can take their average, weighted by their frequency, as global recall, precision and F-measure.[1]

Table 4.12, Table 4.13 and Table 4.14 show the precision, recall and F-measure scores for Task A Event-Timex, Task B Event-DocTime and Task C Event-Event respectively, broken down by class. They show that some classes are much harder than others. The vague classes BEFORE-OR-OVERLAP, OVERLAP-OR-AFTER and VAGUE show null scores (on the test data they show 0 precision, recall and F-measure for the three scores), probably because of their low frequency in the data, at least partly. The instances of these classes are also naturally harder to classify, since they

[1] Precision is defined as the number of true positives tp divided by the sum of the number of true positives with the number of false positives fp:

$$P = \frac{tp}{tp + fp} \tag{4.1}$$

Recall is the number of true positives divided by the sum of the number of true positives with number of false negatives fn:

$$R = \frac{tp}{tp + fn} \tag{4.2}$$

The F-measure is the harmonic mean of precision and recall:

$$F = 2 \times \frac{P \times R}{P + R} \tag{4.3}$$

For instance, for the class OVERLAP the true positives are the instances correctly classified as OVERLAP, the false positives are the instances incorrectly classified as OVERLAP, and the false negatives are the instances that should have been classified as OVERLAP but were not.

	10-fold cross-validation			Evaluation on test data		
Class	P	R	F	P	R	F
10-fold Cross-validation						
OVERLAP	0.716	0.804	0.758	0.754	0.86	0.804
BEFORE	0.682	0.63	0.655	0.615	0.421	0.5
AFTER	0.621	0.649	0.634	0.452	0.633	0.528
BEFORE-OR-OVERLAP	0	0	0	0	0	0
OVERLAP-OR-AFTER	0	0	0	0	0	0
VAGUE	0.5	0.04	0.074	0	0	0
Weighted avg.	0.65	0.683	0.661	0.596	0.669	0.625

Table 4.12: Precision (P), recall (R) and F-measure (F) of the support vector machine for Task A Event-Timex, broken down by class

are exactly those for which the human annotators could not make a specific decision. The majority classes (OVERLAP for Task A Event-Timex and Task C Event-Event and BEFORE for Task B Event-DocTime) seem to be the easiest, showing F-measures (0.804 for Task A Event-Timex, 0.874 for Task B Event-DocTime, and 0.653 for Task C Event-Event, on unseen data) much higher than the weighted average F-measure for the task and evaluation method.

The majority classes always show higher recall than precision, reflecting a general bias for the majority class even with all the new features. For Task A Event-Timex, this is also the case for the second most frequent class (AFTER) on the unseen test data. The other classes show much poorer recall, which means that this classifier is strongly biased for the two most frequent classes.

In the case of Task B Event-DocTime, recall is higher than precision for the majority class and the third most frequent class (BEFORE and AFTER, respectively). The OVERLAP class, which is the second most frequent class, shows the inverse numbers. The most useful feature for this classifier is verb tense, so this difficulty may be linked to the tense system of Portuguese, possibly with the ambiguity of the present tense. This tense can describe ongoing events, but also past (the historical use of the present tense) and future events. In Portuguese, it is used to describe future events much more often than in English.

In Task C Event-Event, the majority class is once again OVERLAP and the second most frequent class is BEFORE. Here, once again recall lines up with frequency.

	10-fold cross-validation			Evaluation on test data		
Class	P	R	F	P	R	F
10-fold Cross-validation						
OVERLAP	0.718	0.685	0.701	0.847	0.61	0.709
BEFORE	0.881	0.938	0.909	0.808	0.952	0.874
AFTER	0.728	0.762	0.745	0.686	0.729	0.707
BEFORE-OR-OVERLAP	0	0	0	0	0	0
OVERLAP-OR-AFTER	0.333	0.057	0.098	0	0	0
VAGUE	0.364	0.111	0.17	0	0	0
Weighted avg.	0.798	0.825	0.809	0.764	0.792	0.769

Table 4.13: Precision (P), recall (R) and F-measure (F) of the support vector machine for Task B Event-DocTime, broken down by class

	10-fold cross-validation			Evaluation on test data		
Class	P	R	F	P	R	F
10-fold Cross-validation						
OVERLAP	0.625	0.717	0.668	0.65	0.656	0.653
BEFORE	0.53	0.691	0.6	0.425	0.627	0.507
AFTER	0.573	0.62	0.596	0.521	0.595	0.556
BEFORE-OR-OVERLAP	0	0	0	0	0	0
OVERLAP-OR-AFTER	0	0	0	0	0	0
VAGUE	0	0	0	0	0	0
Weighted avg.	0.494	0.581	0.533	0.49	0.55	0.515

Table 4.14: Precision (P), recall (R) and F-measure (F) of the support vector machine for Task C Event-Event, broken down by class

There are many test instances misclassified as BEFORE, which is reflected in its relatively low precision

4.5.4 Discussion

Temporal relation classification is still a hard task, incurring high error rates. Our feature engineering work shows improvements, mostly for Task A Event-Timex. The Table 4.6 and the Table 4.7 above in Section 4.5.1 show that this approach can produce improvements of more than 5 percentage points on Task A Event-Timex. With cross-validation on the training data, the results go up more than 10 percentage points. This is a very dramatic improvement. The improvements are statistically significant for Task A Event-Timex and Task B Event-DocTime. Task C Event-Event proved harder, and no statistically different solution was found.

Ablation tests show that the information coming from syntax is very relevant for Task A Event-Timex. For Task B Event-DocTime, the most important information is related to tense, the previous temporal relation of the same kind, and some lexical information about the event term. For Task C Event-Event, tense is once again the most useful piece of information.

Error analysis indicates that there is a strong bias for the majority class in all these tasks. The very seldom seen classes have a high impact on the error found.

These results compare very favorably with the state of the art for English. Task A Event-Timex shows particularly better results than the ones in either TempEval, with the added comment that this task was easier in the second TempEval as temporal relations between events and times not directly related in the phrase structure were not considered. The language difference, however, means that this comparison must be seen with a grain of salt.

It must be noted that the best solutions that were found here require language specific features. The biggest improvements often come from incorporating syntactic information in these classifiers, which is highly language specific. Other features, such as the ones about properties of events mentioned in the proximity of the entities in the temporal relation under classification, are language specific in a different way: they have a substantial impact on classifier performance in our work, with Portuguese. But since each of them is trying to capture specific syntactic patterns, the usefulness of each of these features is dependent on syntactic properties of the language. Also grammatical tense, another one of the most important features found, is language dependent. This result shows that research on the processing of different languages is important; it is not to be expected that the results obtained with one language carry over to every other unchanged. Our choice of using Portuguese data and tools for the processing of Portuguese is thus justified, even if it required the additional effort of producing data in this language.

Temporal relation classification is a hard task. It often requires a combination of reasoning, grammatical knowledge and knowledge of the world. Many examples show this difficulty, like the one presented above in (33), in Section 4.4.5, and repeated here in (37):

(37) In Washington today, the Federal Aviation Administration released air traffic control tapes from the night the TWA Flight eight hundred went down.

As mentioned in that Section, examples such as these require common sense reasoning. Broad scale reasoning of this sort is beyond the current technology.

4.6 Summary

This chapter focuses on the classification of temporal relations. Given a temporal relation between two identified entities (an event and a time, or two events) mentioned in a text, the goal is to automatically determine the type of that relation (BEFORE, OVERLAP or AFTER).

The data set used to experiment and evaluate the proposed solutions is TimeBankPT, presented in the previous chapter. This data set contains temporal relations grouped in three different tasks. Task A Event-Timex is to classify temporal relations between events and times mentioned in the same sentence; Task B Event-DocTime is about temporal relations between events and the time in which the document was created; Task C Event-Event is about events occurring in different sentences.

Section 4.2 presents baselines for these tasks that consist of machine learning classifiers trained with features that are readily available in the annotations. Section 4.3 describes several natural language processing tools that are used to create more features in order to enrich these classifiers. These new features are explored in Section 4.4. Many different kinds of classifier features are tried, capturing various sorts of information that are considered relevant to the problem of temporal relation classification. They encode grammatical features, reasoning, knowledge of the world, and combinations of these. Finally, in Section 4.5 the evaluation of the new classifiers, with the extended set of features, is presented, and they are compared to the previously described baselines.

The results show substantial improvements for Task A Event-Timex, for which almost all of the new features were intended. Some improvement can also be seen for the other tasks with these features. They are very competitive with the state of the art for other languages, and they are the first results of temporal relation classification for Portuguese.

Chapter 5

Full Temporal Processing

Temporal relation classification, explored in Chapter 4, is arguably the most interesting task of temporal processing, as it displays the highest error rates among all the tasks of temporal processing, according to the results of TempEval-2, where all tasks were explored. Chapter 4 was therefore entirely devoted to it.

Temporal relation classification assumes that all other temporal annotations are available, namely those for temporal expressions and events, as many of the features for the classifiers of temporal relations depend on these annotations.

For these two reasons (the easier nature of automatically annotating temporal expressions and events, and the dependency of temporal relation classification on annotations for them), it is both interesting and important to integrate the classifiers developed in Chapter 4 (for the problem of temporal relation classification) in a full temporal processing system, which annotates raw text with complete temporal annotations about temporal expressions, events and temporal relations. It is interesting because the hardest problem has already been solved in Chapter 4. It is important because without a means to automatically produce the remaining temporal annotations, it is not possible to use the work developed in the previous chapter to process raw text. The first part of this chapter presents one such full temporal processing system. This system tags raw text with annotations similar to the ones used in TimeBankPT and the English data used in the first TempEval (i.e., a slightly simplified version of TimeML).

The second goal of this chapter is to demonstrate the usefulness of temporal processing by describing the integration of this temporal processing system in a deep linguistic processing system. Several applications of temporal processing have been listed in Section 1.3. Here, we present a different one. Deep computational grammars seek to produce representations of the truth-conditional meaning of input sentences. Truth conditions may not convey the full meaning of sentences, but they are the best approximation that we currently have. However, these deep grammars have problems in processing time, for a number of reasons that will be made clear. A temporal processing system can be used as a dedicated temporal module of these grammars. In this way, the meaning representations produced by the deep grammar are extended with additional information about time. Or, seen in a different way, this information about time extracted by the temporal processing system is combined with rich information about the truth conditions of the sentences in a text. This integration effort is explored in the second part of this chapter, where evaluation results show that it is fruitful.

5.1 Outline

This chapter has two parts. The first part, in Section 5.2, describes and evaluates LX-TimeAnalyzer, a system that automatically annotates raw text with full temporal annotations. This system uses the classifiers developed in the previous chapter for the problem of temporal relation classification, and a number of other classifiers and solutions for the remaining problems involved in temporal processing. The second part, in Section 5.3, describes the integration of LX-TimeAnalyzer with a deep processing grammar. This deep grammar produces meaning representations of input sentences, and these are enriched with the temporal information extracted by LX-TimeAnalyzer. Finally, a summary of the main contributions in this chapter is presented in Section 5.4.

5.2 Automatic Temporal Annotation

In order to automatically extract temporal relations from an unannotated piece of text, it is necessary to first extract other kinds of information from that text. The classifiers presented in the previous chapter rely on TimeML annotations of events and time expressions in order to determine the type of temporal relations.

It is thus necessary to annotate temporal expressions and event terms first. This section describes a simple approach to automatically produce such annotations. The aim here is not to improve the state of the art when it comes to identifying and normalizing temporal expressions and events. Since the state of the art in most of these tasks uses machine learning techniques with data that is annotated in the same way as TimeBankPT is, our goal is rather to see TimeBankPT as an opportunity to develop a full temporal processing system that is not much different from this state of the art. That is, this part of our work is not the focus of our research. It can be seen as an application of the work on temporal relation classification presented in Chapter 4.

These tasks of identifying and normalizing temporal expressions and event terms are, however, not entirely trivial. Temporal expression annotation is more than normalizing date expressions, as temporal expressions denote more than calendar dates and clock times. They can refer to points of finer and coarser granularity, durations and sets of times. Furthermore, they may not be full date and time expressions. They can be incomplete, ambiguous and anaphoric. Event recognition is a difficult task, because many events in the annotated data are denoted by nouns, and it is often difficult for classifiers to distinguish between nouns that denote events (e.g. *accident*) from nouns that do not (e.g. *person*).

5.2.1 Requirements

It is helpful to break down the problem of full temporal annotation into smaller tasks, in order to make it easier to address. The several tasks be be automatized are:

- **Delimiting temporal expressions**
  The first step towards annotating temporal expressions is to find their boundaries in a text.

- **Identifying event terms**
  The first step towards annotating events is to find the event denoting words in a text.

- **Normalizing and annotating temporal expressions**
  The exact date or time that the expression refers to must be determined and represented in the ISO 8601 standard. Initial sub-tasks may consist in determining whether the expression is anchored in another temporal expression or not (for temporal expressions such as *two days later*), and if so identifying the relevant anchor.

- **Annotating events**
  All the attributes that are appropriate for EVENT elements must have their value annotated.

- **Identifying the arguments of temporal relations**
  The pairs of entities that are to be related temporally need to be identified.

- **Classifying temporal relations**
  The temporal classifiers presented in Chapter 4 must then be run in order to complete the annotations of temporal relations.

5.2.2 Related Work

TERN 2004 (Section 2.4) and the second TempEval (Section 2.7) featured tasks about temporal expressions and events. TERN 2004 was exclusively concerned with the recognition and normalization of temporal expressions. TempEval-2 (Verhagen *et al.*, 2010) included more tasks than the first TempEval. The added tasks were concerned precisely with the identification and normalization of temporal expressions and events. Task A of TempEval-2 was concerned with temporal expressions: determining their extent and the value of the features type and value of TIMEX3 elements. Task B of TempEval-2 focused on events: their extent in a text and the value of the attributes class, tense, aspect, polarity and modality of EVENT elements (this last attribute, modality, was not used in the first TempEval and it is absent from the annotations, as well as from those in TimeBankPT). Note that these are not the same tasks as Task A Event-Timex and Task B Event-DocTime of the first TempEval.

Table 5.1 lists the results of TempEval-2. As the table shows, these tasks show substantially lower error rates than temporal relation classification. Several of them

	Scores[1]	
Task	English	Spanish
Task A: temporal expressions		
Extents	0.86	0.91
type	0.98	0.99
value	0.85	0.83
Task B: events		
Extents	0.83	0.88
class	0.79	0.66
tense	0.92	0.96
aspect	0.98	0.89
polarity	0.99	0.92

Table 5.1: Best system results for the various identification and normalization tasks of TempEval-2. The evaluation measures used were the F-measure for the extents tasks and the percent accuracy for the attributes tasks.

can be solved with near perfect performance. The table does not show the results for determining the value of the pos attribute of EVENT elements, which encodes the part-of-speech of the annotated event word (whether it is a noun, verb, etc.). This problem is addressed by dedicated natural language processing tools, namely part-of-speech taggers, like LX-Suite (Section 4.3.1), which currently also operate almost perfectly.

The best performing system at TERN 2004 was Negri & Marseglia (2004). The authors used a symbolic approach. The system operates in two phases. The first phase is concerned with detection and delimitation of temporal expressions and works in the following way:

1. Tag the input text with a part-of-speech tagger.
2. Feed the tagged input to a set of around 1,000 basic handcrafted rules that:
   (a) Detect the possible time expressions in the input text.
   The rules look for specific nouns (*hour*, *day*), names (*Friday*, *August*), adverbs (*tomorrow*) and numbers in the input text. The list of names of calendar dates (e.g. *April Fools' Day*, *Hanukkah*) was mined from

WordNet (Fellbaum, 1998).[1]

(b) Determine their extent.

The system then checks whether the surrounding words should be included in the same time expression. Relevant words can be nouns (*beginning*), adverbs (*before*, *shortly*), adjectives (*next*, *following*), numbers, etc.

(c) Gather all contextual information relevant for the following normalization phases.

This consists in identifying words in the temporal expressions that convey information that are to be represented in the different attributes of TIMEX2 elements (val, mod, etc.). [2]

3. Apply specialized rules to resolve ambiguities between multiple possible taggings.

The second phase, for normalization, is made up of the following steps:

- Anchor selection

  In this phase, relative temporal expressions such as *three years later* are connected to absolute time expressions like *2001*. The latter is called the anchor. This is necessary for normalization (*2004* in this example). The system considers potential anchors based on granularity (the two expressions refer to a calendar year—they have the same granularity). The possible anchors of a time expression typically precede it in the text. Depending on the words found in the relative time expressions, the system assigns as its anchor either the document's creation time (for a relative time expression such as *last year*)

[1] WordNet is a lexical database where words and phrases are organized in concepts, the so-called WordNet synsets (or sets of synonyms). WordNet records several relations between synsets, such as hyponymy (more specific concepts) and hypernymy (more general ones). For instance, the list of direct hyponyms of the synset "religious holiday, holy day" includes "fast day", "Christian holy day", "Jewish holy day" and "Dormition, Feast of Dormition" (a celebration in the Eastern Orthodox Church).

[2] In the annotation scheme used in TERN 2004, temporal expressions are marked with TIMEX2 tags instead of the TIMEX3 tags used in TimeML. Here, the val attribute corresponds to the value attribute of TIMEX3 elements.

or the previous temporal expression of compatible granularity (for a relative time expression like *the previous year*).

- Dates normalization

  This step is dedicated to filling the value of the val attribute of TIMEX2 elements.

- Attributes normalization

  In this final step the internal representation is converted in the annotations that the system must output.

This system achieved a 0.93 F-measure for the identification problem. For normalization, the system scored between 0.69 and 0.87 for the various attributes of TIMEX2 elements, with the highest score (0.87 F-measure) being seen with respect to the val attribute. The authors report that one particularly evident source of error is direct speech. For instance, in a sentence like (38), the system will predict the anchor of the second temporal expression in bold to be the document's creation time when in fact it must be connected with *1998*.

(38) *He concluded the* **1998** *annual meeting saying: "***The next year** *will be the eve of a new era for our company".*

Elliptical noun phrases also pose difficulties. In the example in (39), *this* stands for *this Winter*, but, with the noun missing, this temporal expression is difficult to detect.

(39) *Evelyn Griffin has seen* **80 winters**. **This**, *she says, was the coldest.*

Another difficulty reported by the authors is ambiguity with proper names. For instance, *April* can be a person's name, and *20th Century Fox* is a company's name but *20th Century* erroneously gets tagged as a temporal expression.

The systems at TERN 2004 were evaluated against a test corpus of around 50,000 words.

Since then, machine learning approaches have been able to match these results. Ahn *et al.* (2007) replaces the large set of handcrafted rules typical of systems for this task by a series of machine learned classifiers and a much smaller set of rules.

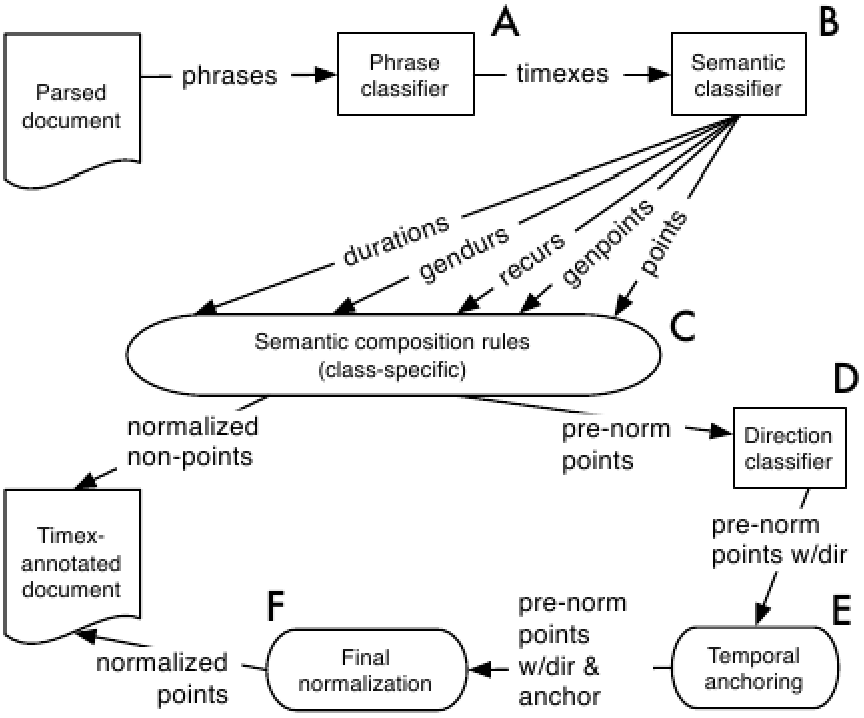


Figure 5.1: Ahn *et al.* (2007)'s system for timex recognition

As just mentioned, the system of Negri & Marseglia (2004) at TERN 2004 contained around 1,000 hand coded rules just for detecting temporal expressions. Figure 5.1 shows the overall architecture of the system of Ahn *et al.* (2007).

The system begins with parsed documents as input. They are parsed with the constituency parser of Charniak (2000). A classifier decides which phrases are temporal expressions (component A in that figure). This contrasts with other approaches. Recognition of temporal expressions or other phrases based on machine learning often consists in learning a classifier that assigns to each word in a text one of three categories: B (the word is at the beginning of a phrase of the target type), I (the word is inside but not at the beginning of such a phrase), O (the word is outside a phrase of the target type). IOB tags have been used by e.g. Llorens *et al.* (2010a) in TempEval-2 in the context of temporal expression recognition. It must be noted that all annotation schemes (TimeML, TERN, etc.) disallow timexes embedded in other timexes. In those cases when one arguably finds embedding, only the outermost expression is tagged: <TIMEX>*the day after tomorrow*</TIMEX>, not <TIMEX>*the day after* <TIMEX>*tomorrow*</TIMEX></TIMEX>. Because of this restriction, IOB tags are sufficient to represent the full extent of temporal expressions. The approach followed by Ahn *et al.* (2007) was to instead use a machine learned model on the parsed text, classifying each phrase as being a time expression or not. The classifier features used include character type patterns, lexical features, a context window of two words to the left, the syntactic category of the phrase, its head, the initial word, that word's part-of-speech, and the dependency parent of the head, and the corresponding dependency relation.

Another classifier assigns a semantic class to the recognized timexes (this is component B in Figure 5.1). These classes distinguish *inter alia* durations, dates, and times. They pertain to the val attribute of TIMEX2 elements. The classifier features are the same as the ones used in component A. Based on semantic class, component C then maps lexical items found in the expression to a semantic representation. In the case of durations (e.g. *three hours*), it tries to identify the unit (*hour*) and the amount (*three*). For dates and times, it tries to fill in the various relevant values: year, month, etc. Component C is composed of 89 handcrafted rules. The remaining components deal with timexes that need an anchor in order to be normalized. One is responsible for finding this anchor (E). This is done in a way very similar to how Negri & Marseglia (2004) find timex anchors, described above. Another machine learned classifier determines whether the timex refers to a point before or after that of its anchor (D). For instance, a time expression like *two days later* denotes a date (two days) after its anchor date. Although this information is not annotated, training and test examples can be constructed by comparing the val attribute of a timex to that of its anchor. This classifier used a superset of the features that were used for recognition and semantic classification, which contains features describing the tense of nearby verbs and features comparing the timex with the document creation time.

The system was evaluated with the data of TERN 2004. It achieved a 0.87 F-measure for the identification of temporal expressions (component A), not very far from the best system of TERN 2004, that scored 0.93. Each component in Figure 5.1 produces some error, placing an upper bound on the final quality of the normalization process. Overall, the F-measure for correctly filling in the val attribute of a temporal expression is 0.899, compared to the 0.872 score for the best system at TERN 2004 (which was fully rule-based).

For Portuguese, the second HAREM challenge of named entity recognition (already mentioned in Chapter 2) included a track for temporal expressions. The data used for HAREM included 14,056 words and 193 normalized temporal expressions. It covered both recognition and normalization, and the best system was XIP (Hagège *et al.*, 2008b), a rule-based system: for recognition, the system obtained an F-measure of 0.76; for normalization the F-measure was 0.74.

5.2.3 Approach

Because a considerable amount of annotations are provided in the training and test data of TimeBankPT, it supports the training and evaluation of temporal processing systems for Portuguese. LX-TimeAnalyzer is a first attempt at full temporal processing of Portuguese, following this approach.

In Table 5.2, information is included about the system's performance, with evaluation scores for each sub-task that was evaluated in TempEval-2 (with the exception of temporal relation classification, which is reported in Chapter 4). The evaluation scores are the same as the ones used in TempEval-2 (Verhagen *et al.*, 2010), and they are relative to the performance on the unseen test data, using the entire training set for training. For the recognition tasks, the F-measure is used. For the remaining tasks, classification accuracy is presented.

The results are not entirely comparable to those of TempEval, since the data and the languages are different. The Portuguese data, TimeBankPT, are an adaptation of the English data used in the first TempEval, while the results in Table 5.1 refer to TempEval-2. The English data of TempEval and TempEval-2 are not completely identical, although there is a large overlap between them. For the data of the first TempEval there are unfortunately no published results that we know of concerning the identification and normalization of temporal expressions and event terms, as TempEval focused only on temporal relations. Our results are thus not fully comparable to the results for English, and they are even less comparable to the results for Spanish, as they are based on completely different data.

Task	Score
Temporal expressions	
Extents	0.86
type	0.91
value	0.81
Events	
Extents	0.72
class	0.74
tense	0.95
aspect	0.96
polarity	0.99

Table 5.2: Evaluation of LX-TimeAnalyzer on the identification and normalization tasks, using the test data. The evaluation measures used were the F-measure for the problems of identifying the extents of event and time expressions and accuracy for the tasks dealing with the attributes.

Figure 5.2 shows the system's architecture. LX-TimeAnalyzer expects the document's creation time (DCT) to be encoded in the name of the file for the document to process, i.e. it takes as input not just the document but also the DCT. This is because the DCT is often easy to get from meta-data or by the client. The text to process is first tagged with LX-Suite (morphological analysis). The resulting text, annotated with this morphological information, is then processed. The several subtasks include: event identification (finding event terms in text), event normalization (assigning values to the several attributes of EVENT elements), identifying temporal expressions (finding their boundaries in text), normalizing temporal expressions (filling in the value of the various attributes of TIMEX3 elements), finding TLINKs, i.e. temporal relations (identifying their arguments), and finally classifying them (the task to which the previous chapter is devoted). In the following sections, these components of the system are described.

LX-TimeAnalyzer is also described in Costa & Branco (2012c) and Costa & Branco (2012b).

Morphological Analysis The document to be processed is first tagged with LX-Suite. As mentioned in Section 4.3.1, this tool annotates each word with a wealth of information, namely: its part-of-speech, its dictionary form, a code describing how

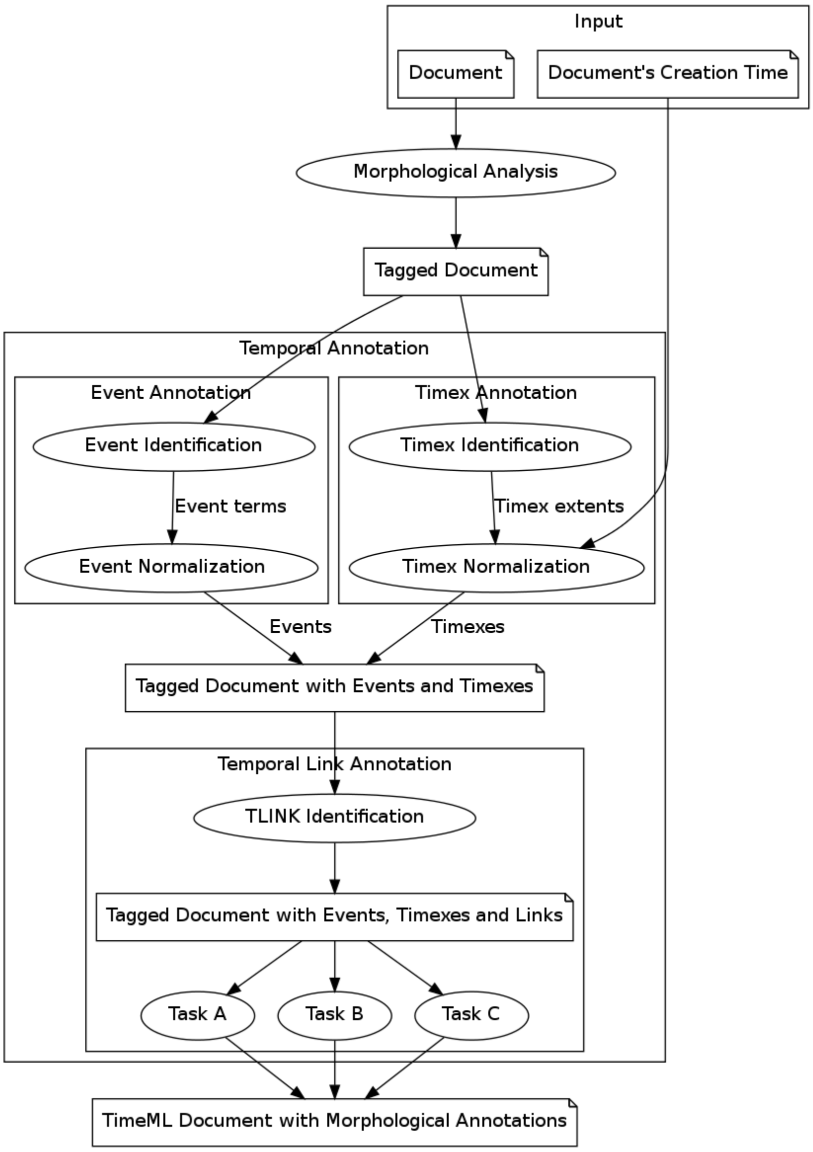


Figure 5.2: Architecture of LX-TimeAnalyzer

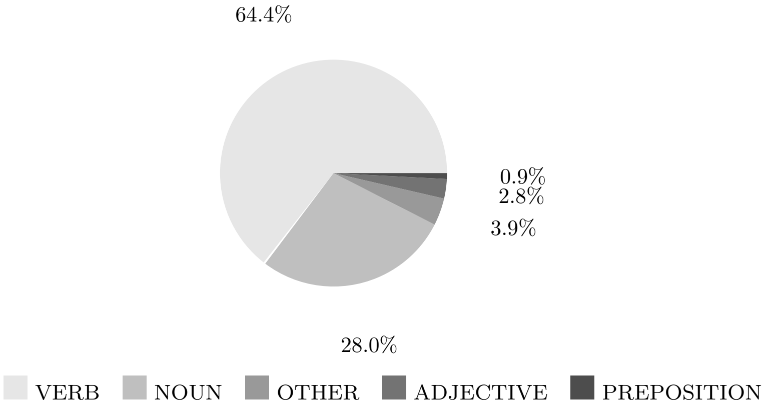


Figure 5.3: Part-of-speech distribution of event terms

the word is inflected, etc. This information is then used in the subsequent phases of processing, mostly as classifier features for machine learned classifiers used for the several tasks at hand. The values for the attributes pos, tense and stem of EVENT elements are directly taken from the output of LX-Suite.

Event Identification A simple solution to identifying event terms in text is to classify each word as to whether it denotes an event or not. This strategy is not very efficient, since some very frequent words cannot possibly denote events (e.g. determiners, conjunctions, etc.). Figure 5.3 shows the distribution of parts-of-speech for event terms, according to the training data. 92% of all event terms are verbs or nouns. Nevertheless, we followed this simple approach.

An important observation is that event terms are not just verbs and not all verbs are tagged as events (e.g. auxiliary verbs do not denote events). Training a decision tree with part-of-speech information as the only feature produces a model with an F-measure of 0.56 on the test data. This classifier just associates verbs with events. Additional features are thus needed.

The classifier features tried are the following:

- **Features about the last characters of the lemma**
  A Boolean attribute represents whether the lemma ends in one of several suffixes from a handcrafted list. This list includes suffixes such as *-mento* (roughly equivalent to the English *-ment*, as in *displacement*). The motivation is that this information may be useful especially to separate eventive nouns from

	Eventive Nouns	Non-eventive Nouns
Number	Count (%)	Count (%)
Singular	1,433 (74)	6,101 (59)
Plural	491 (26)	4,177 (41)
Total	1,924	10,278

Table 5.3: Distribution of singular and plural forms of eventive vs. non-eventive nouns in the training data

non-eventive nouns. There are additional attributes that include information about the last two characters of the lemma and the last three characters of the lemma; they are intended to capture suffixes not covered by the list of suffixes.

- **The part-of-speech and the inflection tag assigned by the tagger.**
  As shown in Figure 5.3, information about part-of-speech can rule out many words in a document. The inflection tag may also be relevant. For instance, even though singular forms are more common than plural forms for both eventive and non-eventive nouns, this difference is sharper in the case of eventive nouns, as shown in Table 5.3 (plural eventive nouns denote multiple or repeated events, which are possibly less commonly mentioned).

- **The part-of-speech and the inflection tag of: the preceding word token, the following word token, the preceding word token bigram, the following word token bigram**.
  These attributes are used in order to capture some contextual information. We consider both isolated words and word token bigrams (sequences of two words).

- **Whether the preceding token was classified as an event**
  The intuition is that adjacent event terms are infrequent.

Training a decision tree with these attributes (Weka's Witten & Frank (2005) implementation of the C4.5 algorithm was used) on the training data results in a classifier with an F-measure of 0.72 evaluated on the test data. As shown in Table 3.3 in Chapter 3, there are 6,790 annotated events in the training data (out of a total of 68,351 word tokens) and 1,097 event terms in the test data (from a total of 9,829 words), which is a considerable amount of data. Our result is somewhat worse than the best systems of TempEval-2 for both English (0.83) and Spanish (0.88). These systems followed a similar approach to ours, but they used additional classifier attributes based on the output of a syntactic parser (this was also tried, but it did not improve the results) and WordNet (which was not possible to use, as no similar resource for Portuguese is as comprehensive). We believe that the information taken from WordNet is probably the major cause of the differences, as the structure of WordNet can be used to discover event terms in texts. Because there is a synset for *event* "something that happens at a given place and time", extracting its hyponyms provides a very large list of nouns that can denote events. Indeed, the only systems in TempEval-2 achieving a result for this task above the 0.8 score resorted to the WordNet (Llorens *et al.*, 2010a; UzZaman & Allen, 2010).

Event Normalization This step is concerned with the annotation of the several attributes appropriate for TimeML EVENT elements, as described in Section 3.6.1.

The values of many of the attributes of EVENT elements are already provided by the morphological analyzer: stem, tense and pos. Three attributes are not, however: aspect, polarity and class.

A principled annotation of the polarity attribute (which encodes whether the event occurs in a positive or negative context) requires syntactic parsing. Nevertheless, we tried to simply check whether one of the three preceding words is a negative word (*não* "not", *nunca* "never", *ninguém* "nobody", *nada* "nothing", *nenhum/nenhuma/nenhuns/nenhumas* "no, none", *nenhures* "nowhere") and there is no other event intervening between this negative word and the event that is being annotated. On the test data, the accuracy of this simple heuristic is 98.9%, which is similar to the best score in TempEval-2 for English (99%) and better than the one for Spanish (92%).

In the Portuguese data, the attribute aspect only encodes whether the verb form is part of a progressive construction. This attribute is also computed symbolically, and the implementation simply checks for gerund forms (e.g. *fazendo*) or constructions involving an infinite verb form immediately preceded by the preposition *a* (*a fazer*). On the evaluation data, its accuracy is 95.6%.

The most interesting problem of event normalization is determining the value of the class attribute of EVENT elements. It is also the hardest, with the best system for English scoring 79% in TempEval-2 and the best one for Spanish correctly classifying only 66% of the test instances. This attribute of EVENT elements encodes information about aspectual type (Section 2.2.2), which is sensitive to both lexical and contextual (i.e. syntactic) information. For this attribute, a specific classifier was trained (also a decision tree, with Weka). This classifier takes advantage of a very minimal set of features:

- **The lemma of the event term being classified**
  This type of information is highly lexicalized, so it is expected that the lemma of the word token can be quite informative.

- **Contextual features**
  These attributes encode the part-of-speech of the previous word and that of the next word, and the following bigram of inflection tags.

We experimented with more features, similar to the ones used for event detection, but they did not improve the results. We obtained a result of 74%.

Temporal Expression Identification In order to identify temporal expressions, we trained a classifier that, to each word in the text, assigns one of three labels: B (begin), I (inside), O (outside).

Once again, a decision tree classifier was employed and the features are:

- **Features about the current token**
  This includes the token's part-of-speech and its inflection tag. Additionally, there is an attribute that checks whether the current token's lemma is part of a handcrafted list of temporal adverbs. This is specially useful for the B(egin) class, which is the one with the highest error rate.

- **Features about the previous token and features about the following one**
  Once again these features are taken from the morphological analyzer and encode part-of-speech and inflection tag.

- **The classification for the previous token**
  This is relevant because tokens classified as I(nside) cannot directly follow tokens classified as O(utside).

- **Whether there is white space before the current token and the previous one**
  The reason behind this attribute is to treat punctuation and special symbols in a special manner (they are tokenized separately; e.g. a time expression such as *2000-10-20* is tokenized into five word tokens).

- **Whether (i) the current token's lemma was seen in the training data at the beginning of a temporal expression, or (ii) it was seen inside a temporal expression, or (iii) the bigram of lemmas formed by the current token's lemma and the next one's was seen inside a temporal expression**
  Instead of using an attribute encoding the lemma of each word directly, we used a series of Boolean attributes capturing information about the lemma that are expected to help classification.

As shown in Table 5.2, this component shows an F-measure of 0.86 for recognizing words as being a part of a temporal expression, i.e. a member of the B(egin) or I(nside) classes.[1] The evaluation score is identical to the score for the best system of TempEval-2 working with the English data.

Temporal Expression Normalization This problems consists in identifying the value of the TIMEX3 attributes type and value.

LX-TimeAnalyzer solves this problem symbolically. The normalization rules take as input the following parameters:

- The word tokens composing the temporal expression, and their morphological annotation

[1] This is the way this task is evaluated in TempEval-2. That is, the evaluation does not differentiate between the B and I classes. This distinction is relevant only when two temporal expressions occur adjacently. This situation of two distinct temporal expressions being mentioned next to each other happens only once in the training data and never in the test data, so it has no consequence on this evaluation (which is on the test data).

- The document's creation time
- An anchor. This is another temporal expression that is often required for normalization. For instance, an expression like *the following day* can only be normalized if its anchor is known. We use the previous temporal expression that occurs in the same text and that is not a duration. This simple heuristic is similar to previous approaches found in the literature.
- The broad tense (present, past, or future) of the closest verb in the sentence where it occurs, with the distance being measured in number of word tokens from either boundary of the time expression. This broad tense is similar to the simplified tense employed in the previous chapter: for example, all past tenses are treated as *past*. This is used for instance in order to decide whether a time expression like *February* refers to the previous or the following month of February (relative to the document's creation time).

These rules are not implemented in a dedicated format, they are simply implemented in a Java method. It takes approximately 1,600 lines of code and is recursive: e.g. when normalizing an expression like *terça de manhã* "Tuesday morning", the expression *terça* "Tuesday" is normalized first, and then its normalized value is changed by appending TMO (with T being the time separator and MO the way to represent the vague expression "morning"); its type is also changed from DATE to TIME. The same method fills in both the value and the type attributes of TIMEX3 elements.

This implementation was conducted by looking at the examples in the training data, and additionally to a small set (c. 5,000 words) of news pieces taken from on-line newspapers. It has 91% accuracy for the type attribute and 81% accuracy for the value attribute, which is somewhat below the state of the art for English and Spanish in TempEval-2, but above the best results in HAREM for Portuguese, although these comparisons must be taken with a grain of salt due to the differences in the language or the data used.

Identifying Temporal Relations This sub-task is performed symbolically. For Task A Event-Timex, it is very simple: all temporal relations between the events and temporal expressions mentioned in the same sentence are considered. Since LX-Suite also detects sentence boundaries, finding these pairs of entities that are to be temporally related is trivial. Task B Event-DocTime is about temporal relations between an event and the document creation time, and it is also easy. We create a temporal relation for every event in every sentence. Task C Event-Event, which deals with temporal relations between the main events of two consecutive sentences, is not as straightforward.. It requires determining the main verb of a sentence, which in turn requires syntactic parsing. Instead of getting this information from a syntactic parser, we just consider the first mentioned event in a sentence to be its main event, for the purposes of Task C Event-Event.

Classifiers for TempEval Tasks: Task A Event-Timex, Task B Event-DocTime and Task C Event-Event The final sub-task, that of classifying the three types of temporal relations, is addressed by the classifiers worked out in Chapter 4.

5.3 Hybrid Temporal Processing: Integration with Deep Processing

The full-fledged processing of temporal information by deep grammars presents specific challenges that make this goal difficult to achieve. To a large extent, these difficulties stem from the fact that the temporal meaning conveyed by grammatical means interacts with many extra-linguistic factors, such as world knowledge, causality, calendar systems, and reasoning, among others. These have been identified in Chapter 4 as contributing to the difficulties in temporal relation classification. Since some of these factors are not well understood, deep grammars, which are rule-based, struggle to accommodate them.

This section presents a hybrid strategy that explores the complementarity of the symbolic and probabilistic methods in a way that their strengths can be amplified and their shortcomings mitigated. In concrete terms, the deep semantic representations produced by a deep processing grammar for temporal information is improved with the outcome of LX-TimeAnalyzer. LX-TimeAnalyzer itself already benefits from symbolic approaches: the temporal relation classification task described in Chapter 4 benefits highly from representations coming from syntactic parsers and enriched with temporal annotations created by a rule-based component, as these representations provide the best classifier feature for Task A Event-Timex. The extraction component is robust and draws from different sources of information. Machine learning makes it possible, as in many cases we do not know explicitly how these different factors combine to produce the final temporal meaning being expressed. The deep computational grammar delivers richer truth-conditional meaning representations of input sentences, which include a principled representation of temporal information, on which higher level tasks, including reasoning, can be based.

This section describes this integration effort and presents an evaluation of the resulting hybrid system. This approach shows performance results that increase the quality of the temporal meaning representations and improve the performance of each component in isolation.

5.3.1 Motivation

Deep linguistic processing aims at providing grammatical representations of sentences, including full-fledged semantic representations. This is undertaken by computational grammars whose handcrafted rules encode the regularities uncovered by theoretical linguistics. While these grammars typically deliver all precise linguistic relations and fine-grained semantic analyses that are possible for a given sentence, they perform less well when it comes to resolving ambiguity and getting at the appropriate representation of that sentence given its context of occurrence.

The inverse tension is observed in shallow processing systems. Resorting to statistical methods, these systems are much better at resolving ambiguity, but they perform much worse when it comes to get at the sophistication of deep semantic representations.

The linguistic expression of time forms a highly intricate semantic subsystem that offers a particularly good illustration of the complementarity between the two approaches and the gap to bridge. Like in any other grammatical dimension, here too ambiguity is pervasive, and each sentence in isolation may bear different temporal readings.

Deep grammars typically handle such proliferation of readings by resorting to some underspecification formalism that allows for its packing. Although this makes it possible to address the efficiency problems associated with this ambiguity, rule-based grammars offer limited means to subsequently resolve this ambiguity and to support real-world applications that need to rely on the actual information conveyed by sentences in their contexts.

As we have been seeing, the area of temporal information extraction has encouraged the development of systems able to extract from texts important pieces of information concerning time. But there is so far little or no exploration of how to integrate them into the deep principled semantic representations of the sentences, so that they can help support higher-level temporal processing and reasoning systems. In the opposite direction, the sophisticated linguistic relations encoded in deep representations that may be important to improve the accuracy of temporal information extraction are also waiting to be explored.

We explore the complementarity of the two approaches—symbolic methods and probabilistic ones—and also combinations of them that amplify their strengths and mitigate their drawbacks. To this end, linguistically principled and data-driven methods are integrated in multiple ways at different stages of processing. On the one hand, deep semantic processing is informed by shallower temporal information extraction procedures to resolve ambiguity and reduce underspecification. That is, LX-TimeAnalyzer is used as this module specialized in temporal extraction. On the other hand, data-driven temporal extraction, as materialized in LX-TimeAnalyzer, is already informed by high-level linguistic information, such as aspect (as described in Section 4.4.2) and syntax (Section 4.4.5).

5.3.2 Background: Deep Linguistic Processing

Two key elements will be integrated, with the purpose of combining temporal information extraction and deep semantic representations: a deep grammar, that produces such representations, and temporal information extraction technology, which identifies and normalizes events, dates and times mentioned in a text, as well as classifies temporal relations holding between these entities, i.e. LX-TimeAnalyzer, presented above in Section 5.2 and featuring the classifiers of temporal relations developed in Chapter 4.

Deep linguistic processing grammars associate each input sentence with its grammatical representation (including morphology and syntax) as well as a representation of its meaning (semantics). In this work, we use LXGram as the working grammar. LXGram is a deep grammar for Portuguese with a few years of development (Branco & Costa, 2010).

This grammar is based on the Head-Driven Phrase Structure Grammar (HPSG) grammatical framework (Pollard & Sag, 1987, 1994; Sag *et al.*, 2003). HPSG resorts to a unification based grammatical representation formalism with a type system featuring multiple inheritance and recursive data structures called typed feature structures. HPSG is a linguistic framework for which there is a substantial amount of published work. This allows for the straightforward implementation of well known grammatical analyses, which are linguistically grounded and have undergone scientific scrutiny. It also has a positive impact in reusability and extendability, because more people can understand it immediately. The HPSG literature has produced very accurate analyses of long distance dependencies, and a general strong point of computational HPSGs, among many others, is precisely the implementation of this key phenomenon of natural language syntax.

LXGram is implemented in the LKB (Copestake, 2002), an integrated development environment for typed feature structure grammars in general, popular within the HPSG community. The LKB provides a graphical user interface, debugging tools and very efficient algorithms for parsing and generation with the grammars developed there. Several broad-coverage HPSGs have been developed in the LKB; the largest ones are for English (Copestake & Flickinger, 2000), German (Crysmann, 2007) and Japanese (Siegel & Bender, 2002), but there are non-trivial LKB grammars for several other languages (http://wiki.delph-in.net). The grammars developed with the LKB are also supported by the PET parser (Callmeier, 2000), which allows for faster parsing times due to the fact that the grammars are compiled into a binary format first. PET also allows several input methods, including interfacing with external morphological analyzers, which we make use of (LXGram runs on the output of LX-Suite, presented above in Section 4.3.1). These systems also allow the training and use of a statistical model to discriminate between competing analyses for each sentence (Oepen *et al.*, 2002; Toutanova *et al.*, 2005; Velldal, 2007). The model is trained on a treebank. This facility is also used with LXGram to rank the parses produced for a given sentence. The grammar outputs all possible parses for a given input sentence, and this model selects the most probable one.

LXGram is built upon the core Grammar Matrix system (Bender *et al.*, 2002), which contains a set of implemented grammatical constraints relevant to many languages. The grammar employs Minimal Recursion (MRS; Copestake *et al.* (2005)) as the formalism for the semantic representations it produces. MRS is briefly explained below.

A main feature of MRS is that it supports underspecified semantic representation. An MRS representation is a tuple containing a global top, a bag of relations labeled with handles and a bag of constraints on handles. Relations labeled with handles are called *elementary predications*, but we will also refer to them as relations in this text. Conjunction is represented by shared labels. Handles can also appear as arguments of these relations, and they are used to represent scope. The main kind of constraint on handles is equality modulo quantifiers ($=_q$), which means that either the two handles are the same handle or a quantifier relation (but not another type of relation) intervenes between the two. They enable the underspecification of the scope between the various relations. An example MRS representation for the sentence *A black cat can fly* is:[1]

$$
\begin{array}{l}
<h1,\\
\quad \{h2 : \_a\_q(x3, h4, h5),\\
\quad\ h6 : \_black\_a(x3),\\
\quad\ h6 : \_cat\_n(x3),\\
\quad\ h7 : \_can\_v(h8),\\
\quad\ h9 : \_fly\_v(x3)\},\\
\quad \{h1 =_q h7, h4 =_q h6,\ h8 =_q h9\} >
\end{array}
$$

This representation corresponds to the two scoped formulas that can be obtained from it by scope resolution:

- $\_a\_q(x3,\ \_black\_n(x3) \wedge \_cat\_n(x3),\ \_can\_v(\_fly\_v(x3)))$
  (There is a black cat that possibly flies.)

[1] We follow the convention of including part-of-speech inspired labels in the names of the relations in an MRS representation: *n* for relations denoted by nouns, *a* for those related to adjectives and adverbs, *q* in quantifier relations, *v* in verbal relations, etc.

- $\_can\_v(\_a\_q(x3,\ \_black\_n(x3) \wedge \_cat\_n(x3),\ \_fly\_v(x3)))$
  (It is possible that there is a black cat that flies.)

This is how the scope ambiguity between the existential quantifier and the modal operator is captured. The first reading is obtained when the constraints on the handles are resolved this way: $h1 = h2,\ h4 = h6,\ h5 = h7,\ h8 = h9$. The second one is when $h1 = h7,\ h8 = h2,\ h4 = h6,\ h5 = h9$.

MRS representations are straightforwardly encoded in the typed feature structures manipulated by HPSGs. For the sake of readability of this text, we abstain from presenting them in that format.

For the sake of experimentation, a concrete grammar has to be used. The solutions put forth are tested with this working grammar but their principles can be easily adapted or transferable to other deep computational grammars delivering an underspecified semantic representation, developed under other grammatical frameworks or for other languages.

Existing computational HPSGs typically dodge the issue of tense and aspect entirely, deeming it to be too complicated to be worth implementation efforts. But because MRSs are used by applications and this sort of information is important, a common approach is to enrich the output MRSs with information about grammatical tense and aspect. For instance, the MRS representation for our sentence *A black cat can fly* could look like:

$$
\begin{array}{l}
<h1,\\
\quad\{h2: \_a\_q(x3, h4, h5),\\
\quad\ \ h6: \_black\_a(x3),\\
\quad\ \ h6: \_cat\_n(x3),\\
\quad\ \ h7: \_can\_v(e10\{tense\colon\ present\}, h8),\\
\quad\ \ h9: \_fly\_v(e11, x3)\},\\
\quad\{h1 =_q h7, h4 =_q h6,\ h8 =_q h9\} >
\end{array}
$$

Here, two event variables have been added to the relations for *can* and *fly*, an approach similar to that of Davidson (1967). These event variables can have features of their own. The one for *can* has a *tense* feature with the value *present*. This is an indication of the verb tense used in the verb form corresponding to this relation.

This approach has the disadvantage of mixing semantic information with morphological information. The motivation for our work is also to eliminate the need to include information about morphology in semantic representations, as far as tense and aspect are concerned.

5.3.3 Semantic Representation of Tense and Aspect

A semantic representation for tense and aspect was implemented in the grammar taking into account the possibility of it being extended with additional information relevant to time coming from temporal information extraction systems.

Related Work There is a vast linguistic literature on tense and aspect. Poulsen (2011) offers an overview of much of the literature on the linguistics of tense and aspect. Some of the work that is the most relevant to our present purposes is referred here, much of which has been described in more detail earlier in Section 2.2.

Davidson (1967) is the first author to reify events. In HPSG, this approach has been popularized in a number of analyses, including Sag *et al.* (2003), which is an introductory book to HPSG, as well as in several HPSG implementations, like the English Resource Grammar (Baldwin *et al.*, 2005) and the Grammar Matrix (Bender *et al.*, 2002). A survey of the advantages over the alternatives can be found in Kamp & Reyle (1993, pp. 504–10).

Reichenbach (1947) described tenses as temporal relations between several pairs of times, not just an event time and an utterance time (or speech time). In particular, he introduced the concept of a reference time that mediates the relation between those two times. This idea has been maintained in subsequent work by other authors.

Some influential ideas originating in Discourse Representation Theory (DRT), of Kamp & Reyle (1993), have also crept into many analyses of tense. This is the case of the observation that past tense denotes overlap of the event time with a past time in the case of stative situations but inclusion in the case of non-stative situations.

Intricately related to tense is aspect. A large body of literature exists on this topic, with the work of Vendler (1967) and Dowty (1979) being seminal.

Pustejovsky (1991) posits a separate level of representation for the event structure associated with predicates and their arguments and advocates the decomposition of events into sub-events. For instance, a sentence like *the door closed* is analyzed as a process (*the door closing*) followed by a state (*the door is closed*). This is similar in spirit to the work of Moens & Steedman (1988).

In the framework of HPSG, Van Eynde (2000a) develops an analysis for the Dutch tenses and temporal auxiliaries inspired by DRT in its semantic aspects. The work of Yoshimoto & Mori (2002) combines HPSG with a DRT analysis of tense. Bonami (2002) is an HPSG analysis of aspect shift inspired by the work of de Swart (1998a, 2000). This phenomenon is treated by positing implicit aspectual operators. Flouraki (2006) focuses on aspectual constraints on the various tenses of Modern Greek, modeling them with HPSG. Relevant to our work is that of Goss-Grubbs (2005), which develops an analysis of tense and aspect for English using MRS. This work encodes aspectual type by typing event variables, and it also resorts to positing explicit aspectual operators in the semantic representations. It does not make use of explicit temporal relations or the various Reichenbachian times (reference time, speech time, etc.); instead it encodes tense as a feature of time variables.

Relevant to our work is also that of Bobrow *et al.* (2007), in as much as it is about a computational system that produces meaning representations of its input which contain non-trivial information about time. In its representations, the system includes explicit temporal relations between events and the speech time. It does not, however, include information about aspect or make use of reference times.

5.3.3.1 MRS Representation

In connection to this, the grammar was extended with an implementation of tense and aspect inspired by much of the literature just referred to. The following running example illustrates the various aspects of the implementation:

(40) *A atriz mudou-se de França para os Estados Unidos em*
the actress moved from France to the United States in
fevereiro de 1947.
February of 1947
The actress moved from France to the United States in February 1947.

The MRS representation for this sentence, as produced by the grammar, is shown in Figure 5.4.

The following sections provide details on this representation as far as tense and aspect are concerned, describing the implementation of tense and aspect in the working grammar. Additionally, there is an implementation of backshift or sequence of tense in this grammar, also described below.

$$
\begin{array}{l}
<h1, \\
\quad \{h3\colon \_o\_q(x4, h5, h6), \\
\quad\quad h7\colon \_atriz\_n(x4), \\
\quad\quad h8\colon at(e2\ \{culmination\colon\ +\}, t9), \\
\quad\quad h8\colon before(t9, t10\ \{t\text{-}value:\ utterance\text{-}time\}), \\
\quad\quad h8\colon aspectual\text{-}operator(e2, e12, h11), \\
\quad\quad h11\colon \_mudar\_v(e12, x4), \\
\quad\quad h11\colon \_de\_p(e14, e12, x13), \\
\quad\quad h15\colon proper\_q(x13, h16, h17), \\
\quad\quad h18\colon named(x13,\ \text{"França"}), \\
\quad\quad h11\colon \_para\_p(e20, e12, x19), \\
\quad\quad h21\colon \_o\_q(x19, h23, h22), \\
\quad\quad h24\colon named(x19,\ \text{"Estados Unidos"}), \\
\quad\quad h11\colon \_em\_p(e26, e12, x25), \\
\quad\quad h27\colon udef\_q(x25, h28, h29), \\
\quad\quad h30\colon \_fevereiro\_n(x25), \\
\quad\quad h30\colon \_de\_p(e31, x25, x32), \\
\quad\quad h33\colon proper\_q(x32, h34, h35), \\
\quad\quad h36\colon named(x32,\ \text{"1947"})\}, \\
\quad \{h1 =_q h8,\ h5 =_q h7,\ h16 =_q h18,\ h23 =_q h24,\ h28 =_q h30, \\
\quad\quad h34 =_q h36\} >
\end{array}
$$

Figure 5.4: MRS for *A atriz mudou-se de França para os Estados Unidos em fevereiro de 1947* “The actress moved from France to the United States in February 1947”

5.3.3.2 Tense

It is important to distinguish between grammatical tense and semantic tense: we will use the first expression to refer to inflectional morphology alone, and the second one to refer to the temporal and aspectual meanings conveyed by grammatical tenses.

Each predicate denoted by a verb, adjective, preposition or adverb receives a Davidsonian semantic representation (Davidson, 1967; Parsons, 1990), with an event variable as its first argument. This variable is not explicitly quantified, but assumed to be bound by an existential quantifier. This is in line with a substantial amount of the HPSG literature, including computational implementations such as the English Resource Grammar (Baldwin *et al.*, 2005) and the Grammar Matrix (Bender *et al.*, 2002). An example is the predicate *_mudar_v* (for the verb form corresponding to English "move") in Figures 5.4 and 5.6 (below): its first argument ($e12$) is an event variable. Events have the type e in the grammar.

Additionally, an *at* relation is employed that relates this event variable with a temporal index (in Figure 5.4 this relation is labeled with $h18$). Temporal indices refer to times. In the existing literature on tense, some authors use quantified time variables, while other authors use free time variables. Partee (1973) presents arguments for a free variable approach. Our temporal indices are compatible with this approach. This temporal index occurring as an argument of the *at* relation represents the event time. Temporal indices have their own type t in the grammar, and a feature T-VALUE is appropriate for this type t. This feature locates the index in the time line.

Depending on the grammatical tense, there are then temporal relations between temporal indices, in the spirit of Reichenbach, who describes tense as temporal relations between three times: the event time E, the reference time R and the speech time or utterance time S. For our purposes, we do not need full Reichenbachian representations for many of the tenses: in some cases we will represent the temporal relation between the event time and the speech time directly, and say nothing about the reference time. For instance, we assume semantic present to be a temporal relation between S and E, in particular a temporal overlap relation. We follow Discourse Representation Theory (Kamp & Reyle, 1993, p. 541) in further assuming that the speech time is seen as punctual, which means that this overlap relation is more specific than just overlap, and it is an inclusion relation: the event time includes the utterance time.

In our example sentence in (40), the Portuguese verb is in the *pretérito perfeito* tense. The semantics of this tense is ambiguous between a simple perfective past (i.e. the situation occurred in the past and is culminated; in Reichenbach's system, E and R are simultaneous and R precedes S) and a present perfect (the situation has a resulting state that holds and is relevant at the present; in Reichenbach's system, E precedes R and R and S are simultaneous). Since it is not possible to underspecify this distinction in the semantic representations, there are two options: duplicate the number of analyses provided by the grammar for each verb with this tense in the input (this is the approach of Van Eynde (2000a), for Dutch, but it is computationally costly and does not seem justifiable as both representations essentially describe a past event); or use a simplified representation that covers both interpretations. We chose the second route. The event time is before the utterance time and, accordingly, there is a temporal relation *before* with the event time as its first argument in the MRS representations.

The second argument of the temporal relation $before$ is another temporal index with a T-VALUE specified to have the value *utterance-time*. This is how the speech time is represented. According to what has been presented so far, the relevant representation fragment is thus:

$$at(e2,\ t9)\ \wedge\ before(t9,\ t10\ \{\textit{t-value}:\ \textit{utterance-time}\})\ \wedge \\ \_mudar\_v(e2,\ x4)\ \ldots$$

That is, the event described by the form of the verb *mudar* "move" occurred in a time that precedes the utterance time. In this text, we will sometimes use s to represent the speech time, as short-hand notation for a temporal index with the value *utterance-time* for its T-VALUE feature, as in:

$$at(e2,\ t9)\ \wedge\ before(t9,\ s)\ \wedge\ \_mudar\_v(e2,\ x4)\ \ldots$$

Grammatical tense presents two levels of ambiguity that must be resolved:

- The same form can correspond to more than one grammatical tense. An English example is the verb form *put*, which can, for instance, be present tense or past tense. Portuguese also contains similar ambiguities, e.g. forms like *corremos* (“we run” or “we ran”).

- The same grammatical tense can cover more than one meaning when it comes to locating a situation in time. An English sentence like *I leave tomorrow* shows that present tense can refer to the future. Usually this tense locates an event in the present. Portuguese shows similar cases.

This two-fold ambiguity is accounted for by a two-layer analysis in the working grammar. The first layer consists in a set of rules that map surface form to grammatical tense. The second layer consists in a set of rules that map grammatical tense to semantic representations of tense. Both are implemented as lexical rules, i.e. unary rules that apply to single lexical items (verb forms in this case).

We distinguish between imperfective and perfective tenses as they occur in Portuguese as well as several other languages (e.g. the remaining Romance languages or Slavic languages[1] or Greek). This distinction interacts with aspectual type, as presented in Section 2.2.2 (our encoding of aspect in MRS representations in presented in the next section): perfective tenses constrain the whole clause to be telic whereas imperfective tenses constrain it to be atelic (Bonami, 2002; de Swart, 1998a, 2000; Flouraki, 2006).

We assume that present cannot be perfective and, similarly to Michaelis (2011), that languages without perfective vs. imperfective distinctions show ambiguity in the other tenses. The examples in (41) are hers and support this last claim.

(41) a. At the time of the Second Vatican Council, they *recited the mass* in Latin.

b. He lied to me and I *believed* him.

The highlighted phrase in the English sentence in (41a) is telic (cf. *They recited the mass **in** 20 minutes*), but the sentence can nevertheless have an atelic reading (i.e., *. . . they used to recite the mass. . .*). In (41b) the highlighted verb is lexically stative, but the clause where it occurs can have a telic reading (i.e., *. . . I believed*

[1] Slavic languages are usually analyzed not as having perfectivity distinctions in their tense system but rather as having perfective and imperfective verbs (i.e. most verbs come in pairs formed by a perfective verb and an imperfective verb). This distinction is irrelevant for this discussion.

Semantic imperfective present:	"...fuma"	("...smokes")
$at(e,t) \wedge includes(t,s) \wedge \ \_fumar\_v(e,x) \ldots$		
Semantic imperfective past:	"...fumava"	("...smoked")
$at(e,t1) \wedge overlap(t1,t2) \wedge before(t2,s) \wedge \ \_fumar\_v(e,x) \ldots$		
Semantic perfective past:	"...fumou"	("...smoked")
$at(e,t) \wedge before(t,s) \wedge \ \_fumar\_v(e,x) \ldots$		

Table 5.4: The meaning of some tenses

what he said at that one time). This sort of aspectual coercion is similar to the one found with the perfective and imperfective past tenses in languages where the difference between them is marked. Therefore, the English simple past must be ambiguous between a perfective and an imperfective past tense.

Similarly, future tense (or future constructions) is ambiguous in English as well as Romance languages with respect to perfectivity, in contrast to languages like Greek and Russian, that show perfectivity distinctions also in the future tenses.

The examples in Table 5.4 show the sort of temporal representation that we have in mind, using the verb *fumar* "smoke" for illustration. We leave future tense aside, as it adds nothing new to the discussion, although future tenses and future constructions are implemented in LXGram. As can be seen in this table, from the semantics for the imperfective past it does not follow that the event time does not overlap the speech time. This is indeed a possibility, and it contrasts with the perfective past. As the pair of sentences in (42) shows, the imperfective past can describe situations that still hold at present; similar sentences with the perfective past sound strange.

(42) a. O João ontem **estava** doente e hoje ainda está. (imperfective past)
John was ill yesterday and still is today.

b. ?? O João ontem **esteve** doente e hoje ainda está. (perfective past)
John was ill yesterday and still is today.

These representations are inspired by Kamp & Reyle (1993) and Van Eynde (1998). They also make use of several times: the event time (identical to Reichenbach's E), the location time (similar to R, but in their work it is the time described by temporal expressions modifying the verb) and the perspective point (similar to S), and assume an interaction between the meaning of tenses and the aspectual type of the verb (recall the aspectual type distinctions described in Section 2.2.2). In the case of the past tenses, these authors assume that the relation between the location time of a situation and the perspective point is determined by aspectual class. For states this is one of overlap. For non-stative situations this is, more specifically, one of temporal inclusion. It follows from the event time being included in the location time and the location time preceding the utterance time (the past tense semantics) that the event time also precedes the utterance time. This is essentially the simplified representation that we use here for the perfective past. Unlike these pieces of work, we do not make this distinction depend on the aspectual type of the verb but rather assume that it is the difference between imperfective and perfective tenses. It just happens that perfective tenses constrain the whole clause to be telic whereas imperfective tenses constrain it to be atelic (Bonami, 2002; de Swart, 1998a, 2000; Flouraki, 2006), which means that imperfective tenses trigger no aspect shift when they combine with states, and neither do perfective tenses when they combine with culminations or culminated processes. The following Portuguese examples, based on those in (41) above, motivate our departure from their analysis:

(43) a. Na altura do Concílio Vaticano II, recitaram a missa em Latim. (perfective)
At the time of the Second Vatican Council, they recited the mass in Latin (they did that just once).

b. Na altura do Concílio Vaticano II, recitavam a missa em Latim. (imperfective)
At the time of the Second Vatican Council, they recited the mass in Latin (they used to do that).

(44) a. Ontem acreditei nele. (perfective)
Yesterday I believed him (I believed what he said yesterday).

b. Ontem acreditava nele. (imperfective)
Yesterday I believed him (Yesterday I was under the assumption that he always speaks the truth).

The examples in (43) both exhibit the phrase *recitar a missa* "recite the mass", which is a culminated process (i.e. a telic situation). The sentences in (44) contain the stative verb *acreditar* "believe". In all cases there is a preposition phrase or an adverb that locates the described situations in time (i.e. this temporal expression identifies the location time). The examples with the perfective forms describe situations that happen only once and within the time interval referred to by these modifiers. The imperfective sentences describe situations that are more prolonged in time and may extend outside the boundaries of these intervals.

5.3.3.3 Aspect

Aspectual type is described for with the help of three Boolean features: CULMINATION (positive for culminations and culminated processes), PROCESS (positive for processes and culminated processes) and STATE (positive for states). These features are appropriate for event variables.

Even though aspectual type is also a lexical property, it is difficult to annotate it (Pustejovsky *et al.*, 2006), as mentioned before. In our implementation, we abstain from encoding aspectual type in the lexicon.

However, contextual (i.e. syntactic) constraints on aspect are indeed implemented. These are represented by aspectual operators, which are functions from situation descriptions to situation descriptions, and they appear as relations in the MRS representations.

For instance, we represent a function from state descriptions to culmination descriptions as $\textit{aspectual-operator}(e_2\{\textit{culmination} : +\}, e_1\{\textit{state} : +\}, X)$. Here, e_1 is a state, e_2 is a culmination, and X is the semantic representation for the state e_1. The event variable (e_2 in this example) of the resulting situation is included in the representation, since Davidsonian representations are being used. We also make use of an extra argument, which is just a pointer for the event variable of the argument (e_1 in this example), because it is useful when post-processing MRS representations.

We follow Bonami (2002) in assuming that all aspectually sensitive relations allow for at most one implicit aspectual operator. These implicit aspectual operators account for aspectual coercion. Therefore every context that allows aspectual coercion must introduce either zero or one aspectual operators in the semantic representation: zero if no aspectual coercion actually occurs, and one otherwise.

Because it is not possible to underspecify the number of relations in an MRS, one *aspectual-operator* is introduced in every aspectually sensitive context, although in general it is not specified which operator it is (in line with Bonami (2002)). That is, one underspecified operator is always introduced: this is the *aspectual-operator* predicate just mentioned. We assume that sometimes it stands for a dummy relation (i.e. the identity function), in the cases when no aspectual shift actually occurs.

Several elements are sensitive to aspectual type. Tense is one of them. Consider the two example sentences below. They correspond to the English sentence *Samuel had a son yesterday.*

(45) a. O Samuel teve um filho ontem.

b. O Samuel tinha um filho ontem.

The difference between the two is grammatical tense, but they also convey different temporal and aspectual meanings. In the first one the verb is in the *pretérito perfeito* (the perfective past). In the second one the verb is in the *pretérito imperfeito* (the imperfective past).

As already mentioned above, perfective aspect constrains the whole event to be telic (a culmination or a culminated process). Imperfective aspect constrains it to be a state in Portuguese. The first sentence means one of Samuel's sons was born yesterday, whereas the second one simply says that one of his sons existed yesterday.

The grammar assigns to the first sentence a semantic representation expressing this:

> $at(e\{culmination: +\},\ t) \wedge before(t,\ t2\{t\text{-}value:\ utterance\text{-}time\}) \wedge aspectual\text{-}operator(e,\ e2,\ ter(e2,\ X))$, where X is the representation for the verb's arguments.

This representation is similar to the one presented above in the discussion about tense, but it includes information about aspect as well. In particular, an *aspectual--operator* was added scoping over the relation for the main verb in this sentence. This operator is introduced in the semantics by the lexical rule responsible for semantic tense (together with the temporal relations seen in this MRS fragment), as tenses impose aspectual constraints at the clausal level (Bonami, 2002). The constraint that the event variable e is telic (its feature CULMINATION has the value +) also comes from the *pretérito perfeito* tense.

By contrast, the second sentence receives a representation like:

> $at(e\{state : +\},\ t)\ \wedge\ overlaps(t,\ t2)\ \wedge$
> $before(t2,\ t3\{t\text{-}value\ utterance\text{-}time\})\ \wedge$
> $aspectual\text{-}operator(e,\ e2,\ ter(e2,\ X))$, where X is the representation for the verb's arguments.

Unlike the *pretérito perfeito* tense, which introduces an aspectual operator that produces telic situations, the *pretérito imperfeito* constrains the whole clause to be a state. In this example, this is encoded in the event variable e, with its feature STATE constrained to have the + value.

The verb *ter* "have", instantiating the third argument of the *aspectual-operator* relation, is a state. Even though lexical aspect is not encoded in the grammar (and therefore there is no restriction on the aspectual features of $e2$) for the reasons mentioned above, our encoding of aspect at the syntactic level, as it was just illustrated, is important because it can capture distinctions such as the one illustrated by this pair of sentences.

Additionally, it can be straightforwardly extended with lexical aspect: if we knew that "have" is lexically a state, then the *aspectual-operator* in the second sentence is a function from states to states (i.e. it is the identity function, and does not change the basic meaning of the verb). The aspectual operator in the first sentence would be a function from states to telic situations. One such operator can be the inchoative operator, which is the correct reading for this sentence (i.e. the state begins to hold yesterday). This final step is not deterministic: for instance, in the example in (44a), we also find a coercion of a state into a telic situation caused by the perfective past, but in that example the result is not an inchoative interpretation, but a different kind of change in meaning. For this reason, we can not identify the exact aspectual operator in context, and we use this abstract *aspectual-operator* relation every time in the representations. This *aspectual-operator* can be seen as the supertype of all aspectual operators. The analysis of aspect coercion implemented in the grammar is essentially the same as the analysis of de Swart (1998a), de Swart (2000) and Bonami (2002).

The implementation of aspect in the grammar interacts with many elements that are sensitive to aspect: many verbs, which impose aspectual constraints on their complements (some examples are the progressive auxiliary, which combines with processes, but also verbs like *stop* and *finish*); durational adverbials (*for* adverbials, which combine with processes, and *in* adverbials, which combine with culminated processes, are widely studied with respect to this phenomenon), tenses (as just briefly illustrated), etc.

A full description of the semantics of all tenses implemented in the grammar would be tedious, but an example with the present tense can also be presented. A sentence like *O Samuel tem um filho* "Samuel has a son" receives an MRS representation along the following lines:

> $at(e\{state: +\},\ t)\ \wedge\ includes(t,\ t2\{t\text{-}value:\ utterance\text{-}time\})\ \wedge$ $aspectual\text{-}operator(e,\ e2,\ ter(e2,\ X))$, where X is the representation for the verb's arguments.

Here t is the event time, and $t2$ is the utterance time. The present tense is assumed to be an imperfective tense, similar to the past imperfective tense mentioned above: it is associated with an overlap relation, and constrains the clause where it occurs to describe a state. Like Discourse Representation Theory, we assuming that the semantic present is special in that this overlap relation is more specific than just overlap, and it is an inclusion relation: the event time includes the utterance time. Because the verb *ter* "ter" is a state lexically, this is another example where the aspectual operator involved is the identity function.

5.3.3.4 Backshift

LXGram contains an implementation of backshift for Portuguese inspired by the work of Costa & Branco (2012a).

The following pairs of sentences, adapted from Michaelis (2006), illustrate the phenomenon of backshift, visible in indirect speech. Each sentence in parentheses is the direct speech counterpart of the embedded clause in the same line:

(46) a. Debra said she **liked** wine. ("I **like** wine")

b. Debra said she **likes** wine. ("I **like** wine")

c. Debra said she **brought** the wine. ("I **brought** the wine")

d. Debra said she **had brought** the wine. ("I **brought** the wine")

When the matrix verb is a past tense form, the verb tenses found in the embedded clauses are sometimes different from the tenses used in direct speech (46a, 46d), but not always (46b, 46c). For instance, in this context we sometimes find the simple past instead of the simple present in English (46a). In this respect English is in sharp contrast with Russian, where present tense can be used in similar embedded contexts with the same meanings as the English sentences using the simple past (example from Schlenker (2004)):

(47) Petya skazal, čto on plačet. (present tense in the embedded clause)
Petya said that he was crying.

The same phenomenon is also visible in Portuguese:

(48) a. A Debra disse que **gostava** de vinho. ("**Gosto** de vinho")

b. A Debra disse que **gosta** de vinho. ("**Gosto** de vinho")

c. A Debra disse que **trouxe** o vinho. ("**Trouxe** o vinho")

d. A Debra disse que **tinha trazido** o vinho. ("**Trouxe** o vinho")

An initial observation is thus that English and Portuguese use tense in an absolute way (the embedded past tense in (46a) is used to locate a situation in the past), whereas Russian uses it in a relative way (the embedded present tense in (47) marks a situation that was present at the time that the situation in the matrix clause held). Based on similar data, Comrie (1986) argues that English exclusively uses tense in an absolute way. However, the example in (49), from Rodríguez (2004), shows that in some cases English also uses tense in a relative way. In this example, the past tense is associated with a situation that may hold in the future with respect to the speech time. The past tense here signals precedence with respect to the time of the event in the higher clause (which is in the future). The phenomenon is thus more complicated than a simple separation between languages that use tense in a relative fashion and languages that use it in an absolute manner.

(49) María **will tell** us after the party tomorrow that she **drank** too much.

The same is true of Portuguese:

(50) A Maria **dir**-nos-**á** amanhã depois da festa que **bebeu** demais.

Several verbs trigger tense shifts in their complement. Reporting verbs are often identified with this group, but other verbs, like belief verbs or verbs like *decide* or *remember*, create similar contexts.

The phenomenon is also known as transposition, sequence of tenses or *consecutio temporum*, although some authors use some of these expressions in a broader sense, encompassing constraints on the co-occurrence of tenses in the same sentence. We reserve the term backshift to refer to the more specific case of the complements of the class of verbs just mentioned. We focus on backshift, in this narrow sense. This is because backshift is more constrained than the general co-occurrence of different tenses in the same sentence. For instance, Rodríguez (2004) points out that relative clauses are temporally independent, as illustrated by the example in (51a). The same can be observed in Portuguese, as in (51b).

(51) a. Felipe spoke last night with a girl that was crying this morning.

b. O Filipe falou ontem à noite com uma rapariga que estava a chorar hoje de manhã.

Here, two past tenses are found, and the verb of the relative clause refers to a situation that temporally follows the one denoted by the matrix verb. In turn, in backshift contexts involving two past tense forms, the embedded tense never signals a time that temporally follows the time associated with the embedding tense, as the ungrammaticality of the sentence in (52) shows:

(52) * A Debra disse ontem à noite que trouxe uma garrafa de vinho hoje de manhã.
Debra said last night that she brought a bottle of wine this morning.

A novel account of backshift was developed and implemented in LXGram, and it is described in Costa & Branco (2012a), as mentioned earlier. Backshift is treated as the result of the combination of two dimensions. The first one is acknowledging that tense, as it is visible in morphology, is ambiguous, as Section 5.3.3.2 argues. The second dimension consists in classifying the meanings of the tenses along a number of lines: present vs. past vs. future; perfective vs. imperfective aspect, relative vs. absolute. The first two lines determine which kinds of temporal relations are involved in the meaning of tenses (inclusion, overlap or precedence relations), as we have just seen in the previous sections. The third line is how the arguments in these relations are chosen: absolute tenses always take the speech time as one of the arguments of one of these relations; relative times look at a perspective point, which can be the speech time or the time of another event, depending on the syntactic context.

This analysis contains novel aspects. It provides a very clean distinction between absolute and relative tenses, making it depend on the use of two features in its HPSG implementation. It correctly constrains the possible readings of past under past constructions depending on grammatical aspect, which no other theory of backshift explains. This point is mentioned more clearly at the end of the following presentation of our analysis of backshift.

Analysis of Backhift For the purpose of handling backshift phenomena, we separate semantic tenses into two groups: relative tenses and absolute tenses. The *absolute tenses* always refer to the utterance time directly: they introduce in the semantic representation a temporal relation with the utterance time as one of its arguments. In turn, the *relative tenses* introduce a relation with a perspective point as one of its arguments. This perspective point is the utterance time if the corresponding verb is the head of the main clause of a sentence.[1] This perspective point

[1] This perspective point is similar to the perspective point assumed by DRT. Assuming that, in the case of matrix clauses, the perspective point is always the utterance time is a simplification that we make here because we are only interested in describing backshift (i.e. embedded clauses). The following example, from Kamp & Reyle (1993), illustrates the issue:

(1) Mary got to the station at 9:45. Her train would arrive at 10:05.

The perspective point of the second sentence must be the event time of the first sentence, so that this example can be accounted for by saying that conditional verb forms and *would* + infinitive constructions convey a semantic future tense anchored in a past perspective point. More cases where the perspective point of a main clause does not coincide with the utterance time are presented in Kamp & Reyle (1993, p.595 and following ones). Since computational grammars process each is instead the event time of a higher verb, if that higher verb is a verb like *say*, triggering backshift.

For the HPSG implementation of such an analysis, revolving around this distinctive constraint of the perspective point and the utterance time, three features are employed: UTTERANCE-TIME, which represents the utterance time, or speech time; PERSPECTIVE-POINT, for this perspective point; and EVENT-TIME, for the event time. As mentioned before we use the type *t* for these features.

The event time is always the second argument of the *at* relation introduced in the MRS representations by the lexical rules responsible for the semantic tenses. These rules add this *at* relation, as well as the remaining relations between temporal indices that we associate with the different tenses, presented above in Table 5.4, in Section 5.3.3.2. They also add the relation for the aspectual operators described in Section 5.3.3.3.

The utterance time must be accessible at any point in a sentence, because adverbs like *yesterday* or *today* always refer to it (e.g. *today* refers to the day that includes the speech time). In HPSG, each word in a sentence is represented by an instance of the type *word* and each phrase by an instance of the type *phrase*. The feature UTTERANCE-TIME is unified in all words and phrases present in a feature structure representation of a sentence. Therefore, in each phrase, the UTTERANCE-TIME of the mother node is unified with that of each of its daughters. Similarly, in lexical rules, the UTTERANCE-TIME of the mother node is also unified with the UTTERANCE-TIME of the daughter node. Additionally, in the grammar's start symbol (i.e. the description of what constitutes a valid sentence), the features UTTERANCE-TIME and PERSPECTIVE-POINT are unified: the perspective point is thus the utterance time in matrix clauses.

Because some verbs like *say* trigger backshift in their complement, but other elements do not, the relation between an item's perspective point and that of its complement is controlled lexically, i.e. for each word. For most items (the default case) they are unified, but in the case of backshift triggering elements, the PERSPECTIVE-POINT of the complement is the EVENT-TIME of the head. In HPSG, the lexicon is an association between words and lexical types. The lexical types describe all the

sentence in isolation, cases like this are beyond the scope of our work.

grammatical and semantic properties appropriate for the words they are associated with in the lexicon. This relation between a verb's perspective point (or event time) and the perspective point of its complement is encoded in the lexical types.

The absolute tenses look at the feature UTTERANCE-TIME in order to find one of the arguments for the relevant temporal relation that they introduce in the semantics. The relative tenses look at the attribute PERSPECTIVE-POINT instead. As an example, the semantic perfective past tense is a relative tense. Consider the following examples:

(53) a. O Kim mentiu. "Kim lied."
$proper\_q(x,\ named(x,\ "Kim"),$
$at(e_1\{culmination:\ +\},\ t_1)\ \wedge\ before(t_1,\ s)\ \wedge$
$aspectual\text{-}operator(e_1,\ e_2,\ \_mentir\_v(e_2,\ x)))$

b. O Kim disse que mentiu. "Kim said he lied."
$proper\_q(x,\ named(x,\ "Kim"),$
$at(e_1\{culmination:\ +\},\ t_1)\ \wedge\ before(t_1,\ s)\ \wedge$
$aspectual\text{-}operator(e_1,\ e_2,\ \_dizer\_v(e_2,\ x,\ e_3)\ \wedge$
$at(e_3\{culmination:\ +\},\ t_2)\ \wedge\ before(t_2,\ t_1)\ \wedge$
$aspectual\text{-}operator(e_3,\ e_4,\ \_mentir\_v(e_4,\ x))))$

The second argument of the *before* relation associated with the semantic perfective past is not the utterance time (as has been presented so far) but rather the perspective point, because this tense is a relative tense. In the case of main clauses this perspective point is the utterance time (since the two features UTTERANCE-TIME and PERSPECTIVE-POINT are unified in the grammar's start symbol)—this is what happens in examples such as (53a), and it is also the case of the matrix verb in (53b). In the case of clauses occurring as the complement of verbs that trigger backshift, this perspective point is the event time of the higher verb. The example in (53b) is thus correctly analyzed as saying that the event of Kim lying precedes the saying event, as can be seen from the semantic representation provided in (53b).

By contrast, the semantic tense given by the English and the Portuguese present tense, in examples like (46b) and (54) below, is an absolute tense.

The semantic present carries an inclusion relation between the event time and another time. Because it is an absolute tense, this other time is always the utterance

Grammatical Tense	Semantic Tense
Presente (present)	Absolute (imperfective) present
Pretérito imperfeito (imperfective past)	Relative (imperfective) present
Pretérito imperfeito (imperfective past)	Relative imperfective past
Pretérito perfeito (perfective past)	Relative perfective past

Table 5.5: Mapping between some grammatical tenses and some semantic tenses, for Portuguese

time, regardless of whether it occurs in backshifted contexts or regular ones. Since the semantic absolute present, being an absolute tense, finds the second argument of the *includes* relation that it adds to the semantic representation in the feature UTTERANCE-TIME, this argument will always be the utterance time, even in backshift contexts, as in (54).[1]

(54) O Kim disse que está feliz. “Kim said he is happy.”
$proper\_q(x,\ named(x,\ "Kim"),$
$at(e_1\{culmination:\ +\},\ t_1)\ \wedge\ before(t_1,\ s)\ \wedge$
$aspectual\text{-}operator(e_1,\ e_2,\ \_dizer\_v(e_1,\ x,\ e_3)\ \wedge$
$at(e_3\{state:\ +\},\ t_2)\ \wedge\ includes(t_2,\ s)\ \wedge$
$aspectual\text{-}operator(e_3,\ e_4,\ \_feliz\_a(e_4, x))))$

We follow the strategy mentioned above in Section 5.3.3.2 of letting a grammatical tense be ambiguous between two or more semantic tenses. The relation between grammatical tense and semantic tense is shown in Table 5.5 for some Portuguese tenses.

The following examples illustrate each of the semantic tenses considered in this table under the influence of a higher past tense verb: the absolute present, denoting overlap with the utterance time, and represented by the Portuguese grammatical present in (55a); the relative present, signaling overlap with the perspective point, and materialized in the Portuguese grammatical imperfective past in (55b); the relative imperfective past, marking precedence with respect to the perspective point, associated with a stative interpretation of the clause and realized by the Portuguese

[1] The meaning of the “present under past” is not trivial (Manning, 1992), and we opt for a simplified view of it here.

grammatical imperfective past in (55c); and the relative perfective past in (55d), similar to the relative imperfective past but associated with telic situations instead of stative ones and denoted by the Portuguese grammatical perfective past. After each example, under parentheses, one finds its direct speech counterpart.

(55) a. O Kim disse que é feliz. ("Sou feliz")
Kim said he is happy. ("I am happy")
Absolute present

b. O Kim disse que era feliz. ("Sou feliz")
Kim said he was happy. ("I am happy")
Relative present

c. Ontem o Kim disse que era feliz quando era pequeno. ("Era feliz quando era pequeno")
Yesterday Kim said he was happy when he was a child. ("I was happy when I was a child")
Relative imperfective past

d. O Kim disse que já almoçou. ("Já almocei")
Kim said he already had lunch. ("I already had lunch")
Relative perfective past

In Portuguese, in backshifted contexts, the grammatical imperfective past is ambiguous: it can be a semantic relative present tense (denoting temporal overlap with the matrix event and corresponding to the grammatical present in direct speech), as in (55b), or a semantic relative imperfective past tense (marking anteriority with respect to the matrix event and corresponding to the grammatical imperfective past in direct speech), as in (55c).

The relative present signals a temporal overlap relation between the time of the event denoted by the verb used in this tense and the perspective point: this is the reading for the example in (55b), where the two events overlap. We give this relative present tense (denoted by grammatical past in backshift contexts) a semantic representation similar to that presented above for the absolute present tense (denoted by grammatical present), the only difference is that the perspective point is used as the second argument of the *includes* relation (it is a relative tense rather than an absolute one). This example is thus analyzed as saying that the event time for the event described in the embedded clause includes the time of the event introduced by the matrix verb.

It must be noted here that our analysis, implemented in LXGram, and just described here, makes a very strong prediction about the relation between perfectivity distinctions and the temporal relation between the two verbs in sentences like the ones in (55). More specifically, an embedded perfective past tense never allows overlap readings with the matrix event, because there is no semantic present tense associated with the grammatical perfective past. This is valid for both Portuguese and English (assuming, like Michaelis (2011), that the English grammatical simple past is ambiguous between a semantic perfective past and a semantic imperfective past). No other analysis of backshift found in the literature accounts for this.

Related Work Many analyses of backshift and sequence of tense can be found in the literature, some of which we describe briefly. Reichenbach (1947), in his famous analysis of tense as involving temporal constraints between the speech time S and a reference time R on the one hand and between that reference point R and the event time E on the other, mentions the *permanence of the R-point*: a sentence like * *I had mailed the letter when John has come* is ungrammatical because the temporal constraints between R and S are incompatible in the two tenses involved (the past perfect constrains R to precede S while the present perfect constrains them to be simultaneous).

However, Reichenbach did not develop a full account of backshift. A Reichenbachian analysis of this phenomenon is that of Hornstein (1991), that posits a sequence of tense rule which associates the speech time S of an embedded clause with the event time E of the higher clause. In this analysis a conditional form of a verb is considered to be, underlyingly, a future form, which is transformed into a conditional form in backshift contexts. As pointed out by Gutiérrez & Fernández (1994), this fails to explain why the two tenses combine differently with adverbs like *yesterday*. If the conditional form in (56b) is a future form in some abstract representation, (56b) should be ungrammatical just like (56a) is:

(56) a. * Juan asegura que Pilar asistirá ayer a la fiesta.
Juan affirms that Pilar will attend the party yesterday.

b. Juan aseguró que Pilar asistiría ayer a la fiesta.
Juan affirmed that Pilar would attend the party yesterday.

The work of Comrie (1986) suffers from the same problem, as it also consists in a sequence of tense rule that transforms the tenses found in direct speech into the ones found in reported speech.

According to Declerck (1990), when two situations are located in time, there are two possibilities: either both of them are represented as related to the time of speech (absolute use of the tenses), or one situation is related to the time of speech while the second is related to the first (relative use, in the second case). In the second case, the simple past simply denotes overlap with a previous situation. This is very similar to our proposal, but we classify the different tenses as to whether they are relative or absolute, whereas Declerck (1990) assumes both possibilities for all tenses and lets pragmatics disambiguate, but these pragmatic conditions are never made explicit.

For Stowell (1993), past morphology is like a "past polarity" item that needs to be licensed by a Past operator (that in English is covert) outscoping it. The Past operator is what conveys the temporal precedence constraints present in the semantics. Past morphology can be bound by Past operators in different (higher) clauses, which explains sentences like (55b). The analysis of Abusch (1994) is similar in spirit, but it resorts to semantic rather than syntactic constraints.

Like us, Michaelis (2011) also assumes that the English simple past is ambiguous between two tenses (a perfective/eventive one and an imperfective/stative one). Because of this, and similarly to us, she is in a position where it is possible to account for the interplay between aspect and tense—i.e. perfective past clauses in backshift contexts are always anterior to the main clause event—, which the rest of the literature on backshift cannot explain.

However, the author fails to notice that and instead analyzes examples like (57), which is hers, as an example of an embedded imperfective/stative tense (when its translation to other languages shows that it should be viewed as an instance of a perfective tense). She then tries to obtain precedence effects from constraints coming from this imperfective tense, by deriving from it a semantic content similar to that of the English present perfect, which the grammatical imperfective past never has in languages like the Romance ones.

(57) He said that he paid $2000 for his property in 1933.

This relation between aspect and the possibility of the two past under past readings had been noticed by Enç (1986), who associates it with lexical aspect. The author mentions that statives allow two interpretations, one of simultaneity (58a) as well as one of precedence (58b) with respect to the event in the main clause. In the same context, non-statives do not exhibit the two readings that statives do. They only allow the precedence reading, as in (58c).

(58) a. John remembered that Jane was not even eighteen.

b. John remembered that Jane was not even eighteen when he met her.

c. John remembered that Jane flunked the test.

As the following examples in Portuguese show, this contrast is dependent not on the lexical aspect of the verb but on the aspectual type of the entire clause, i.e. whether a perfective or imperfective tense is used (as they constrain the aspectual type of the clause, as mentioned above).

(59) a. O John lembrou-se que a Jane tinha dezoito anos. (imperfective)
John remembered that Jane was eighteen.

b. O John lembrou-se que a Jane tinha dezoito anos quando a conheceu. (imperfective)
John remembered that Jane was eighteen when he met her.

c. O John lembrou-se que a Jane teve dezoito anos. (perfective)
John remembered that Jane was (once) eighteen.

d. O John lembrou-se que a Jane chumbou no teste. (perfective)
John remembered that Jane flunked the test.

e. O John lembrou-se que a Jane chumbava no teste. (imperfective)
John remembered that Jane flunked the test (e.g. she flunked it every time she tried).

f. O John lembrou-se que a Jane chumbava no teste quando a conheceu. (imperfective)
John remembered that Jane flunked the test when he met her (e.g. she flunked it every time she tried).

These examples show the combinations of perfectivity and the two lexical aspect classes considered by Enç (1986). The clauses with perfective past tense forms can only be interpreted as describing a situation that precedes the matrix one. The ones with imperfective forms are ambiguous and allow both simultaneity as well as precedence readings. The precedence readings are easier when the temporal location of the situation is mentioned explicitly, hence the *when* clauses. Our analysis correctly describes this generalization.

The collection of papers in Lo Cascio & Vet (1986) is about tense phenomena, including sequence of tense phenomena. Particularly relevant are those of Lo Cascio (1986), Rohrer (1986), Lo Cascio & Rohrer (1986) and Rigter (1986). Lo Cascio (1986) distinguishes between deictic tenses (those directly linked to the utterance time) and anaphoric tenses (those linked to the utterance time indirectly). This is similar to our distinction between absolute and relative tenses. Our use of a perspective point draws on the work of Rohrer (1986), which is an analysis of backshift for French in Discourse Representation Theory. Like us, the author uses it to relate embedded tenses to the time of matrix situations. More specifically, "the time denoted by the event of the matrix sentence becomes the temporal perspective point of the complement clause". The perspective point is necessary for those cases when the main verb shows future tense and the embedded one shows a past tense, like examples such as (49) illustrate. In such cases, past tense merely indicates precedence with respect to the perspective point, but not necessarily with the utterance time.

Van Eynde (1998) is a DRT-inspired analysis of English tenses in HPSG that also discusses transposition or sequence of tenses. Although he considers data such as the sentence in (60), rather than data involving the complement clauses of verbs like *say*, the data are nevertheless very similar. In the second sentence of (60) the simple past is a semantic present relative to a past perspective point introduced in the first sentence. However, the author does not discuss the use of simple past tenses to convey temporal precedence with the perspective point in transposition contexts, a possibility that is clearly available in backshift contexts, as examples like (46c) show.

(60) Mary had been unhappy in her new environment for more than a year. But now she felt at home.

More generally, the treatment of tense and aspect in HPSG includes the work of Van Eynde (1994, 2000b), Bonami (2002), Goss-Grubbs (2005), and Flouraki (2006), among others.

5.3.4 Full-Fledged Temporal Processing

The previous sections described an implementation of tense and aspect in a computational grammar. Because these systems rely heavily on grammatical properties, the representations of time that they can produce are limited to what the grammar of a language says about time. In the case of a language like Portuguese, it mostly has to do with the grammatical tense of verbs. But as we have seen in the previous chapter, the extraction of temporal relations from natural language texts requires more than just grammatical knowledge.

This section describes how the deep semantic representations produced by the grammar (in Section 5.3.3) are integrated with the temporal information coming from the temporal extraction system (in Section 5.2), with the purpose of expanding these representations.

The temporal extraction system outputs information that can be combined with the semantic representations delivered by the grammar,yielding semantic representations enriched with more and better information about time. In some cases, it is preferable to compute these pieces of temporal information outside the grammar; in other cases it is not even possible to compute them in the grammar. One such example is the normalization of temporal expressions. The normalization of temporal expressions like *two days before* require the output of arithmetic operations: once its anchor date is determined, it is necessary to subtract two days from it; a calendar system is also required, so that we know that e.g. subtracting two days from March 1, 2012 gets us to February 28, 2012, but going back two days from March 1, 2011 gets us to February 27, 2011. Deep grammars are implemented with specialized description formalisms and in platforms that do not even make arithmetic operations available.[1]

Typically, those specialized grammatical formalisms have a number of characteristics: they are developed exclusively with grammatical modeling in mind and do not support operations that are not directly needed for this modeling; they are categorical (they let one say whether a sentence is either grammatical or ungrammatical, not whether it is better or worse than an alternative), thus making it difficult to represent gradient or statistical information; and, since computational efficiency is an important concern for these systems, many are very restrictive.[2] Another characteristic of computational grammars is that their context is limited, as they typically only look at one sentence at a time. Because of this, they do not have access to information present in other parts of the document, which temporal extraction systems can take advantage of.

The expression of time in natural language and its meaning representation make particularly good cases where these limitations can be felt, as these tasks deal with a number of aspects that require extra-linguistic knowledge and as such are difficult or even impossible to implement in their full breadth in these specialized formalisms: arithmetics and calendar systems (for the normalization of temporal expressions, as just mentioned), reasoning (temporal relations have several logical properties that can be exploited, such as the transitivity of temporal precedence), the modeling of world knowledge and pragmatics (where statistical information about what is usual or expected may constitute important heuristics to determining the chronological order of the described situations), etc. Note that all these different kinds of information are explored by the classifiers of temporal relations described in Chapter 4.

[1] This is the case of LXGram and all grammars implemented in the LKB. The LKB accepts a language called TDL—Type Description Language (Krieger & Schäfer, 1994)—, which has no support for arithmetic operations. By contrast, modern programming languages make arithmetic operations available, and it is possible to find for them good implementations of calendar systems—e.g. LX-TimeAnalyzer makes use of Joda-Time 2.0 (Section 4.4.4), which provides many calendar operations as well as many operations on time intervals.

[2] For instance, the LKB, where LXGram is developed, is very fast, but, for efficiency reasons, does not allow the direct encoding of many kinds of constraints that are standard in the HPSG literature (Melnik, 2005).

It is possible to use a temporal processing system in order to augment these semantic representations output by the grammar in the following ways:

- Extending the representations
  It is possible to add to the MRS representations output by the grammar further temporal information that the grammar does not have access to.

- Specifying the representations
  The MRS representations are in many points underspecified, and in some of these cases they can be made more specific.

- Correcting the specifications
  Since the grammar only looks at grammatical information but the temporal extraction system is sensitive to other kinds of information, it is often more accurate than the grammar in resolving time related ambiguity and as such its output can be used to correct the MRS representations.

The following paragraphs provide details on these aspects of our contribution. To that end we return to our running example, presented above in (40) and repeated here in (61).

(61)

A	*atriz*	*mudou-se*	*de*	*França*	*para*	*os*	*Estados Unidos*	*em*
the	actress	moved	from	France	to	the	United States	in
fevereiro	*de*	*1947.*						
February	of	1947						

The actress moved from France to the United States in February 1947.

The temporal annotation obtained by the temporal extraction system for this running example is displayed in Figure 5.5. That example shows two annotated temporal relations, namely an overlap relation between the moving event and the month of February 1947, and a temporal precedence relation between this event and the document creation time.

The semantic representation obtained by the grammar for this example is shown in Figure 5.4. The objective is thus to enrich this representation with the temporal annotations shown in Figure 5.5.

<TIMEX3 tid="t0" functionInDocument="CREATION_TIME" value="2012-01-10T15:00:00"/>
<s>*A atriz* <EVENT eid="e5">*mudou*</EVENT>*-se da França para os Estados Unidos em* <TIMEX3 value="1947-02" tid="t15">*fevereiro de 1947*</TIMEX3>*.*</s>
<TLINK lid="l2" eventID="e5" relType="BEFORE" relatedToTime="t0"/>
<TLINK lid="l3" eventID="e5" relType="OVERLAP" relatedToTime="t15"/>

Figure 5.5: Example text with (simplified) temporal annotations. The English translation is *The actress moved from France to the United States in February 1947.*

Extending the MRS representations The outcome of this combination is presented in Figure 5.6. As can be seen by comparing Figures 5.4, 5.5 and 5.6, there are several pieces of information that are incorporated into the resulting MRS representation.

The first one is the information about the document's creation time (the TIMEX3 element in Figure 5.5). Temporal extraction systems register when a document was created (in our example this is "2012-01-10T15:00:00"), which can be determined from meta-data or with heuristics. The grammar does not have access to it. This information can be incorporated in the MRS representations, specifying the utterance time. The normalized value for the document's creation time is used to fill in the T-VALUE of the temporal index for the utterance time. In Figure 5.6, this is the temporal index $t10$.

The second type of information to add is about temporal expressions. These are not detected by the grammar, as they cannot be normalized in the grammar anyway, by the reasons just mentioned. An argument is added to the relation for the head word of that expression that was identified as a temporal expression by the extraction system. This argument is instantiated with a temporal index whose T-VALUE feature contains the normalized representation of the time expression. In our example, the temporal expression *fevereiro de 1947* "February 1947" is originally given the MRS representation:

$$\begin{aligned}&< h27,\ \{h27\colon udef\_q(x25,\ h28,\ h29),\ h30\colon \_fevereiro\_n(x25),\\ &h30\colon \_de\_p(e31, x25, x32),\ \ h33\colon proper\_q(x32, h34, h35),\\ &h36\colon named(x32,\ \text{"1947"})\},\ \{h28 =_q h30,\ \ h34 =_q h36\} >\end{aligned}$$

An extra argument is added to the *_fevereiro_n* relation, with the label $h30$,

$$
\begin{array}{l}
<h1,\\
\quad \{h3\colon \_o\_q(x4, h5, h6),\\
\qquad h7\colon \_atriz\_n(x4),\\
\qquad h8\colon at(e2\ \{culmination\colon\ +\}, t9),\\
\qquad h8\colon before(t9, t10\ \{t\text{-}value:\ \text{"2012-01-10T15: 00: 00"}\}),\\
\qquad h8\colon aspectual\text{-}operator(e2, e12, h11),\\
\qquad h11\colon \_mudar\_v(e12, x4),\\
\qquad h11\colon \_de\_p(e14, e12, x13),\\
\qquad h15\colon proper\_q(x13, h16, h17),\\
\qquad h18\colon named(x13,\ \text{"França"}),\\
\qquad h11\colon \_para\_p(e20, e12, x19),\\
\qquad h21\colon \_o\_q(x19, h23, h22),\\
\qquad h24\colon named(x19,\ \text{"Estados Unidos"}),\\
\qquad h11\colon \_em\_p(e26, e12, x25),\\
\qquad h27\colon udef\_q(x25, h28, h29),\\
\qquad h30\colon \_fevereiro\_n(x25, t69\ \{t\text{-}value:\ \text{"1947-02"}\}),\\
\qquad h30\colon overlaps(t9, t69),\\
\qquad h30\colon \_de\_p(e31, x25, x32),\\
\qquad h33\colon proper\_q(x32, h34, h35),\\
\qquad h36\colon named(x32,\ \text{"1947"}\},\\
\quad \{h1 =_q h8,\ h5 =_q h7,\ h16 =_q h18,\ h23 =_q h24,\ h28 =_q h30,\\
\qquad h34 =_q h36\} >
\end{array}
$$

Figure 5.6: Extended MRS for *A atriz mudou-se de França para os Estados Unidos em fevereiro de 1947* “The actress moved from France to the United States in February 1947”

filled with a temporal index containing the normalized value for the temporal expression, as Figure 5.6 shows: $< h30\colon \_fevereiro\_n(x25, t69\ \{$*t-value* ”1947-02”$\}) >$.[1]

Finally, additional temporal relations detected by the temporal extraction system are incorporated in the MRS.

The only temporal relations originally present in the MRS representations are the ones directly related to verb tense, since the grammar only looks at grammatical information, due to the limitations mentioned above. These are always between an event and the utterance time or the event of the higher clause in the case of backshift phenomena (Section 5.3.3.4).

But temporal information systems can extract more temporal relations than those. These extra relations can be added to the MRS representations. In our example this is the *overlaps* relation between the event time *t9* of the moving event and the temporal index *t69* for the time conveyed by the temporal expression *fevereiro de 1947* “February 1947”: $< h30\colon overlaps(t9, t69) >$.

[1] believe it can be improved. However, this issue is far from trivial, although it may seem so at first. The intuitive alternative would be to replace the entire material in the original MRS for this temporal index. In this example, the five relations (and the two handle constraints) for the expression *fevereiro de 1947* “February 1947” would be completely eliminated from the MRS and replaced by a temporal index. This temporal index would occur as the second argument of the *_em_p* relation, for the preposition corresponding to English *in*: $\_em\_p(e26, e12, t69\{$*t-value*”1947-02”$\})$. This alternative has two problems that must be noted.

The first one is illustrated by a sentence like *2007 saw the birth of the iPhone*. Here, a temporal expression occurs as the subject of a verb. With the intuitive representation, the first argument of the predicate for the verb *to see* would end up being a temporal index. This seems wrong, as the first argument of that predicate would not be of the expected type.

The second problem is related to examples like *that awful year*. This is a time expression that includes material (namely the adjective *awful*) that is not present in the normalized value of the temporal expression (which would just consist of a number representing a calendar year). Replacing the entire MRS representation of this noun phrase for a temporal index would create a representation that does not include all the information present in the analyzed input sentence.

We believe that the problem of adequately modeling the semantic representation of temporal expressions is an interesting question for linguistics to further clarify, for these reasons. As such, an admittedly simplistic solution was chosen in our integrated representation.

Increased Semantic Specification The temporal relations identified by the grammar can be made more specific on the basis of the output of the temporal information processing system. One example illustrating this deals with the following sentence, taken from the training data of TimeBankPT, with the original English sentence also presented below in italics:

(62) Esperava-se que Bush autorizasse os comandantes navais a usar "a mínima força necessária" para interditar os navios de carga para o Iraque e a partir do Iraque, disse um oficial americano.
Bush was expected to authorize naval commanders to use "the minimum force necessary" to interdict shipments to and from Iraq, a U.S. official said.

In TimeBankPT (and in the English data set used in TempEval), there are TimeML annotations for this sentence describing temporal relations between the document's creation time and several events, namely those represented by *esperava-se* "it was expected", *usar* "use", and *disse* "said". Similarly, the temporal extractor is capable of identifying these temporal relations.

The temporal semantics implemented in the grammar also encodes several temporal relations between situations described by finite verb forms and the speech time, which is similar to the document's creation time. However, in some cases, these semantic representations are less specific than the TimeML annotations.

A case in point is the imperfective past tense in indirect speech contexts, which is exemplified in this sentence with the verb form *esperava* "was expected". Here the semantics will encode that the event signaled by *esperava* overlaps the one given by *disse* "said" (cf. Section 5.3.3.4). This is expected, because this tense is associated with these kinds of readings in this context.[1] This semantic representation does not say anything about the relation between the embedded situation and the speech

[1] "Past under past" constructions (Abusch, 1994; Comrie, 1986; Declerck, 1990; Hornstein, 1991; Michaelis, 2011; Stowell, 1993) are ambiguous in English. For example, in *John said he was ill* the two situations described can be simultaneous, but in *John said he fell down* the one described by the embedded verb precedes the one in the matrix clause. In Portuguese, the two interpretations are distinguished by the past tense used: the imperfective past is used in the former case, and the perfective past is used in the latter one (Section 5.3.3.4).

time or document's creation time. This is not a shortcoming of the implemented grammar, it is what is justified from the point of view of the linguistic analysis. But this information is readily available in the output of the temporal extractor, and therefore can be incorporated in the final MRS representation.

Another case that is not trivial to treat in the grammar alone is the conditional forms of verbs. The grammar implementation assigns them a future of past interpretation: the described event occurs at a time that follows another time that precedes the speech time. Therefore, the direct relation between events introduced by verb forms in this tense and the speech time is not available in the MRS representation produced by the grammar, and in fact can be any one.

In the annotated data, however, there are cases of temporal annotations between events introduced by verbs in the conditional and the document's creation time.

Corrections to the temporal representations In some cases, the temporal extraction system can be used to correct the MRSs output by the grammar.

In cases of conflict between the initial temporal relations identified by the grammar and the ones given by the temporal extraction system, the initial representations produced by the grammar can be corrected if the temporal relations identified by the extractor are considered more reliable than the ones that the grammar produces.

This is because the grammar only looks at grammatical tense, whereas the temporal information system takes other features into account, and can identify cases where grammatical tense is misleading. An example of this is the case of the historical present, that is, the grammatical present being used to describe a past event, such as in the sentence *In 1939 Germany invades Poland.* This is an important property of our proposal.

Another example where corrections are fruitful is also connected to the use of present tense in Portuguese. English allows this tense to be used to describe future events, as in *The train leaves tomorrow.* In Portuguese this is much more pervasive, and because of that each occurrence of this tense is given this reading, as well as a present reading, by the grammar. The representations for the two different readings (present and future) are not underspecified. Rather, each occurrence of this grammatical tense is ambiguous between present and future, triggering two distinct analyses. As mentioned before, the system uses a statistical model to discriminate between competing analyses for each sentence. By causing the analysis to branch out in these cases, the choice of present vs. future is determined by this parse selection model.

Not surprisingly, as far as this distinction goes, this parse selection model performs quite poorly when compared to a dedicated temporal annotation system, as shown below in the next section. That is, there are several cases when the best interpretation given by the grammar erroneously assigns future semantics to present tense verb forms or vice-versa. In these cases, the integration component corrects the final MRS representation by changing the temporal relations there so that it is in accordance with the output of the temporal extractor.

5.3.5 Evaluation

A test suite of sentences exemplifying the phenomena that the grammar should be able to deal with was created. It contains sentences in the various tenses, sentences with forms of the auxiliary *ter* “have” combining with a past participle, sentences with a progressive construction similar to the English construction composed of *be* and an *-ing* form, sentences with forms of *ir* “go” with an infinite (similar to English “going to” constructions), and sentences featuring adverbs like *hoje* “today”, *ontem* “yesterday”, and *amanhã* “tomorrow”, which feature different combinatorial possibilities with the different tenses. This test suite is used for regression tests during grammar development and contains 38 sentences. The grammar is able to correctly parse all of these sentences and provides correct temporal representations for them.

The test suite is useful to check for bugs in the implementation and guarantee that the expected results are seen, but it might not be representative of what is seen in practical scenarios. So an evaluation with unseen data was conducted.

Evaluating this approach presents challenges. There is no gold-standard available with MRS annotations that contains temporal information similar to what is presented here. In an effort to create such a data set, it is quite difficult to produce MRS representations manually, as they contain many reentrancies. For these reasons, we resort to manual evaluation. Since the temporal extractor was developed using the train set of TimeBankPT, the test part of this corpus is unseen and can be used for evaluation of the integrated solution as well.

To this end, the 20 documents comprising the test portion of TimeBankPT were parsed with the grammar. On large corpora of native Portuguese text taken from newspapers and the Wikipedia, the grammar is capable of analyzing around 1/3 of all sentences (Branco & Costa, 2010). In the present case, 24% of the sentences in the test set of TimeBankPT got a parse.[1] Since the integration of the grammar with the extractor is not meant to increase the coverage of the former, the sentences that receive no parse were left out of this evaluation exercise. There remain 84 sentences in the test set.

This section provides evaluation results for the several tasks directly involved in the integration of the grammar with the temporal extraction system. First, the recognition and normalization of temporal expressions is discussed. This task is performed by the temporal extractor and then combined with the MRS representations output by the grammar, as discussed above. Here the results for the integrated output are the same as those for the temporal extractor.

After that, evaluation results are presented for two problems that are similar to the Task A Event-Timex and Task B Event-DocTime of TempEval discussed above. Since the temporal extractor identifies events and temporal expressions and temporal relations between these, and these temporal relations are added to the MRS representations, the performance of the extractor and that of the integrated system are discussed. Finally, evaluation results are provided for the classification of temporal relations between events and the speech time or the document's creation time (i.e. Task B Event-DocTime of TempEval). In this respect both the grammar and the temporal extractor are evaluated in isolation, since each can output these temporal relations. The integrated system, which corrects the MRS representations with the information coming from the extractor, is also evaluated.

Task C Event-Event of TempEval is not used by our integrated approach. Since Task C relates events mentioned in different sentences, a discourse representation

[1] We assume that this lower coverage it due to the fact that many of the documents composing this data set are taken from the Wall Street Journal (as TimeBankPT is a translation of the English corpus used in TempEval), and there was no effort to have the grammar deal with text from the financial and economic domains, which contain quite a number of syntactic idiosyncrasies.

	Grammar	Extractor	Integrated System
Recognition of temporal expressions	n/a	88%	88%
Normalization of temporal expressions	n/a	84%	84%
Task A Event-Timex	n/a	57%	57%
Task B Event-DocTime, finite verb forms	75%	83%	94%

Table 5.6: Accuracy of the grammar, the temporal extraction system and the integrated system for several tasks, evaluated on the parsed sentences of the test data of TimeBankPT. N/a marks results that are not available as the grammar is not intended to perform the corresponding tasks

is necessary to combine them in an informed way. This is not something that the typical deep linguistic technology does at the moment.

Table 5.6 summarized the results discussed in the rest of this section.

Evaluation of temporal expression recognition and normalization Since the integrated system enriches the original MRS representations with representations for the temporal expressions that occur in the underlying text, this aspect was evaluated.

As mentioned above, we restricted our attention to the sentences for which there was a parse produced by the grammar. We looked at all temporal expressions that can be found in these sentences. The system was evaluated with respect to two factors. First, we want to know how many temporal expressions are recognized correctly. Second, we also want to know how they are normalized, since these normalized values appear in the final representations.

Temporal relations are somewhat infrequent and, in these 84 sentences, only 32 such expressions occur. Of these, 88% are recognized correctly. The remaining ones are either not recognized at all or their boundaries are not identified correctly. 84% are recognized correctly and also normalized correctly (or 96% of the ones that are recognized correctly). From the point of view of normalization, the difficult cases are very vague ones such as *the night* or *the day*. These cases fail to be normalized and as such are not incorporated in the final MRS representations.

Although some of the temporal expressions occurring in this data set fail to be recognized and incorporated in the final MRS representations, the ones that are indeed inserted there are almost all correctly normalized (96%).

Evaluation of temporal relations between mentioned times and events As mentioned above, the final MRS representations also include temporal relations between the times and dates and the events mentioned in the input sentences, since these relations are delivered by the temporal extractor (cf. Task A Event-Timex of the first TempEval).

These temporal relations occurring in the semantic representations of the parsed sentences were checked for correctness. There are only 45 such relations, because only a few sentences contain multiple temporal expressions and multiple events. 57% of these relations are correctly encoded. A considerable number of the errors occur when the times and events being related are mentioned very far apart in the sentence or the syntactic relationship between the expressions denoting them is not direct. If we restrict our attention to pairs of events and times that are mentioned in the same clause, this score goes up to 68%.

Since the grammar provides us with this information, we are considering only adding these temporal relations to the MRS representations in these cases when the relevant expressions occur in the same clause. So even though temporal information processing technology still has a considerable amount of error, to some extent we can at least increase precision by sacrificing recall in a straightforward way if this is considered preferable.

Evaluation of temporal relations with the speech time One final aspect to evaluate is how many of the temporal relations between events and the speech time or document's creation time, output by the final integrated temporal processing system, are correct. This is similar to the Task B Event-DocTime of TempEval.

The grammar assigns temporal relations to events and states represented by finite forms of verbs only, for the reasons already mentioned. TimeBankPT includes annotations also for events denoted by words of other parts-of-speech, most importantly nouns. Even though the extractor can also identify these, it is sub-par in doing so, as mentioned above. For this reason, the integrated system does not expand MRS representations with temporal information for events that are not given by verbs, and likewise we also ignore them in this evaluation.

For each sentence, only the preferred parse output by the grammar, as determined by the parse selection model, is considered. The grammar produced a correct output for 75% of all temporal relations between the situations described by verbs in these parsed sentences and the document's creation time/speech time.

As mentioned above, one difficulty is assigning the correct meaning to present tense verb forms. As they are ambiguous between future and present semantic values and this distinction is chosen by a general parse selection model, it is often incorrectly resolved. The temporal extractor is much better at this particular problem, as it employs several features that are relevant to it. For instance, aspectual type is very relevant—depending on the language, the future interpretation of present tense is much harder or even impossible with stative verbs (Van Eynde, 1998, p. 249). The grammar has no information about lexical aspect, but the extractor has some, in the form of the aspectual indicators as well as the feature class. This problem accounts for 56% of the errors produced by the grammar for this task. Other errors were less interesting and had a smaller impact overall.

The temporal extractor gets 83% of these temporal relations between finite verb forms and the speech time/document's creation time right, better than the 75% of the grammar. The largest source of error has to do with identifying events: many of the verbs for which the grammar produces temporal relations are not recognized as events by the temporal extractor, and therefore no relation is posited for them. Note that TimeML does not annotate verbs used in generic statements (such as *Lions are mammals*, *Sony produces electronic devices*) as events, and furthermore the annotations for event terms that occurred fewer than 20 times in the English data used in TempEval were removed. Therefore the training data of TimeBankPT, which is also used to train the event identification model used in LX-TimeAnalyzer, contains many examples of verbs that are not annotated as being event terms.[1]

[1] As a side note, if one removes these cases and looks only at those that were identified by both the grammar and the temporal extractor, the success rate of the later in classifying the temporal relation with the document's creation time goes up to 97%. This is substantially better than the results presented in Chapter 4 for the Task B Event-DocTime of TempEval because here we are looking exclusively at events denoted by verbs, which are easier to order with respect to the utterance time than those given by words with a different part-of-speech.

The system combining the output of the grammar and that of the temporal extractor corrects the temporal relations identified by the grammar according to the output of the temporal extractor, and leaves them unchanged when the temporal extractor agrees with the grammar or does not identify them. It delivers temporal relations between finite verbs and the speech time/document's creation time with 94% accuracy. This is a better result than either the grammar (75%) or the temporal extractor (83%) in isolation.

This result shows that integrating a specialized temporal extractor with a deep grammar can be fruitful in practice in increasing the quality of the temporal meaning representations.

5.3.6 Discussion

The integrated approach described in this section is a novel contribution to the processing of the linguistic expression of time by means of the integration of data-driven and linguistically principled methods at different stages of processing. To this end, it was discussed how to enrich temporal extraction with deep linguistic information on aspectual type and how to combine the outcome of this temporal information extraction system with the semantic representations produced by a deep processing grammar.

This combination helps to resolve the ambiguity preserved in the underspecified semantic representation. It also allows for the representations produced by deep grammars to encode extra-linguistic information—e.g. the normalized representation of the speech time—that is relevant to interpret these representations but hard to obtain with these grammars alone.

The output of the deep grammar is enhanced with the output of the temporal extraction system, and this system is informed in different ways: with lexical information that is difficult to encode manually, such as the data-mined information about lexical aspect (Section 4.4.2), and with extra-linguistic information about world knowledge (Section 4.4.3), the calendar system and reasoning (Section 4.4.4). As such, these kinds of information, that are very difficult to integrate in a deep grammar, are eventually reflected in the semantic representations without the grammar having to handle them (if this happens to be possible at all).

Finally, with the present contribution towards a full-fledged processing of time, our work adds to the overall discussion and quest on how to obtain progresses in natural language processing by means of hybrid systems that combine the complementarity of the symbolic and probabilistic approaches in a way that their strengths can be amplified and their shortcomings mitigated.

5.4 Summary

The current chapter has two main goals. The first one is to integrate the classifiers of temporal relations developed in the previous chapter in a temporal processing system that is able to extract full temporal information from raw text in Portuguese: temporal expressions, events, and temporal relations between them.

The second goal is to show the utility of this temporal processing system in an application. To this end, we focused on the integration of the temporal processing system with a deep grammar. This grammar already produces meaning representations of input sentences, but they lacked temporal information. The grammar is extended with an analysis of tense and aspect that allows it to produce representations featuring some information about time, and then temporal processing technology is used to enrich these representations with further information. Even for some very simple problems, like determining the temporal relation between situations denoted by verbs and the time the sentence was uttered (for which verb tense is a strong cue), this integrated approach shows improvements. In fact, the integrated result is better than that produced by the grammar or the temporal extractor in isolation.

The result is a combination of a computational implementation directly based on linguistic theory with a data-driven component. This combination offers the best of both approaches: from the deep grammar, detailed meaning representations are extracted that take into account many details of the grammatical properties of language. The phenomenon of backshift, a fine-grained analysis of which was presented here, is an example of this that is relevant for the processing of time. With the data-driven temporal extraction component, one can explore extra-grammatical sources of information that can help the problems of temporal extraction even when they are not well understood. The resulting combination thus takes advantage of both approaches, benefiting from what each one has to offer in order to solve the problem of the temporal processing of natural language.

Chapter 6

Conclusions

This dissertation focused on the problem of extracting temporal information from texts written in Portuguese.

In this final chapter, in Section 6.1, we present a short overview of what is covered in each of the chapters making up the present text. The main goals and contributions of our work are reviewed in Section 6.2, and our results are assessed. The insights and conclusions gained from this research are summarized in Section 6.3. Section 6.4 identifies directions for future work, and concluding remarks are made in Section 6.5.

6.1 Summary of the Thesis

The contents of the present thesis can be summarized as follows.

Introduction Chapter 1 is a general introduction to the area of temporal processing, its applications and the challenges inherent to it. It also presents the way this dissertation is organized, its goals and its main contributions.

Related Work Chapter 2 introduces some of the most important work in the field of temporal information processing. Some fundamental concepts are introduced in this chapter, as well as some views about the way time is mentioned in natural language. Reichenbach (1947) describes the various verb tenses of English by considering three salient times—the speech time (when the sentence is uttered), the event time (the time when the event denoted by the verb happened) and the reference time. He then describes the meaning of the various grammatical tenses through temporal relations between the speech time and the reference time and between the reference time and the event time. For instance, the English simple past is viewed as conveying that the time of the event associated with the verb is simultaneous with a reference time that precedes the speech time. Vendler (1957) and Dowty (1979) are important pieces of work on aspectual type: situations can have different temporal structure: some (like *John was ill yesterday*) are homogeneous, holding in every subinterval of the interval in which they are reported to be true; others have a natural endpoint (as in *John ate a whole cake yesterday*), etc. Prior (1957, 1967, 1969) developed a calculus to reason about situations bound in time. His work extends traditional logic with four operators that refer to time, allowing some inferences about time to be formalized. Allen (1983, 1984) describes a comprehensive set of temporal relations between intervals and postulates rules that describe which inferences are possible from sets of these relations.

This same chapter then focuses on computational work, mentioning several challenges that have been put forth recently, as well as the data sets that they have used and the solutions that have been found by using these data sets. The Message Understanding Conferences (MUC-6, 1995; MUC-7, 1998) eventually took an interest in time expressions as part of named entity recognition tasks. This sort of task gained importance on its own, motivating the Temporal Expression Recognition and Normalization (TERN) challenge in 2004 (Ferro *et al.*, 2004). Since then, an interest has developed in more detailed annotations of time and the automated extraction of more phenomena related to time from text. The TimeML specification (Pustejovsky *et al.*, 2003a), setting the standard for annotations of natural language data related to time and events, has matured; data sets such as the TimeBank (Pustejovsky *et al.*, 2003b), a corpus of English text with temporal annotations, have surfaced; and competitions like the two TempEval challenges (Pustejovsky & Verhagen, 2009; Verhagen *et al.*, 2007, 2010), focusing on extracting temporal information from unstructured documents, have been conducted. In all of these, the focus has shifted to temporal relations between events and times or dates (*What has happened before what, after what, or simultaneously with what?*). In addition to identifying temporal relations implicit in texts, there has also been a focus in identifying mentioned events, dates and times, i.e. the entities that are part of these temporal relations. The research in this field has been dominated by machine learning approaches and has focused mostly on English. Work on the temporal processing of other languages has started, with the appearance of annotated data sets for Chinese (Cheng *et al.*, 2008), French (Bittar *et al.*, 2011), Korean (Im *et al.*, 2009), etc.

Data Chapter 3 presents the data set developed to be used to experiment with temporal information processing solutions for Portuguese. This corpus is TimeBankPT, which was developed by adapting the English data set used in the first TempEval to the Portuguese language. The temporal annotations that are used in the TempEval data and in TimeBankPT were described. Some shortcomings of the original resource were mentioned—low inter-annotator agreement, some difficult instances in the test data and few training instances for some of the classes—, which should be kept in mind when interpreting results from tools or solutions that resort to the data of TempEval—and consequently the results based on TimeBankPT as well.

We explained how this adaptation of the English data set to Portuguese, to create TimeBankPT, was carried out, and we presented an effort to automatically detect annotation errors, based on the logical properties of the temporal relations being annotated. Finally, we provided an assessment of the differences between the original English corpus and TimeBankPT, as well as a discussion on the size of TimeBankPT.

Classification of Temporal Relations Chapter 4 focuses on the classification of temporal relations. Given a temporal relation between two identified entities (an event and a time, or two events) mentioned in a text, the goal is to automatically determine the type of that relation (BEFORE, OVERLAP or AFTER).

The data set used to experiment and evaluate the proposed solutions is TimeBankPT, presented in the previous chapter. This data set contains temporal relations grouped in three different tasks: one is to classify temporal relations between events and times mentioned in the same sentence; the second one is about temporal relations between events and the time in which the document was created; the third task is about events occurring in different sentences.

First, we presented baselines for these tasks that consist of machine learning classifiers trained with features that are readily available in the annotations. We then described several natural language processing tools that are used to create more features in order to enrich these classifiers. These new features were explored next. Many different kinds of classifier features were tried, capturing various sorts of information that are considered relevant to the problem of temporal relation classification. These new features encode grammatical properties, the result of reasoning, knowledge of the world, and combinations of these. Some of the new features are based on the tools presented earlier, while others are taken from other sources. Finally, the evaluation of the new classifiers, with the extended set of features, was presented, and they are compared to the previously described baselines.

The results show substantial improvements for the task of classifying temporal relations between events and times mentioned in the same sentence, for which almost all of the new features were intended. Some improvement can also be seen for the other tasks with these features. The results are very competitive with the state of the art for other languages, and they are the first results of temporal relation classification for Portuguese.

Temporal relation classification is a hard problem. The low inter-annotator agreement for temporal relation classification, mentioned in the previous chapter, and the substantially high error rates visible in these automated approaches to temporal relation classification are evidence for that difficulty. On the one hand, there is still much room for improvement. On the other hand, the state of the art of automated temporal relation classification is already performing on the same level as humans, which means that it is already as useful as manual classification. Eventual improvements on this task will put computers outperforming humans.

Full Temporal Processing Chapter 5 had two main parts. In the first part, the classifiers of temporal relations developed in the previous chapter are integrated in a temporal processing system that is able to extract full temporal information from raw text in Portuguese: temporal expressions, events, and temporal relations between them. To this end, solutions are developed to identify times, dates and events mentioned in text.

In the second part, the utility of this temporal processing system is shown with an application. To this end, we focused on the integration of the temporal processing system with a deep grammar. Deep grammars process input sentences, producing representations of their grammatical properties (at several levels, like morphology and syntax) as well as representations of the meaning of those sentences. As such, the deep grammar used in this work already produced meaning representations of input sentences, but they lacked temporal information. First, the grammar is extended with an analysis of tense and aspect that allows it to produce representations featuring some information about time, namely information which can be obtained by looking exclusively at the grammatical properties of the input sentences (i.e. verb tense). There are several limitations to what these deep systems can do about temporal information. Therefore, after that, the full temporal processing technology developed in the first part of this chapter and in the previous one is used to enrich these representations with further temporal information. Even for some very simple problems, like determining the temporal relation between situations denoted by verbs and the time the sentence was uttered (for which verb tense is a strong cue), this integrated approach shows improvements. In fact, the integrated result is better than that produced by the grammar or the temporal extractor in isolation.

The result is a combination of a computational implementation directly based on linguistic theory, and resorting fully to handcrafted rules, with a data-driven component. This combination offers the best of both worlds (the symbolic approaches and the probabilistic methods): from the deep grammar, detailed meaning representations are extracted that take into account many details of the grammatical properties of language, providing highly structured outputs.

The phenomenon of backshift (which has to do with the meaning of verb tenses in indirect speech contexts), a fine-grained analysis of which was presented here, is an example of these detailed analyses that deep grammars can provide for complex linguistic phenomena, and backshift is a phenomenon that is very relevant for the processing of time.

With the data-driven temporal extraction component, one can explore extra-grammatical sources of information, such as world knowledge, that can help the problems of temporal extraction even though they are not well understood.

The resulting combination thus takes advantage of both approaches, benefiting from what each one has to offer in order to solve the problem of the temporal processing of natural language: detailed analyses of complex phenomena and integration of sources of knowledge that are considered to be useful to solve the problem but lack a formal model of their behavior.

6.2 Contributions: Assessment

The present work achieves several goals and makes a number of contributions to the research in the field of temporal processing. Here, we review the contributions put forward in Chapter 1, assessing how successful we were in achieving them.

Developing a corpus of Portuguese with temporal annotations Our work developed and made available a data set for Portuguese that supports the creation of tools as well as research in the field of temporal information extraction. This data set, TimeBankPT, is made up of 182 documents, each with rich annotations about time that follow a *de facto* standard. The events, times and dates mentioned in these texts are annotated, as well as temporal relations between them. It was obtained by translating and adapting the annotations of the English data used in the first TempEval challenge. As such, the results of systems evaluated on it are more comparable to the state of the-art, for English, which is evaluated on very similar data. This data set is available for free at http://nlx.di.fc.ul.pt/~fcosta/TimeBankPT. It is used to train and evaluate the solutions developed in this doctoral work.

Developing state-of-the-art temporal extraction technology for Portuguese We developed solutions to the several problems needed to fully and automatically extract temporal information from texts, also allowing their automatic annotation concerning mentioned events, times, dates, and temporal relations. Put together, these solutions create a full temporal extraction system capable of processing raw text, extracting from it all kinds of information relevant to temporality. Except in one case (noted below), the performance of the various components is comparable to the state of the art for English, which is by far the language for which most research has been conducted in this field.

- Recognizing temporal expressions in text

  Expressions that occur in text and refer to times and dates must be detected. These are crucial for the temporal understanding of a text, as they precisely locate in time the events mentioned in that text. The first step is thus to identify them. The solution developed resorts to machine learning and classifies each word in a text as belonging or not to one of these temporal expressions. Evaluation shows it has an F-measure of 0.86. This problem was addressed in Chapter 5.

- Normalizing the identified temporal expressions

  The exact dates and times that the identified temporal expressions refer to must be determined, and represented in a standard format. This is important since many such expressions, such as *tomorrow* or *next Monday* can only be interpreted in context. The standard representation produced at this stage does not make context necessary in order to exactly locate the referred date in the time line. The solution developed for this task uses a large set of hand-coded rules. Evaluation results put it at 0.91 accuracy. This problem was studied in Chapter 5.

- Recognizing terms that denote events in text

  By events we mean any situation that can be located in time. Mentions in text to them must be also recognized. They are mostly given by verbs, but many nouns also denote events. We relied on machine learning, and the solution developed classifies each word in a text as to whether it denotes an event or not. This is the only task where our results are considerably below the state of the art, with evaluation showing we perform with an F-measure of 0.72 (with the state of the art for English at 0.83). The main cause of this substandard performance seems to be the lack of a WordNet that can be used to help this task. WordNets organize nouns in concepts and arrange them in a hierarchy where more general concepts are linked to more specific concepts of the same type. Thus, pulling the hiearchy under the concept *event* allows one to obtain a very large list of nouns that denote events. This helps the problem considerably, because events denoted by nouns are very hard to recognize. We focused on this problem in Chapter 5.

- Identifying properties of these event terms

  Several properties of these words that refer to events must also be determined. These are properties that are useful in order to determine temporal relations involving the referred event. For instance, one of them is, if the event is denoted by a verb, its grammatical tense. There are several such properties to extract; one is extracted with machine learning, and the remaining ones are obtained with small sets of handcrafted rules or from the output of other natural language processing tools. The evaluation results are generally quite good, in some cases achieving almost perfect accuracy. Chapter 5 addressed this task.

- Classifying temporal relations between entities mentioned in text

  Once mentions to times, dates and events have been identified in a text, it is also interesting to determine temporal relations involving these events, effectively temporally linking them to these times and dates and to other events as well. The solutions we developed are based on machine learning approaches. Depending on the types of entities involved in these temporal relations, performance is at the same levels as that of human annotators. For temporal relations between events and times/dates mentioned in the same sentence, our results are particularly competitive with the state of the art, achieving 0.67 accuracy. The entire Chapter 4 was devoted to this problem.

Improving the automatic classification of temporal relations We devoted a considerable amount of attention to the problem of temporal relation classification: given a pair of temporally bound entities mentioned in a text (an event and another event, time or date), does the first one temporally precede the second one, does the first one temporally follow the second one, or do they temporally overlap? We tried different sources of information with the purpose of improve the tasks of temporal relation classification. Several sources of knowledge can help identify temporal relations between events and dates and times mentioned in a text, such as:

- Grammatical knowledge, not only about words (such as verb tense, part-of-speech) but also about syntactic relations between words;

- Lexical knowledge, for instance about prepositions (e.g. *before*) and about conjunctions (e.g. *when*) in a text, but also about other words;

- Knowledge about the world (e.g. in the presence of a verb like *predict*, events of predicting precede the predicted events);

- Reasoning: if one entity temporally precedes a second one, which in turn precedes a third one, then the first one must also precede the third one.

Accordingly, we created classifier features encoding all these types of information that we can extract from the texts. We then checked their impact on this specific problem of classifying temporal relations. These different sources of information had a positive impact on these classifiers of temporal relations, specially when combined with information about the syntactic structure of sentences.

This work was presented in Chapter 4. It achieved results that are very competitive with the state of the art. When classifying temporal relations between events and times mentioned in the same sentence, it achieves an accuracy of 0.67. This is substantially better than the best result in TempEval (0.62), which used similar data but for English. It is also better than the best result in TempEval-2 for this task (0.65), with the added virtue of this task being somewhat easier in TempEval-2 (where the harder cases, of events and times in the same sentence but not syntactically related, were not considered), and with the caveat that the data set used was not as similar. For the other kinds of temporal relations (relations between events and the document creation time, or between two events mentioned in consecutive sentences), our results are in line with the state of the art for English.

Improving the deep language processing of temporality An existing computational grammar of Portuguese was extended with an analysis of the meaning of the verb tenses. This grammar parses sentences and outputs grammatical and meaning representations of them. There are several similar grammars for other languages, using the same formalisms for grammatical description and meaning representation, and developed and deployed with the same tools. These implementations do not try to provide rich representations for the meaning of tense and aspect. Our work here is innovative, as we do precisely this.

This novel analysis covers not just the meaning of tense and aspect of verbs occurring in simple sentences, but it also handles the interactions between different tenses occurring in the same sentence in specific syntactic configurations (indirect speech). This was described in Chapter 5.

The temporal implementation in the grammar has some limitations, stemming from difficulties inherent to this kind of system. For instance, it does not recognize temporal expressions, and there is some error in the classification of the temporal relations it uncovers in the processed sentences: temporal relations between verbs and the document creation time are classified with 0.75 accuracy, which is somewhat lower than what the dedicated temporal relation classifiers just mentioned above are capable of delivering. This is mostly due to the ambiguity inherent in language, which these symbolic systems do not deal with well.

Because the grammar is based on linguistic theory, and so that it produces meaning representations that conform to what is found in the linguistics literature, it is also more limited in scope than the temporal extraction system. For instance, the represented events are almost exclusively verbs, with eventive nouns not represented as such. Semantic representations featuring events denoted by nouns are not common in the literature and, as far as we are aware, there is no published work on how eventive nouns can be identified and represented semantically. So some of the limitations are linked to limitations in areas that feed natural language processing. In other words, the implementation of tense and aspect in the grammar has the goal of making it produce semantic representations similar to the ones found in the linguistics literature on these topics. However, this is quite different from (and in many respects more limited than) the amount and kind of temporal information that current temporal extraction technology can deliver.

Improving full-fledged temporal processing The shortcomings of the temporal implementation in the grammar were addressed by combining it with the temporal extraction system developed as part of the previous goals. Among others, the temporal extractor includes the sophisticated temporal relation classifiers mentioned above. The meaning representations produced by the grammar, and now including several pieces of information about time, are expanded with more temporally relevant information coming from the temporal extractor. Among other things, this includes information about temporal expressions and normalized representations of the times and dates that they refer to and temporal relations between these times and dates and the events and states mentioned in the text. In this integrated approach, the original temporal relations identified by the grammar can also be corrected on the basis of the temporal relations identified by the temporal extractor.

This work was presented in Chapter 5. The evaluation results are positive: in some respects, the combined system performs better than either of its components (the grammar and the temporal extractor). For instance, the classification of temporal relations between a situation denoted by a verb and the document creation time has an accuracy of 0.94, while the grammar alone has precision problems and the temporal extractor alone has recall problems (for detecting events denoted by verbs) that put their accuracy substantially lower than this.

This integrated system can be seen as an application of the temporal extraction system developed in this doctoral work. The temporal extractor is used to augment and improve an existing natural language processing system by acting as a dedicated temporal module. This sort of information is useful to many other different natural language processing systems, and a temporal extraction component similar to ours can be used in a similar way.

6.3 Insights

This work brings about some insights related to temporal information processing and natural language processing in general that are worth mentioning.

Multiple Knowledge Sources Considering the problem of temporal relation classification, our hypothesis was that multiple knowledge sources are needed to solve this problem. Once again citing Derczynski & Gaizauskas (2010):

> Recent improvements (. . .) still yield marginal improvements (. . .). It seems that to break through this performance "wall", we need to continue to innovate

with and discuss temporal relation labeling, using information and knowledge from many sources to build practical high-performance systems.

Indeed, our work focused on encoding many different types of knowledge in the classifier features used to address the problem. As mentioned before, it seems clear to us that temporal relation classification (and also temporal temporal processing in general) requires access to many different types of information. In this respect it is strikingly different from many other natural language processing tasks, that can be addressed by just looking at the grammar and the lexicon.

Accordingly, our work resorted to a number of different levels of grammatical information—lexical, morphological, syntactic and semantic information—arithmetic operations (needed to compute the exact dates that many temporal expressions refer to, such as *three days before*), knowledge about our calendar system (also needed for this task), reasoning and even knowledge about the world (for instance, which kinds of events often precede in time which other kinds of events).

Furthermore, our work on extending an existing computational grammar with a temporal extractor working as its temporal module makes this need of extra-linguistic knowledge very obvious. On the one hand, the grammar is implemented in a formalism that is exclusively intended to model grammatical knowledge. On the other hand, the temporal extractor includes all the different kinds of knowledge just mentioned. Surely, the increased performance of the resulting integrated approach is, at least in part, a consequence of this contribution of extra-grammatical knowledge.

Overall, our results support this hypothesis that temporal processing absolutely needs contributions from all these different sources. Furthermore, the present study reflects how, in some of its subsystems, language interacts with other cognitive processes.

Methods Specific to the Portuguese Language Another insight that can be obtained from our work is that in many interesting problems the biggest gains are obtained from language specific methods. For instance, the most important classifier feature for the machine learning classifier developed in this thesis to determine the type of temporal relation holding between events and times mentioned in the same sentence is based on the output of a syntactic parser, enriched with hand-made rules that reflect a lot of knowledge of the Portuguese language. It would not have been possible without parsing technology for this language. In fact, many of the other approaches developed in this work require existing natural language technology.

This insight reinforces the need to extend research on natural language processing to multiple languages. Research in this field is heavily biased towards English, but it is important to address other languages as well. If the best results are indeed obtained with language specific methods, this means that developing natural language processing solutions to a new language is not as simple as retraining an existing system with new data. More basic technology may be required, and the best solutions may even require different approaches.

This result supports our choice of investing on the temporal processing of Portuguese, a language that until now had been lacking much of this temporal extraction technology. It also justifies our work on developing an annotated resource for Portuguese, which was fruitfully used in our work and can be explored by future work as well.

Hybrid Natural Language Processing Another important point is that this thesis showcases the synergy between knowledge-rich approaches and data-driven probabilistic models.

Our work combines knowledge-rich, rule-based approaches with data-driven, probabilistic methods at various different stages.

The shallower tasks (and their sub-tasks) of processing temporal expressions and event mentions in text resorts to a combination of these two types of techniques. Some of these tasks are better addressed by probabilistic approaches while others can be better solved with rule-based components.

The more interesting task of classifying temporal relations between the entities mentioned by these expressions in natural language texts is solved with machine learning approaches that make use of many features. Many of these features are based on rich knowledge-bases (for instance, we extracted from the web several statistics that encode information about the aspectual class of verbs, effectively compiling one such repository) or computed by rule-based components.

Finally, the integration of a deep computational grammar, which is exclusively developed with a symbolic approach, with the temporal extractor consisting of these modules for processing temporal expressions, events and temporal relations is another example of combining these two main ways to approach natural language processing issues.

This combination allows us to take advantage of the strengths of each of these two kinds of methods and to mitigate their weaknesses. The present contribution towards a full-fledged processing of time adds to the overall discussion and quest on how to obtain progresses in natural language processing by means of hybrid systems that combine the complementarity of the symbolic and probabilistic approaches.

6.4 Future Research Directions

The present work addresses many of the problems of temporal extraction with machine learning algorithms, e.g. the problem of classifying temporal relations. Some classification algorithms that have been used recently in this area of temporal processing have shown great promise. They include Conditional Random Fields (Lafferty *et al.*, 2001) and Markov Logic Networks (Richardson & Domingos, 2006), which we briefly mentioned in Chapter 2.

Additional knowledge sources may also improve the tasks present in temporal extraction. Paşca (2007) extracts from the Web associations between dates and important events, with the purpose of automatically creating a large repository of facts that can be consulted in the context of question answering. For instance, this system is able to extract the fact that the transistor was invented in 1947 or that the first Pink Floyd album came out in 1967. This type of knowledge can be useful for temporal relation classification: if we already know when events happened, it becomes trivial to order them. Of course, this only applies when the texts to extract temporal information from mention these important events that are in the repository. It is unlikely that such an approach would improve the results of temporal relation classification when evaluated data sets like TimeBankPT or the TempEval data, because these texts do not make frequent mentions to historical facts. However, performance on different texts might be affected in interesting ways.

Future efforts can also address the limitations of our research. Many of the solutions developed in this doctoral work resort to machine learning approaches based on the data found in TimeBankPT. This is a small corpus of text documents mostly from a specific domain (finance). The evaluation data is of similar nature. As such, the technology developed and the evaluation results reported are limited to this domain. There is a considerable amount of recently published research in the field of domain adaptation, e.g. Blitzer *et al.* (2006), not just for natural language technology, as this problem also affects other areas: for instance, face recognition systems might be trained with certain poses and lighting settings, but applied in arbitrary conditions. In the future, it would be interesting to use the techniques studied in this field with the the technology we developed here.

One task that we were not able to automate with state-of-the-art performance is the identification of events in text. We believe this is due to the lack of a mature lexical resource for Portuguese similar to the English WordNet, and available also for Spanish (a small WordNet is available for Portuguese, namely MultiWordNetPT, developed in the NLX Group). When a large WordNet becomes available, it will be interesting to check its effect on the task of event recognition.

6.5 Concluding Remarks

Temporal processing is an exciting field within natural language processing. Although computational efforts to process the reference to time present in natural language have existed for decades, recent years have seen renewed interest and ambitious goals. The emergence of natural language data featuring rich annotations about time as well as evaluation campaigns targeting them has caused the flourishing of different approaches and allowed their comparison.

In this work, we focused on Portuguese, a language still underrepresented in this field. The present work brings to this language data to be explored that is comparable to that used by the state of the art. Another contribution is the development of full-fledged temporal processing technology for this language. These technological solutions we developed were evaluated on this new data. They are very competitive with the state of the art.

Our work shows the need to integrate multiple knowledge sources and of different kinds (linguistic, logic, etc.) in order to be able to tackle this problem successfully. Our conclusions also stress the importance of conducting research on the natural language processing of multiple languages, not just the typical few, as the best solutions are language specific. Finally, hybrid approaches, combining symbolic and probabilistic methods, have become very popular in the field of natural language processing. They allow the exploitation of the main strong points of each of these methodologies, while addressing their shortcomings. This doctoral work reinforces this idea and explored how this combination can be effected in the field of temporal processing. These two different kinds of approaches were combined at different points of this work, and this combination proved fruitful in increasing the accuracy of full-fledged temporal processing.

Appendix I

Tense Values

The following is a list of all values of the tense attribute of EVENT elements. Each value is accompanied with the name of the tense in Portuguese and examples, using forms of the verb *fazer* “do”.

A few details are noteworthy:

- Compound tenses with the auxiliary *ter* are not treated as PERFECTIVE aspect, but rather as separate tenses. In this case, as well as other cases of tenses involving more than one word token, the auxiliary forms are not inside the EVENT element. For instance, for *teria feito*, only *feito* is inside EVENT tags, but its tense is annotated as CC (*condicional composto*); it is not annotated as a past participle:

  teria <EVENT tense="CC">*feito*</EVENT>

- Mood is also included in these values, because there is not a perfect parallelism between different moods. For instance, indicative mood shows more tenses than subjunctive mood. Another motivating example: present subjunctive seems to be used very frequently for events that follow the document’s creation time, present indicative not as much.

- The construction involving forms of *ir* (“go”) and an infinitive are treated like the compound tenses. Note that they are often the translation of English constructions with *will*, which are annotated as FUTURE tense in the original data. The special annotations for these periphrases with *ir* are intended to capture the future value of this construction. Example:

vão <EVENT tense="IR-PI+INF">*fazer*</EVENT>

- Passives are treated like in the original English corpus: the auxiliary verb is outside the EVENT element and only the participle is inside it, but the tense is that of the auxiliary. Whenever two examples are provided in the list of the tense values that is presented below, the second one is for the passive construction.

  é <EVENT tense="PI">*feito*</EVENT>

- Similar constructions, but involving *estar* ("be"), are treated not very differently. In order to distinguish these constructions from passives with *ser*, the constructions with *estar* are given dedicated tense values. These values are similar to the tense value of the auxiliary *estar*, but they contain the suffix +PPA.

  está <EVENT tense="PI+PPA">*feito*</EVENT>

The list of all tense values used in TimeBankPT is:

C Condicional. Ex.: *faria*, *seria feito*.

CC Condicional composto. Ex.: *teria feito*, *teria sido feito*.

FC Futuro do conjuntivo. Ex.: *fizer*, *for feito*.

FI Futuro do indicativo. Ex.: *fará*, *será feito*.

FIC Futuro composto do indicativo. Ex.: *terá feito*, *terá sido feito*.

GER Gerúndio. Ex.: *fazendo*, *sendo feito*.

GERC Gerúndio composto. Ex.: *tendo feito*, *tendo sido feito*.

IMP Imperativo. Ex.: *faz*, *sê feito*.

INF Infinitivo. Ex.: *fazer*, *ser feito*.

INFC Infinitivo composto. Ex.: *ter feito*, *ter sido feito.*

INFF Infinitivo flexionado. Ex.: *fazer*, *ser feito* (*fazeres*, etc.).

INFFC Infinitivo flexionado composto. Ex.: *ter feito*, *ter sido feito* (*teres feito*, etc.).

INF+PPA Verb *estar* in the infinitivo (INF) combined with a past participle form. Ex.: *está feito.*

IR-C+INF Verb *ir* in the condicional (C) combined with an infinitive form. Ex.: *iria fazer*, *iria ser feito.*

IR-FI+INF Verb *ir* in the futuro do indicativo (FI) combined with an infinitive form. Ex.: *irá fazer*, *irá ser feito.*

IR-INFF+INF Verb *ir* in the infinitivo flexionado (INFF) combined with an infinitive form. Ex.: *ir fazer*, *ir ser feito.*

IR-PC+INF Verb ir in the presente do conjuntivo (PC) combined with an infinitive form. Ex.: *vá fazer*, *vá ser feito.*

IR-PIC+INF Verb *ir* in the pretérito imperfeito do conjuntivo (PIC) combined with an infinitive form. Ex.: *fosse fazer*, *fosse ser feito.*

IR-PII+INF Verb *ir* in the pretérito imperfeito do indicativo (PII) combined with an infinitive form. Ex.: *ia fazer*, *ia ser feito.*

IR-PI+INF Verb *ir* in the presente do indicativo (PI) combined with an infinitive form. Ex.: *vai fazer*, *vai ser feito*

MPCC Mais-que-perfeito composto do conjuntivo. Ex.: *tivesse feito*, *tivesse sido feito.*

MPI Mais-que-perfeito simples do indicativo. Ex.: *fizera*, *fora feito.*

MPIC Mais-que-perfeito composto do indicativo. Ex.: *tinha feito*, *tinha sido feito.*

NONE Used when the event term is not a verb

PC Presente do conjuntivo. Ex.: *faça*, *seja feito*

I. TENSE VALUES

PC+PPA Verb *estar* in the presente do conjuntivo (PC) combined with a past participle form. Ex.: *esteja feito.*

PI Presente do indicativo. Ex.: *faz*, *é feito.*

PIC Pretérito imperfeito do conjuntivo. Ex.: *fizesse*, *fosse feito.*

PII Pretérito imperfeito do indicativo. Ex.: *fazia*, *era feito.*

PII+PPA Verb *estar* in the pretérito imperfeito do indicativo (PII) combined with a past participle form. Ex.: *estava feito.*

PI+PPA Verb *estar* in the presente do indicativo (PII) combined with a past participle form. Ex.: *está feito.*

PPA Past participle form. Ex.: *feito*

PPCC Pretérito perfeito composto do conjuntivo. Ex.: *tenha feito*, *tenha sido feito.*

PPI Pretérito perfeito simples do indicativo. Ex.: *fez*, *foi feito.*

PPIC Pretérito perfeito composto do indicativo. Ex.: *tem feito*, *tem sido feito.*

PPIC+PPA Verb *estar* in the pretérito perfeito composto do indicativo (PPIC) combined with a past participle form. Ex.: *tem estado feito.*

In the training data, the distribution of these values is the one in Figure I.1. The 10 most frequent values account for 94% of all instances.

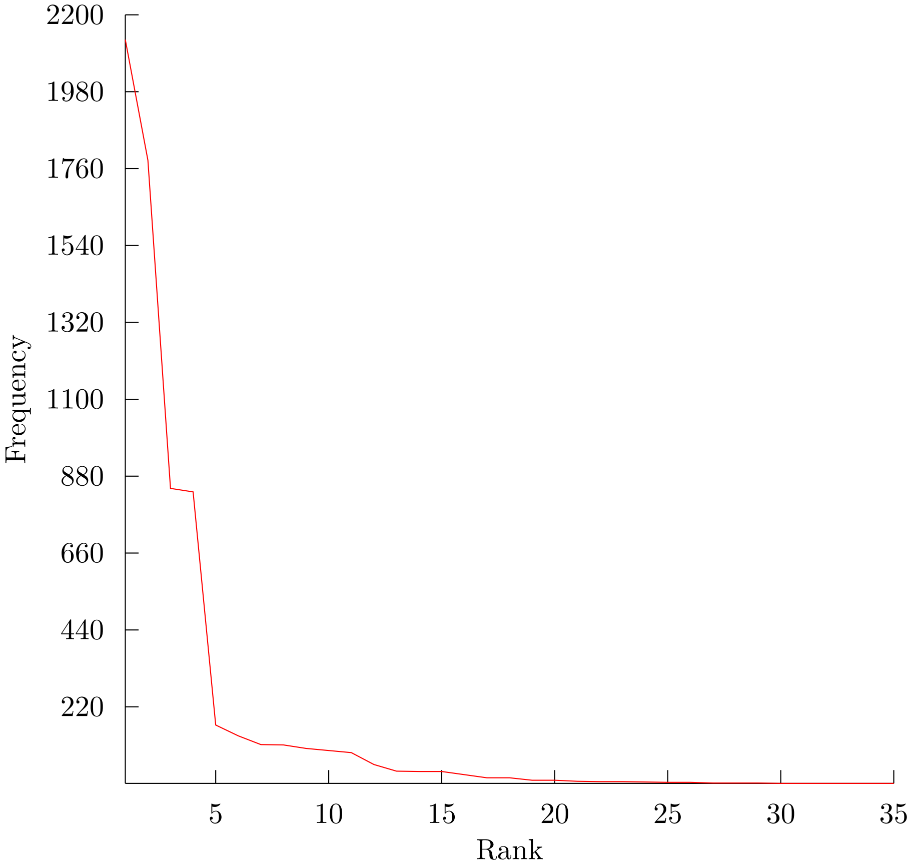


Figure I.1: Frequency of the possible values of `tense` in the training set, ordered from the most frequent to the least frequent one.

Appendix II

Simplified Tense

The classifier features that describe a simplified tense value, based on the annotated values of the TimeML attribute tense of EVENT elements, map these values in the following way:

- NONE → NONE
- PI → PRESENT
- GER → PRESENT
- PPI → PAST
- PII → PAST
- PPIC → PAST
- MPIC → PAST
- MPI → PAST
- PPA → PAST
- PPCC → PAST
- INFC → PAST
- INFFC → PAST

- GERC → PAST
- MPCC → PAST
- FI → FUTURE
- FC → FUTURE
- C → FUTURE
- IR-PI+INF → FUTURE
- IR-PII+INF → FUTURE
- IR-FI+INF → FUTURE
- IR-C+INF → FUTURE
- IR-PC+INF → FUTURE
- IR-INFF+INF → FUTURE
- PC → PRESENT_OR_FUTURE
- INF → PRESENT_OR_FUTURE
- INFF → PRESENT_OR_FUTURE
- IMP → PRESENT_OR_FUTURE
- PIC → PAST_OR_PRESENT

  Values ending in +PPA are mapped like the corresponding value without this suffix. For the remaining values of tense, NONE is returned.

Figure II.1 shows the distribution of these simplified values of tense for the annotated events in the training set of TimeBankPT.

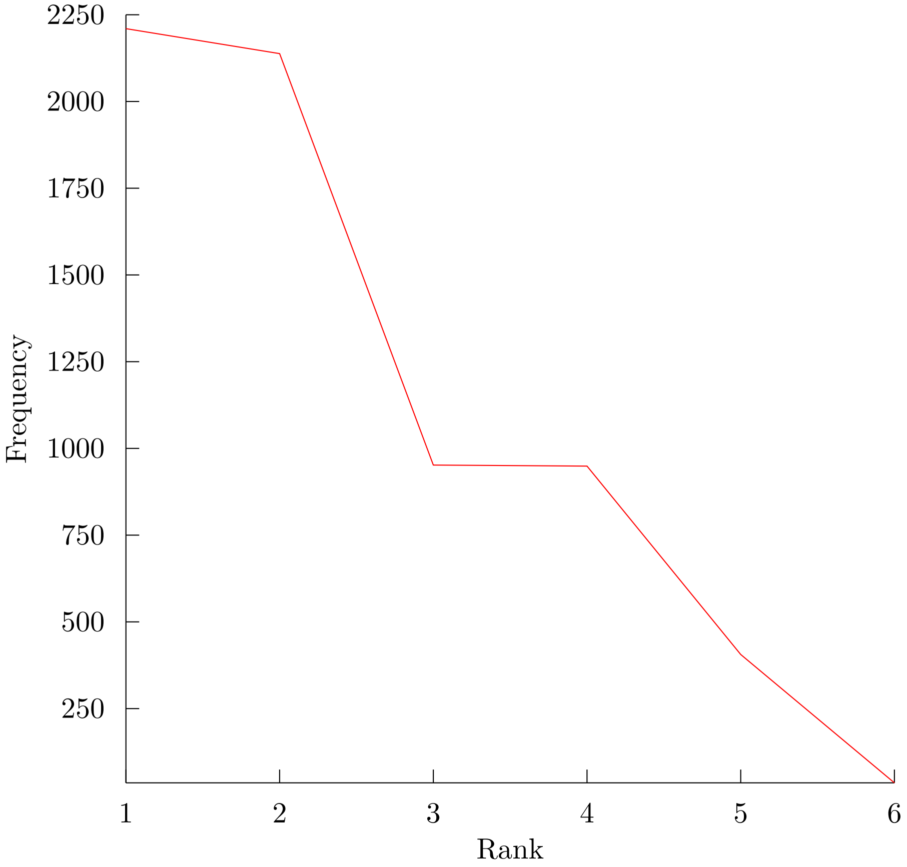


Figure II.1: Frequency of the simplified tense values in the training set, ordered from the most frequent to the least frequent one: 1: PAST, 2: NONE, 3: PRESENT, 4: PRESENT_OR_FUTURE, 5: FUTURE, 6: PAST_OR_PRESENT.

Appendix III

Temporal Direction

This appendix contains the temporal direction expected between a verb and its first complement that can denote an event, for the event terms that occur in the training set of TimeBankPT. Most event terms are not associated with a specific temporal direction. These receive the value NONE and are not listed here.

The English translation is also shown, in italics.

- abandonar → BEFORE *abandon*
- acabar → AFTER *end, finish*
- achar → AFTER *find, think*
- acreditar → AFTER *believe*
- acrescentar → AFTER *add*
- acusar → AFTER *accuse, charge*
- adotar → BEFORE *adopt*
- advertir → BEFORE *warn*
- afetar → AFTER *affect*
- afirmar → AFTER *state*
- agendar → BEFORE *schedule, set*
- ajudar → BEFORE *help, aid*
- alargar → BEFORE *extend, broaden*
- alegar → AFTER *claim*
- alertar → BEFORE *warn*
- ameaçar → BEFORE *threaten*
- antecipar → BEFORE *anticipate, foresee*
- anunciar → BEFORE *announce, report*
- anúncio → BEFORE *announcement*
- apelar → BEFORE *call*
- aplaudir → AFTER *applaud*

- aplauso → AFTER *applause*
- apresentar → AFTER *introduce, file*
- apressar → BEFORE *hurry*
- aprovação → BEFORE *approval*
- aprovar → BEFORE *approve*
- apurar → AFTER *determine*
- arriscar → BEFORE *risk*
- assegurar → BEFORE *ensure*
- assinalar → AFTER *signal*
- assinar → BEFORE *sign*
- assinatura → BEFORE *signing*
- assumir → BEFORE *assume*
- atacar → BEFORE *attack, strike*
- ataque → BEFORE *attack, strike*
- atrasar → BEFORE *stall, delay*
- autorizar → BEFORE *authorize*
- auxiliar → BEFORE *aid*
- avaliar → AFTER *evaluate*
- avançar → BEFORE *advance, push*
- avisar → BEFORE *warn, caution*
- aviso → BEFORE *warning*
- bloquear → BEFORE *block*
- boicotar → BEFORE *undermine*
- candidatar → BEFORE *run*
- capaz → BEFORE *able*
- citar → AFTER *quote*
- combater → BEFORE *fight*
- começar → BEFORE *start*
- compensar → AFTER *offset*
- completar → AFTER *complete*
- comprometer → BEFORE *pledge, undermine*
- conceber → BEFORE *design*
- concluir → AFTER *conclude*
- conclusão → AFTER *conclusion*
- concordar → BEFORE *agree*
- concurso → BEFORE *quiz*
- condenar → AFTER *condemn, convict*
- confessar → AFTER *confess*
- confirmar → AFTER *confirm*
- confrontar → AFTER *face*
- confronto → AFTER *confrontation*
- conhecer → AFTER *know*
- conquistar → BEFORE *conquer*

- conseguir → BEFORE *manage*
- consequência → AFTER *result*
- constituir → BEFORE *appoint*
- contactar → BEFORE *contact*
- contar → AFTER *tell*
- contestar → AFTER *challenge*
- contribuir → BEFORE *contribute*
- convencer → BEFORE *convince*
- conversação → AFTER *talk*
- conversar → AFTER *talk*
- convidar → BEFORE *invite*
- convocar → BEFORE *call*
- corrigir → AFTER *correct*
- criação → BEFORE *creation*
- criar → BEFORE *create*
- criticar → AFTER *criticize*
- de acordo com → AFTER *according to*
- decidir → BEFORE *decide*
- decisão → BEFORE *decision*
- declaração → BEFORE *declaration*
- declarar → BEFORE *declare*
- definir → BEFORE *define*
- deixar → BEFORE *let*
- denunciar → AFTER *report*
- desdenhar → AFTER *downplay*
- desencadear → BEFORE *trigger*
- desenhar → BEFORE *design*
- desenvolvimento → BEFORE *development*
- desintegrar → AFTER *disintegrate*
- desistir → AFTER *quit*
- desvalorização → AFTER *write-down*
- desvalorizar → AFTER *gloss*
- devolver → AFTER *return*
- dificuldade → BEFORE *difficulty*
- discurso → BEFORE *speech*
- discutir → BEFORE *discuss*
- disposto → BEFORE *willing*
- divulgar → BEFORE *release*
- dizer → AFTER *say*
- elaborar → BEFORE *elaborate*
- encerramento → AFTER *close*
- encerrar → AFTER *shut down*
- enfrentar → BEFORE *face*
- entender → AFTER *understand*

- esclarecer → AFTER *make clear*
- escolher → BEFORE *choose*
- esconder → AFTER *hide*
- esforçar → BEFORE *try hard*
- especificar → AFTER *specify*
- especular → BEFORE *speculate*
- espera → BEFORE *wait*
- esperar → BEFORE *wait, expect*
- estimar → BEFORE *estimate*
- estimativa → BEFORE *estimate*
- estimular → BEFORE *boost*
- exigir → BEFORE *demand*
- facilitar → BEFORE *ease*
- falhar → BEFORE *fail*
- fechar → AFTER *close*
- fecho → AFTER *close*
- ficar → BEFORE *stay*
- finalizar → AFTER *finalize*
- fixação → BEFORE *set*
- fixar → BEFORE *set*
- forçar → BEFORE *force*
- formar → BEFORE *form*
- fugir → BEFORE *flee*
- fundar → BEFORE *found*
- habituar → BEFORE *use*
- hipotecar → BEFORE *mortgage*
- impacto → BEFORE *impact*
- impedir → BEFORE *prevent*
- implicar → BEFORE *imply*
- impulsionar → BEFORE *boost*
- impulso → BEFORE *boost*
- indicar → BEFORE *indicate*
- informar → AFTER *inform, report*
- iniciar → BEFORE *begin*
- iniciativa → BEFORE *initiative*
- interessado → BEFORE *interested*
- inventar → BEFORE *invent*
- investigar → AFTER *investigate*
- lançar → BEFORE *release*
- levantar → BEFORE *rise*
- levar → BEFORE *take*
- listar → AFTER *list*
- manifestar → BEFORE *demonstrate*
- mencionar → AFTER *mention*

- mobilizar → BEFORE *rally, mobilize*
- mostrar → AFTER *show*
- mover → BEFORE *move*
- movimento → BEFORE *movement*
- mudança → AFTER *change, move*
- mudar → AFTER *change, move*
- negar → AFTER *deny, negate*
- negociar → BEFORE *negociate*
- nomear → BEFORE *appoint*
- notar → AFTER *notice*
- notícia → AFTER *news*
- observar → AFTER *observe, watch*
- obter → BEFORE *obtain*
- oferecer → BEFORE *offer*
- oferta → BEFORE *offer*
- olhar → AFTER *look*
- opor → BEFORE *oppose*
- ordenar → BEFORE *order*
- organizar → BEFORE *organize*
- orquestrar → BEFORE *orchestrate*
- ouvir → AFTER *listen*
- pagamento → AFTER *payment*
- pagar → AFTER *pay*
- pagável → AFTER *payable*
- parar → AFTER *stop*
- partir → BEFORE *break, leave*
- passar → BEFORE *pass*
- pedir → BEFORE *ask*
- pensar → AFTER *think*
- perda → AFTER *loss*
- perder → AFTER *lose*
- perguntar → BEFORE *ask*
- permitir → BEFORE *allow*
- planear → BEFORE *plan*
- plano → BEFORE *plan*
- posicionar → BEFORE *position*
- prejudicar → BEFORE *harm*
- preocupar → AFTER *worry*
- preparar → BEFORE *prepare*
- pressionar → BEFORE *urge*
- pretender → BEFORE *intend*
- prever → BEFORE *predict, foresee*
- procura → BEFORE *demand, search*
- procurar → BEFORE *search*
- programa → BEFORE *program*

- proibir → BEFORE *forbid*
- prolongar → AFTER *extend*
- prometer → BEFORE *promise*
- propor → BEFORE *propose*
- proposta → BEFORE *proposal*
- prorrogar → AFTER *extend*
- prosseguir → AFTER *continue*
- provar → AFTER *prove, taste*
- proveniente → AFTER *originating*
- provocar → BEFORE *cause, provoke*
- publicar → AFTER *publish*
- publicitar → BEFORE *advertise*
- queixa → AFTER *complaint*
- queixar → AFTER *complain*
- querer → BEFORE *want*
- questionar → AFTER *question*
- ratificar → AFTER *ratify*
- reacender → AFTER *rekindle*
- realização → BEFORE *holding*
- realizar → BEFORE *conduct*
- recalcular → AFTER *recalculate*
- recapitalização → AFTER *recapitalization*
- receber → AFTER *receive*
- recomprar → AFTER *buy back*
- reconhecer → AFTER *recognize, acknowledge*
- recuperar → AFTER *recover*
- recurso → BEFORE *appeal*
- recusar → BEFORE *refuse*
- reestruturação → AFTER *restructuring*
- referir → AFTER *mention, refer*
- refletir → AFTER *reflect*
- reforçar → BEFORE *strengthen*
- reformar → AFTER *reform*
- registar → AFTER *register*
- registo → AFTER *registration*
- regozijar → AFTER *cheer*
- regressar → AFTER *return*
- rejeitar → BEFORE *reject*
- relatar → AFTER *report*
- relato → AFTER *report*
- relatório → AFTER *report*
- renegociar → BEFORE *renegociate*
- repetir → AFTER *repeat*

- reportagem → AFTER *report*
- repudiar → BEFORE *disavow*
- responder → AFTER *answer*
- resposta → AFTER *answer*
- resultado → AFTER *result*
- resultar → AFTER *result*
- retirada → AFTER *withdraw*
- retirar → AFTER *withdraw*
- revelar → AFTER *reveal*
- rezar → BEFORE *pray*
- rumor → AFTER *rumor*
- saber → AFTER *know*
- saída → AFTER *exit*
- sair → AFTER *get out*
- salientar → AFTER *emphasize*
- salvar → BEFORE *save*
- saudar → AFTER *welcome, hail*
- seguir → AFTER *follow*
- segundo → AFTER *according*
- sentir → AFTER *feel*
- sinal → BEFORE *signal, sign, hint*
- sofrer → AFTER *suffer*
- subida → BEFORE *rise*
- subir → BEFORE *rise*
- sublinhar → AFTER *emphasize*
- submeter → BEFORE *file, submit*
- suceder → AFTER *succeed*
- sugerir → BEFORE *suggest*
- sujeitar → BEFORE *subject*
- surgir → AFTER *emerge*
- surpreender → BEFORE *surprise*
- suspender → AFTER *lift, suspend*
- telefonar → AFTER *call*
- telefonema → AFTER *call*
- tentar → BEFORE *try, seek, attempt*
- tentativa → BEFORE *attempt*
- terminar → AFTER *finish, expire, end, conclude, close, terminate*
- tirar → AFTER *draw, take*
- tomar → BEFORE *seize, take*
- tornar → BEFORE *become*
- transição → BEFORE *transition*
- trazer → BEFORE *bring*
- ultrapassar → AFTER *exceed*

- usar → AFTER *use*
- uso → AFTER *use*
- usufruir → AFTER *use*
- utilização → AFTER *usage*
- utilizar → AFTER *use*
- venda → BEFORE *sale, selling, sell-off*
- vender → BEFORE *sell*
- véspera → BEFORE *eve*
- visar → BEFORE *aim*
- voltar → BEFORE *come back, repeat*

Appendix IV

Rules to Order Times and Dates

Given two times or dates time_1 and time_2, then:

- value_1 is the normalized value of time_1, i.e. the string that is the value of the TimeML value attribute of the corresponding TIMEX3 element;
- value_2 is similarly the normalized value of time_2;
- start_1 is the start point of time_1 (if it can be determined);
- start_2 is the start point of time_2 (if it can be determined);
- end_1 is the end point of time_2 (if it can be determined);
- end_2 is the end point of time_2 (if it can be determined).

The rules to determine the start and end points of times and dates from their normalized value are shown at the end of this appendix.

The time time_1 is considered before time_2 if and only if at least one of the following is true:[1]

- time_1 is the document creation time (DCT) and value_2 is FUTURE_REF;
- value_1 is PRESENT_REF and value_2 is FUTURE_REF;

[1] The value PRESENT_REF is used with expressions like *now* or *currently*, PAST_REF occurs with expressions such as *recently*, and FUTURE_REF occurs with expressions like *soon* or *a later date*.

- *value*$_1$ is PAST_REF and *time*$_2$ is the DCT;
- *value*$_1$ is PAST_REF and *value*$_2$ is PRESENT_REF or FUTURE_REF;
- *end*$_1$ precedes *start*$_2$ (according to Joda-Time).

It must be noted that *start*$_1$, *start*$_2$, *end*$_1$ and *end*$_2$ are instants and Joda-Time is able to compare them for precedence.

The time *time*$_1$ is after *time*$_2$ if and only if *time*$_2$ is before *time*$_1$.

The time *time*$_1$ is considered equal to *time*$_2$ if and only if the values for the attributes value and mod are the same.

The time *time*$_1$ includes *time*$_2$ if and only if at least one of the following conditions holds:

- they are equal;
- *value*$_1$ is PRESENT_REF and *time*$_2$ is the DCT;
- *start*$_1$ is before or equals *start*$_2$ according to Joda-Time, and *end*$_1$ is after or equals *end*$_2$, also according to Joda-Time;

The time *time*$_1$ overlaps *time*$_2$ if and only if at least one of the following conditions holds:

- *time*$_1$ includes *time*$_2$;
- *time*$_2$ includes *time*$_1$;
- their endpoints are defined and they overlap according to Joda-Time (their intersection is non-empty).

Finding the Endpoints Given a time *time* with a normalized value string *value*, finding the start and end points *start* and *end* of the correspoding time interval is done in the following way. We use this representation with two points whenever possible in order to be able to compute some of the above operations with Joda-Time.

Most times are described by a *value* with the form of the regular expression dddd(-dd(-dd(Tdd(:dd(:dd(.ddd)?)?)?)?)?)?, where d is a digit. Here, T separates the date from the time, each - and : separates the other fields and the fields occur in the order *year*-*month*-*day*T*hour*:*minute*:*second*.*millisecond*. For these dates and times, *start* and *end* are set to the first and last millisecond (respectively) of the described date or time. If a field is present in *value*, it takes the same value in *start* and *end*. If it is absent, *start* will exhibit the minimum value appropriate for that field and *end* will show its maximum value. Examples:

- A date with the value 2012 has its *start* set with the year 2012, month 1, day 1, hour 0, minute 0, second 0 and millisecond 0. Its *end* is set with the year 2012, month 12, day 31, hour 23, minute 59, second 59 and millisecond 999.

- The date 2012-10-20 has its *start* set at year 2012, month 10, day 20, hour 0, minute 0, second 0 and millisecond 0. Its *end* is set with the year 2012, month 10, day 20, hour 23, minute 59, second 59 and millisecond 0.

- The time 2012-10-20T15:00 is represented with a *start* with year 2012, month 10, day 20, hour 15, minute 0 and millisecond 0. Its *end* has year 2012, month 10, day 20, hour 15, minute 0 and millisecond 999.

Instead of the time, after the date we can also find one of MO (the morning), AF (the afternoon), EV (the evening) or NI (the night). For instance 2012-10-20TAF represents the afternoon of 2012-10-20. The boundaries of these times are not well defined. For practical purposes, we need to define them so that some of the operations above can be computed. We consider them as though MO refers to the period between 06:00:00.000 and 11:59:59.999, AF to the period between 12:00:00.000 and 17:59:59.999, EV to the period between 18:00:00.000 and 23:59.59.000 and NI to the period between 00:00:00.000 and 05:59:59.999, but this choice is somewhat arbitrary and may not be correct for all cases.

Other dates have the form dddd-Wdd, where d is a digit. Here, the first field, with four digits, is the year and the second one, with two, is the number of the week in the year. For instance, 1989-W43 refers to the 43^{rd} week of 1989. Given the week of a year, it is possible to get the exact start and end dates with Joda-Time, which we use in order to obtain the start and end points.

Some dates refer to seasons of the year. For instance 1989-SU is the Summer of 1989. The strings used to encode the season are SP for the Spring, SU for the

Summer, FA for the Fall, and WI for the Winter. For these, we also need to know the start and end dates in order to represent them with these two endpoints. We assume that the Spring starts in March 21 (and includes this day), the Summer starts in June 22 (and includes it), the Fall starts in September 24 (including it) and the Winter starts in December 23 (inclusively). This is not accurate for the southern hemisphere at all, but we do not have access to geographical data.

The temporal expressions annotated in the TempEval can also refer to quarters of a year. For instance, the date 1989-Q3 is the third quarter of 1989. However, they refer to fiscal years, whose start and end dates are variable. For this reason, we can not convert these periods into this representation involving a start and an end point.

The TimeML TIMEX3 elements contain another attribute, besides value, to represent the normalized value of times and dates. This is the mod value. Some of the possible values for this attribute mod are:

- START, as in

  <TIMEX3 value="1989" mod="START">*earlier this year* </TIMEX3>

- MID, as in

  <TIMEX3 value="1997-10" mod="MID">*the middle of October*</TIMEX3>

- END as in

  <TIMEX3 value="1991-02-24" mod="END">*late yesterday*</TIMEX3>

- ON_OR_AFTER, as in

  <TIMEX3 value="1990" mod="ON_OR_AFTER">*1990 and beyond*</TIMEX3>

In these cases, we change the *start* and *end* points after obtaining them from the value attribute, based on the mod attribute. The times annotated with a mod with the value START get their *end* point changed (after asking Joda-Time the total number of milliseconds in the original interval) by subtracting to *end* a number of millisecond equal to 80% of the total number of milliseconds in the original interval. In the case of END, *start* is similarly changed so that the initial 80% milliseconds of the original interval are removed. With the value MID, 40% of the original interval is removed from either end of the interval. These numbers are arbitrary, but the result is better than ignoring this attribute mod, as that would make expressions like *the middle of October* and *October* have the same representation in terms of endpoints, which would be incorrect.

In the case of the ON_OR_AFTER value, the *end* value is increased by 1000 years. Due to the nature of the texts being processed, in practice this effectively puts the *end* of these times after the *end* of every time that is not annotated with a mod attribute with this value.

Finally, durations are ignored, as they cannot be placed in the timeline. These include time expressions such as

> <TIMEX3 tid="t519" type="DURATION" value="P4D" temporalFunction="false" functionInDocument="NONE">*the four days*</TIMEX3>

Appendix V

Optimal Feature Combinations

The following tables present the best combination of features found for the three tasks of temporal relation classification. These optimal combinations are found using greedy search starting with the full set of features and removing them one by one. The evaluation function is classifier accuracy with 10-fold cross-validation on the training data. The classification algorithm is Weka's SMO, which learns suppport vector machines. Each table presents the optimal feature set for one task.

event-indicator-at1	event-indicator-at2
event-simplified-tense	event-temporal-direction
predictor-parser	predictor-dep-parser
order-event-between	event-intervening-preceding-class
event-intervening-following-tense	event-closest-to-timex-pos
event-closest-to-timex-equal-pos	event-closest-to-event-equal-pos
event-closest-to-event-equal-class	timex3-type
timex3-relevant-lemmas	timex3-preposition
closure-B-for-A	

Figure V.1: Optimal feature set for Task A Event-Timex

event-class
event-aspect
event-indicator-st1
event-indicator-pc1
event-indicator-at1
event-simplified-tense
event-polarity
event-temporal-direction
predictor-parser
predictor-dep-parser
event-closest-to-event-temporal-direction
event-closest-to-event-equal-lemma
event-closest-to-event-pos
event-closest-to-event-equal-pos
event-closest-to-event-class
event-closest-to-event-equal-class
event-closest-to-event-equal-tense
event-closest-to-event-simplified-tense
event-closest-to-event-equal-simplified-tense
previous-temporal-relation-type
previous-instance-event-simplified-tense
previous-instance-event-temporal-direction
timexes-majority-for-B
timexes-closest-for-B

Figure V.2: Optimal feature set for Task B Event-DocTime

event-class
event-indicator-at2
event-polarity
event-simplified-tense
predictor-dep-parser
previous-instance-event-simplified-tense
event-temporal-direction
events-equal-lemma
event-indicator-at1
event-tense
event-pos
predictor-parser
previous-instance-event-tense
events-equal-subject-agreement
events-equal-temporal-direction
closure-AB-for-C

Figure V.3: Optimal feature set for Task C Event-Event

References

Abusch, D. (1994). Sequence of tense revisited: Two semantic accounts of tense in intensional contexts. In H. Kampf, ed., *Ellipsis, Tense and Questions*, 87–139, University of Amsterdam, dYANA deliverable R.2.2.B.

Ahn, D., Adafre, S.F. & de Rijke, M. (2005). Recognizing and interpreting temporal expressions in open domain texts. In *We Will Show Them: Essays in Honour of Dov Gabbay*, vol. 1, 31–50.

Ahn, D., Schockaert, S., Cock, M.D. & Kerre, E. (2006). Supporting temporal question answering: Strategies for offline data collection. In J. Bos & A. Koller, eds., *5th International Workshop on Inference in Computational Semantics*, Buxton, England.

Ahn, D., van Rantwijk, J. & de Rijke, M. (2007). A cascaded machine learning approach to interpreting temporal expressions. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, 420–427, Association for Computational Linguistics, Rochester, New York.

Allen, J. (1983). Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843.

Allen, J. (1984). Towards a general theory of action and time. *Artificiall Intelligence*, 23:123–154.

Augusto, J.C. (2005). Temporal reasoning for decision support in medicine. *Artificial Intelligence in Medicine*, 33(1):1–24.

Bach, E. (1986). The algebra of events. *Linguistics and Philosophy*, 9:5–16.

Baldwin, J. (2002). *Learning Temporal Annotation of French News*. Master's thesis, Georgetown University.

Baldwin, T., Beavers, J., Bender, E.M., Flickinger, D., Kim, A. & Oepen, S. (2005). Beauty and the beast: What running a broad-coverage precision grammar over the bnc taught us about the grammar — and the corpus. In S. Kepser & M. Reis, eds., *Linguistic Evidence: Empirical, Theoretical and Computational Perspectives*, 49–70, Mouton de Gruyter, Berlin.

Barreto, F., Branco, A., Ferreira, E., Mendes, A., Nascimento, M.F., Nunes, F. & Silva, J. (2006). Open resources and tools for the shallow processing of Portuguese: the TagShare project. In *Proceedings of LREC 2006*.

Bejan, C.A. & Harabagiu, S. (2010). Unsupervised event coreference resolution with rich linguistic features. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, 1412–1422, Association for Computational Linguistics, Uppsala, Sweden.

Bender, E.M., Flickinger, D. & Oepen, S. (2002). The Grammar Matrix: An open-source starter-kit for the development of cross-linguistically consistent broad-coverage precision grammars. In J. Carroll, N. Oostdijk & R. Sutcliffe, eds., *Procedings of the Workshop on Grammar Engineering and Evaluation at the 19th International Conference on Computational Linguistics*, 8–14, Taipei, Taiwan.

Berners-Lee, T., Hendler, J. & Lassila, O. (2001). The Semantic Web. *Scientific American Magazine*, 34–43.

Binnick, R.I. (1991). *Time and the Verb: A Guide to Tense and Aspect*. Oxford University Press, Oxford.

Bittar, A., Amsili, P., Denis, P. & Danlos, L. (2011). French TimeBank: An ISO-TimeML annotated reference corpus. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, 130–134, Association for Computational Linguistics, Portland, Oregon, USA.

Blitzer, J., McDonald, R. & Pereira, F. (2006). Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006)*, 120–128, Sydney, Australia.

Bobrow, D.G., Cheslow, B., Condoravdi, C., Karttunen, L., King, T.H., Nairn, R., de Paiva, V., Price, C. & Zaenen, A. (2007). PARC's bridge and question answering system. In T.H. King & E.M. Bender, eds., *Proceedings of the GEAF07 Workshop*, 46–66, CSLI Publications, Stanford, CA.

Boguraev, B. & Ando, R.K. (2006). Analysis of TimeBank as a resource for TimeML parsing. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006)*, 71–76, ELRA, Genoa, Italy.

Bonami, O. (2002). A syntax-semantics interface for tense and aspect in French. In F.V. Eynde, L. Hellan & D. Beermann, eds., *The Proceedings of the 8th International Conference on Head-Driven Phrase Structure Grammar*, 31–50, CSLI Publications, Stanford.

Bramsen, P., Deshpande, P., Lee, Y.K. & Barzilay, R. (2006). Inducing temporal graphs. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006)*, 189–198, Sydney, Australia.

Branco, A. & Costa, F. (2010). A deep linguistic processing grammar for Portuguese. In *Lecture Notes in Artificial Intelligence*, vol. 6001, 86–89, Springer, Berlin.

Branco, A. & Silva, J. (2006). A suite of shallow processing tools for portuguese: LX-Suite. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL'06)*, Trento, Italy.

Branco, A., Costa, F., Ferreira, E., Martins, P., Nunes, F., Silva, J. & Silveira, S. (2009). LX-Center: a center of online linguistic services. In *Proceedings of the Demo Session, ACL-IJCNLP2009*, Singapore.

Branco, A., Mendes, A., Pereira, S., Henriques, P., Pellegrini, T., Meneido, H., Trancoso, I., Quaresma, P., de Lima, V.L.S. & Bacelar, F. (2012). *The Portuguese Language in the Digital Age*. White Paper Series, Springer-Verlag, Berlin Heidelberg.

Bruce, B.C. (1972). A model for temporal references and its application in a question answering program. *Artificial Intelligence*, 3(1–3):1–25.

Callmeier, U. (2000). PET — A platform for experimentation with efficient HPSG processing techniques. *Natural Language Engineering*, 6(1):99–108, (Special Issue on Efficient Processing with HPSG).

Carletta, J. (1996). Assessing agreement on classification tasks: the kappa statistic. *Computational Linguistics*, 22(2):249–254.

Carpenter, B. (1992). *The Logic of Typed Feature Structures*. Cambridge University Press, Cambridge.

Caselli, T., Bartalesi Lenzi, V., Sprugnoli, R., Pianta, E. & Prodanof, I. (2011). Annotating events, temporal expressions and relations in italian: the It-TimeML experience for the Ita-TimeBank. In *Proceedings of the 5th Linguistic Annotation Workshop*, 143–151, Association for Computational Linguistics, Portland, Oregon, USA.

Chambers, N. (2012). Labeling documents with timestamps: Learning from their time expressions. In *Proceedings of the 50th Annual Meetings of the Association for Computational Linguistics*, Association for Computational Linguistics, Jeju, Republic of Korea.

Chambers, N. & Jurafsky, D. (2008a). Jointly combining implicit constraints improves temporal ordering. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, 698–706, Association for Computational Linguistics, Honolulu, Hawaii.

Chambers, N. & Jurafsky, D. (2008b). Unsupervised learning of narrative event chains. In *Procedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL 08)*, 789–797, Association for Computational Linguistics, Columbus, Ohio, USA.

Chambers, N., Wang, S. & Jurafsky, D. (2007). Classifying temporal relations between events. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, Prague, Czech Republic.

Charniak, E. (2000). A maximum-entropy-inspired parser. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, 132–139.

Charniak, E. & Johnson, M. (2005). Coarse-to-fine *n*-best parsing and MaxEnt discriminative reranking. In *Proceedings of the 43rd Annual Meeting of the ACL*, 173–180, Association for Computational Linguistics, Ann Arbor.

Cheng, Y., Asahara, M. & Matsumoto, Y. (2008). Constructing a temporal relation tagged corpus of Chinese based on dependency structure analysis. *Computational Linguistics and Chinese Language Processing*, 13(2):171–196.

Chklovski, T. & Pantel, P. (2004). VerbOcean: Mining the Web for fine-grained semantic verb relations. In *In Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-04)*, Barcelona, Spain.

Cleary, J.G. & Trigg, L.E. (1995). K*: An instance-based learner using an entropic distance measure. In *12th International Conference on Machine Learning*, 108–114.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46.

Cohen, W.W. (1995). Fast effective rule induction. In *Proceedings of the Twelfth International Conference on Machine Learning*, 115–123.

Comrie, B. (1976). *Aspect: An Introduction to the Study of Verbal Aspect and Related Problems*. Cambridge University Press, Cambridge.

Comrie, B. (1985). *Tense*. University Press, Cambridge.

Comrie, B. (1986). Tense in indirect speech. *Folia Linguistica*, 20:265–296.

Copestake, A. (2002). *Implementing Typed Feature Structure Grammars*. CSLI Publications, Stanford.

Copestake, A. & Flickinger, D. (2000). An open-source grammar development environment and broad-coverage English grammar using HPSG. In *Proceedings of the Second conference on Language Resources and Evaluation (LREC-2000)*, Athens, Greece.

Copestake, A., Flickinger, D., Sag, I.A. & Pollard, C. (2005). Minimal Recursion Semantics: An introduction. *Journal of Research on Language and Computation*, 3(2–3):281–332.

Costa, F. & Branco, A. (2010). Temporal information processing of a new language: Fast porting with minimal resources. In *ACL2010—Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, 671–677, Association for Computational Linguistics, Uppsala, Sweden.

Costa, F. & Branco, A. (2012a). Backshift and tense decomposition. In S. Müller, ed., *Proceedings of the 19th International Conference on Head-Driven Phrase Structure Grammar, Chungnam National University Daejeon*, 86–106.

Costa, F. & Branco, A. (2012b). Extracting temporal information from Portuguese texts. In H. Caseli, A. Villavicencio, A. Teixeira & F. Perdigão, eds., *Computational Processing of the Portuguese Language—10th International Conference, PROPOR 2012*, vol. 7243 of *Lecture Notes in Artificial Intelligence*, 99–105, Springer, Berlin.

Costa, F. & Branco, A. (2012c). LX-TimeAnalyzer: A temporal information processing system for Portuguese. Tech. rep., Universidade de Lisboa, Faculdade de Ciências, Departamento de Informática.

Costa, F. & Branco, A. (2012d). TimeBankPT: A TimeML annotated corpus of Portuguese. In N. Calzolari, K. Choukri, T. Declerck, M.U. Doğan, B. Maegaard, J. Mariani, J. Odijk & S. Piperidis, eds., *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC'12)*, 3727–3734, European Language Resources Association (ELRA), Istanbul, Turkey.

Cowie, J. & Lehnert, W. (2000). Information extraction. *Communications of the ACM*, 39(1):80–91.

Crysmann, B. (2007). Local ambiguity packing and discontinuity in German. In *Proceedings of the ACL Workshop on Deep Linguistic Processing*, Prague.

Dale, R. & Mazur, P. (2006). Local semantics in the interpretation of temporal expressions. In *Proceedings of the Workshop on Annotating and Reasoning about Time and Events*, 9–16.

Davidson, D. (1967). The logical form of action sentences. In N. Rescher, ed., *The Logic of Decision and Action*, University of Pittsburgh Press.

Davidson, D. (1985). Reply to Quine on events. In E. LePore & B. McLaughlin, eds., *Actions and Events: Perspectives on the Philosophy of Donald Davidson*, 172–176, Blackwell, Oxford.

de Swart, H. (1998a). Aspect shift and coercion. *Natural Language and Linguistic Theory*, 16:347–385.

de Swart, H. (1998b). *Introduction to Natural Language Semantics*. CSLI Publications, Stanford.

de Swart, H. (2000). Tense, aspect and coercion in a cross-linguistic perspective. In M. Butt & T.H. King, eds., *Proceedings of the Berkeley Formal Grammar conference*, CSLI Publications, Stanford.

Declerck, R. (1990). Sequence of tenses in English. *Folia Linguistica*, 24:513–544.

Denis, P. & Muller, P. (2010). Comparison of different algebras for inducing the temporal structure of texts. In *Proceedings of COLING 2010*, 250–258, Beijing, PRC.

Denis, P. & Muller, P. (2011). Predicting globally-coherent temporal structures from texts via endpoint inference and graph decomposition. In *Proceedings of the International Joint Conference on Artificial Intelligence(IJCAI) 2011*.

Derczynski, L. & Gaizauskas, R. (2010). USFD2: Annotating temporal expresions [sic] and TLINKs for TempEval-2. In K. Erk & C. Strapparava, eds., *SemEval 2010—5th International Workshop on Semantic Evaluation—Proceedings of the Workshop*, 337–340, Uppsala University, Uppsala, Sweden.

Dowty, D.R. (1979). *Word Meaning and Montague Grammar: the Semantics of Verbs and Times in Generative Semantics and Montague's PTQ*. Reidel, Dordrecht.

Enç, M. (1986). Temporal interpretation. In *Proceedings of the Fifth West Coast Conference on Formal Linguistics*, Seattle, Washington.

Etzioni, O., Cafarella, M., Downey, D., Kok, S., Popescu, A.M., Shaked, T., , Soderland, S., Weld, D.S. & Yates, A. (2004). Web-scale information extraction in KnowItAll. In *Proceedings of the 13th International Conference on World Wide Web*.

Fellbaum, C., ed. (1998). *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA.

Ferro, L., Mani, I., Sundheim, B. & Wilson, G. (2001). TIDES temporal annotation guidelines, version 1.0.2. Tech. Rep. MTR 01W0000041, The MITRE Corporation, McLean, Virginia.

Ferro, L., Gerber, L., Mani, I., Sundheim, B. & Wilson, G. (2004). TIDES 2003 standard for the annotation of temporal expressions. Tech. rep., The MITRE Corporation, McLean, Virginia.

Filatova, E. & Hovy, E. (2001). Assigning time-stamps to event-clauses. In *Proceedings of the 2001 ACL Workshop on Temporal and Spatial Information Processing*, 88–95, Association for Computational Linguistics, Toulouse, France.

Flouraki, M. (2006). Constraining aspectual composition. In S. Müller, ed., *The Proceedings of the 13th International Conference on Head-Driven Phrase Structure Grammar*, 140–157, CSLI Publications, Stanford.

Forăscu, C. & Tufiş, D. (2012). Romanian TimeBank: An annotated parallel corpus for temporal information. In N. Calzolari, K. Choukri, T. Declerck, M.U. Doğan, B. Maegaard, J. Mariani, J. Odijk & S. Piperidis, eds., *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC'12)*, European Language Resources Association (ELRA), Istanbul, Turkey.

GALTON, A. (1990). A critical examination of Allen's theory of action and time. *Artificial Intelligence*, 42.

GAREY, H.B. (1957). Verbal aspect in French. *Language*, 33(2):91–110.

GOSS-GRUBBS, D. (2005). *An Approach to Tense and Aspect in Minimal Recursion Semantics*. Master's thesis, University of Washington, Seattle, Washington.

GUSEV, A., CHAMBERS, N., KHAITAN, P., KHILNANI, D., BETHARD, S. & JURAFSKY, D. (2011). Using query patterns to learn the duration of events. In *IEEE IWCS-11, 9th International Conference on Web Services*, Oxford, UK.

GUTIÉRREZ, A.C. & FERNÁNDEZ, L.G. (1994). Sequence of tenses in Spanish. *Working Papers in Linguistics, Università Ca'Foscari Venezia*, 4:45–70.

HA, E.Y., BAIKADI, A., LICATA, C. & LESTER, J.C. (2010). NCSU: Modeling temporal relations with Markov logic and lexical ontology. In K. Erk & C. Strapparava, eds., *SemEval 2010—5th International Workshop on Semantic Evaluation—Proceedings of the Workshop*, 341–344, Uppsala University, Uppsala, Sweden.

HAGÈGE, C., BAPTISTA, J. & MAMEDE, N. (2008a). Proposta de anotação e normalização de expressões temporais da categoria tempo para o Segundo HAREM. In C. Mota & D. Santos, eds., *Desafios na avaliação conjunta do reconhecimento de entidades mencionadas: O Segundo HAREM.*, 289–308, Linguateca.

HAGÈGE, C., BAPTISTA, J. & MAMEDE, N. (2008b). Reconhecimento de entidades mencionadas com o XIP: Uma colaboração entre o INESC-L2f e a Xerox. In C. Mota & D. Santos, eds., *Desafios na avaliação conjunta do reconhecimento de entidades mencionadas: O Segundo HAREM.*, 261–274, Linguateca.

HAN, B. & LAVIE, A. (2004). A framework for resolution of time in natural language. *ACM Transactions on Asian Language Information Processing (TALIP)—Special Issue on Spatial and Temporal Information Processing*, 3(1):11–32.

HARABAGIU, S. & BEJAN, C.A. (2005). Question answering based on temporal inference. In *Proceedings of the AAAI-2005 Workshop on Inference for Textual Question Answering*, 27–34, Pittsburg, PA, USA.

HEARST, M.A. (1992). Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th Conference on Computational Linguistics*, vol. 2, 539–545, Nantes, France.

HEPPLE, M., SETZER, A. & GAIZAUSKAS, R. (2007). USFD: Preliminary exploration of features and classifiers for the TempEval-2007 tasks. In *Proceedings of SemEval-2007*, 484–487, Association for Computational Linguistics, Prague, Czech Republic.

HINRICHS, E. (1986). Temporal anaphora in discourses of English. *Linguistics and Philosophy*, 9:63–82.

HORNSTEIN, N. (1990). *As Time Goes By: Tense and Universal Grammar*. MIT Press, Cambridge, USA.

HORNSTEIN, N. (1991). *As Time Goes By*. MIT Press, Cambridge, USA.

HOUAISS, A. (1991). *A Nova Ortografia da Língua Portuguesa*. Ática, São Paulo.

IM, S., YOU, H., JANG, H., NAM, S. & SHIN, H. (2009). KTimeML: Specification of temporal and event expressions in Korean text. In *Proceedings of the 7th Workshop on Asian Language Resources*, 115–122, Association for Computational Linguistics, Stroudsburg, PA, USA.

JOHN, G.H. & LANGLEY, P. (1995). Estimating continuous distributions in Bayesian classifiers. In *Eleventh Conference on Uncertainty in Artificial Intelligence*, 338–345, San Mateo.

KAMP, H. & REYLE, U. (1993). *From Discourse to Logic: An Introduction to Modeltheoretic Semantics, Formal Logic and Discourse Representation Theory*. Kluwer, Dordrecht.

KAMP, H. & ROHRER, C. (1983). Tense in texts. In R. Bauerle, C. Schwarze & A. von Stechow, eds., *Meaning, Use and Interpretation of Language*, 250–269, de Gruyter, Berlin.

KAMP, J.A.W. (1968). *Tense Logic and the Theory of Linear Order*. Ph.D. thesis, University of California, Los Angeles.

KAPLAN, R.M. & BRESNAN, J. (1982). Lexical-Functional Grammar: A formal system for grammatical representation. In J. Bresnan, ed., *The Mental Representation of Grammatical Relations*, MIT Press Series on Cognitive Theory and Mental Representation, chap. 4, 173–281, MIT Press, Cambridge, Massachusetts.

KATZ, G. & AROSIO, F. (2001). The annotation of temporal information in natural language sentences. In *Proceedings of the 2001 ACL Workshop on Temporal and Spatial Information Processing*, Association for Computational Linguistics, Toulouse, France.

KLEIN, D. & MANNING, C. (2003). Fast exact inference with a factored model for NLP. *Advances in Natural Language Processing Systems*, 15:3–10.

KOHAVI, R. (1995). The power of decision tables. In *8th European Conference on Machine Learning*, 174–189.

KOLOMIYETS, O., BETHARD, S. & MOENS, M.F. (2011). Model-portability experiments for textual temporal analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 271–276, Association for Computational Linguistics, Portland, Oregon, USA.

KOZAREVA, Z., RILOFF, E. & HOVY, E. (2008). Semantic class learning from the web with hyponym pattern linkage graphs. In *Proceedings of ACL-08: HLT*, 1048–1056, Association for Computational Linguistics, Columbus, Ohio.

KRIEGER, H.U. & SCHÄFER, U. (1994). TDL — A type description language for constraint-based grammars. In *Proceedings of the 15th International Conference on Computational Linguistics*, 893–899, Kyoto, Japan.

KRIFKA, M. (1992). Thematic relations as links between nominal reference and temporal constitution. In I. Sag & A. Szabolcsi, eds., *Lexical Matters*, 29–53, CSLI Publications, Chicago University Press.

LAFFERTY, J., MCCALLUM, A. & PEREIRA, F.C.N. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 8th International Conference on Machine Learning 2001 (ICML 2001)*, 282–289.

Lapata, M. & Lascarides, A. (2006). Learning sentence-internal temporal relations. *Journal of AI Research*, 27:85–117.

Lascarides, A. & Asher, N. (1993). Temporal interpretation, discourse relations, and common sense entailment. *Linguistics and Philosophy*, 16:437–493.

Lee, C.M. (2010). Temporal relation identification with endpoints. In *HLT-SRWS '10 Proceedings of the NAACL HLT 2010 Student Research Workshop*, 40–45, Association for Computational Linguistics, Stroudsburg, PA, USA.

Lee, C.M. & Katz, G. (2009). Error analysis of the TempEval temporal relation identification task. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions*, 138–145, Association for Computational Linguistics, Boulder, Colorado.

Li, W., Wong, K.F. & Yuan, C. (2005). A model for processing temporal references in Chinese. In *The Language of Time*, Oxford University Press, Oxford, U.K.

Ling, X. & Weld, D.S. (2010). Temporal information extraction. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI-10)*.

Llorens, H., Saquete, E. & Navarro, B. (2010a). TIPSem (English and Spanish): Evaluating CRFs and semantic roles in TempEval-2. In K. Erk & C. Strapparava, eds., *SemEval 2010—5th International Workshop on Semantic Evaluation—Proceedings of the Workshop*, 284–291, Uppsala University, Uppsala, Sweden.

Llorens, H., Saquete, E. & Navarro-Colorado, B. (2010b). TimeML events recognition and classification: Learning CRF models with semantic roles. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, 725–733, Coling 2010 Organizing Committee, Beijing, PRC.

Lo Cascio, V. (1986). Temporal deixis and anaphor in sentence and text: Finding a reference time. In V.L. Cascio & C. Vet, eds., *Temporal structure in sentence and discourse*, 191–228, Foris, Dordrecht.

LO CASCIO, V. & ROHRER, C. (1986). Interaction between verbal tenses and temporal adverbs in complex sentences. In V.L. Cascio & C. Vet, eds., *Temporal structure in sentence and discourse*, 229–249, Foris, Dordrecht.

LO CASCIO, V. & VET, C., eds. (1986). *Temporal Structure in Sentence and Discourse*. Groningen-Amsterdam Studies in Semantics, Foris, Dordrecht.

MANI, I. (2001). *Automatic Summarization*. John Benjamins Publishing Company, Amsterdam/Philadelphia.

MANI, I., PUSTEJOVSKY, J. & GAIZAUSKAS, R., eds. (2005). *The Language of Time: A Reader*. Oxford University Press, USA.

MANI, I., VERHAGEN, M., WELLNER, B., LEE, C.M. & PUSTEJOVSKY, J. (2006). Machine learning of temporal relations. In *Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics*, Sydney, Australia.

MANI, I., WELLNER, B., VERHAGEN, M. & PUSTEJOVSKY, J. (2007). Three approaches to learning TLINKs in TimeML. Tech. Rep. CS-07-268, Brandeis University.

MANNING, C. (1992). Presents embedded under pasts. Manuscript.

MAYOL, L., BOLEDA, G. & BADIA, T. (2005). Automatic acquisition of syntactic verb classes with basic resources. *Language Resources and Evaluation*, 39(4):295–312.

MAZUR, P. & DALE, R. (2008). What's the date? High accuracy interpretation of weekday names. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, 553–560, Coling 2008 Organizing Committee, Manchester, UK.

MAZUR, P. & DALE, R. (2010). WikiWars: A new corpus for research on temporal expressions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'10)*, 913–922.

MCDONALD, R., CRAMMER, K. & PEREIRA, F. (2005). Online large-margin training of dependency parsers. In *Proceedings of the 43rd Annual Meeting of the*

Association for Computational Linguistics (ACL 2005), 91–98, Association for Computational Linguistics, Stroudsburg, PA, USA.

Melnik, N. (2005). From "hand-written" to computationally implemented hpsg theories. In S. Müller, ed., *The Proceedings of the 12th International Conference on Head-Driven Phrase Structure Grammar*, 311–321, CSLI Publications, Stanford.

Michaelis, L. (2006). Tense in English. In B. Aarts & A. McMahon, eds., *The Handbook of English Linguistics*, Blackwell, Oxford.

Michaelis, L.A. (2011). Stative by construction. *Linguistics*, 49:1359–1400.

Mirroshandel, S.A., Ghassem-Sani, G. & Nasr, A. (2011). Active learning strategies for support vector machines, application to temporal relation classification. In *Proceedings of the 5th International Joint Conference on Natural Language Processing*, 56–64, AFNLP, Chiang Mai, Thailand.

Moens, M. (1987). *Tense, Aspect and Temporal Reference*. Ph.D. thesis, Center for Cognitive Science, University of Edinburgh.

Moens, M. & Steedman, M. (1988). Temporal ontology and temporal reference. *Computational Linguistics*, 14(2):15–28.

Móia, T. (2000). *Identifying and computing temporal locating adverbials with a particular focus on Portuguese and English*. Ph.D. thesis, Universidade de Lisboa.

MUC-6 (1995). *Proceedings of the Sixth Message Understanging Conference (MUC-6)*. Defense Advanced Research Projects Agency.

MUC-7 (1998). *Proceedings of the Sixth Message Understanging Conference (MUC-6)*. Defense Advanced Research Projects Agency.

Negri, M. & Marseglia, L. (2004). Recognition and normalization of time expressions: ITC-irst at TERN 2004. Tech. rep., Trento.

Och, F.J. & Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

OEPEN, S., TOUTANOVA, K., SHIEBER, S., MANNING, C., FLICKINGER, D. & BRANTS, T. (2002). The LinGO Redwoods treebank: Motivation and preliminary applications. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002)*, 1253–7, Taipei, Taiwan.

PAN, F., MULKAR-MEHTA, R. & HOBBS, J.R. (2006). An annotated corpus of typical durations of events. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC)*, 77–82, Genoa, Italy.

PAN, F., MULKAR-MEHTA, R. & HOBBS, J.R. (2011). Annotating and learning event durations in text. *Computational Linguistics*, 37(4):727–752.

PARSONS, T. (1990). *Events in the Semantics of English: A Study in Subatomic Semantics*. No. 19 in Current Studies in Linguistics Series, MIT Press, Cambridge, MA.

PARTEE, B. (1973). Some structural analogies between tenses and pronouns in English. *The Journal of Philosophy*, 70:601–609.

PARTEE, B. (1984). Nominal and temporal anaphora. *Linguistics and Philosophy*, 7:243–286.

PAŞCA, M. (2007). Lightweight Web-based fact repositories for textual question answering. In *Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management (CIKM '07)*, 87–96, ACM, New York, NY, USA.

PASSONNEAU, R.J. (1988). A computational model of the semantics of tense and aspect. *Computational Linguistics*, 14(2):44–60.

PLATT, J. (1998). Fast training of support vector machines using sequential minimal optimization. In B. Schölkopf, C. Burges & A.J. Smola, eds., *Advances in Kernel Methods—Support Vector Learning*.

PLATZACK, C. (1979). *The Semantic Interpretation of Aspect and Aktionsarten. A Study of Internal Time Reference in Swedish*. Foris, Dordrecht.

POLLARD, C. & SAG, I. (1987). *Information-Based Syntax and Semantics, Vol. 1*. CSLI Publications, Stanford.

Pollard, C. & Sag, I. (1994). *Head-Driven Phrase Structure Grammar*. Chicago University Press and CSLI Publications, Stanford.

Portner, P. (2003). The (temporal) semantics and (modal) pragmatics of the perfect. *Linguistics and Philosophy*, 26:459–510.

Poulsen, L. (2011). Meta-modeling of tense and aspect and in a cross-linguistic grammar engineering platform. In S. Song & J. Crowgey, eds., *University of Washington Working Papers in Linguistics (UWWPL)*, vol. 28.

Prager, J.M., Brown, E. & Coden, A. (2000). Question-answering by predictive annotation. In *Proceedings of the 23rd Annual International ACM SIGIR Conference of Research and Development in Information Retrieval*, 184–191, Athens, Greece.

Prior, A.N. (1957). *Time and Modality*. Clarendon Press, Oxford.

Prior, A.N. (1967). *Past, Present and Future*. Clarendon Press, Oxford.

Prior, A.N. (1969). *Papers on Time and Tense*. Clarendon Press, Oxford.

Puşcaşu, G. (2007). WVALI: Temporal relation identification by syntactico-semantic analysis. In *Proceedings of SemEval-2007*, 484–487, Association for Computational Linguistics, Prague, Czech Republic.

Pustejovsky, J. (1991). The syntax of event structure. *Cognition*, 41:47–81.

Pustejovsky, J. & Stubbs, A. (2011). Increasing informativeness in temporal annotation. In *Proceedings of the Fifth Law Workshop (LAW V)*, 152—160, Association for Computational Linguistics, Portland, Oregon, USA.

Pustejovsky, J. & Verhagen, M. (2009). Semeval-2010 task 13: evaluating events, time expressions, and temporal relations (tempeval-2). In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions*, 112–116, Association for Computational Linguistics, Boulder, Colorado.

Pustejovsky, J., Castaño, J., Ingria, R., Saurí, R., Gaizauskas, R., Setzer, A. & Katz, G. (2003a). TimeML: Robust specification of event and temporal expressions in text. In *IWCS-5, Fifth International Workshop on Computational Semantics*.

Pustejovsky, J., Hanks, P., Saurí, R., See, A., Gaizauskas, R., Setzer, A., Radev, D., Sundheim, B., Day, D., Ferro, L. & Lazo, M. (2003b). The TIMEBANK corpus. In *Proceedings of Corpus Linguistics 2003*, 647–656.

Pustejovsky, J., Knippen, R., Littman, J. & Saurí, R. (2005). Temporal and event information in natural language text. In *Language Resources and Evaluation*, 39, 123–164.

Pustejovsky, J., Littman, J., Saurí, R. & Verhagen, M. (2006). Timebank 1.2 documentation. http://timeml.org/site/timebank/documentation-1.2.html.

Quine, W.V. (1985). Events and reification. In E. LePore & B.P. McLaughlin, eds., *Actions and Events: Perspectives on the Philosophy of Donald Davidson*, 162–171, Blackwell, Oxford.

Quinlan, J.R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA.

Ravichandran, D. & Hovy, E. (2002). Learning surface text patterns for a question answering system. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, 41–47, Association for Computational Linguistics, Philadelphia, Pennsylvania.

Reichenbach, H. (1947). *Elements of Symbolic Logic*. University of California Press, Berkeley.

Reis, R.A.M.S. (2010). *Marcação Semântica de Páginas Web Apoiada por Parsers de Dependências Gramaticais*. Master's thesis, Faculdade de Ciências da Universidade de Lisboa, Lisbon, Portugal.

Richardson, M. & Domingos, P. (2006). Markov logic networks. *Machine Learning*, 62(1):107–136.

Rigter, B. (1986). Focus matters. In *Temporal structure in sentence and discourse*, 99–132, Foris, Dordrecht.

Ritchie, G.D. (1979). Temporal clauses in English. *Theoretical Linguistics*, 6:87–115.

Rodríguez, J.P. (2004). *Interpreting the Spanish* Imperfecto*: Issues of Aspect, Modality, Tense, and Sequence of Tense*. Ph.D. thesis, The Ohio State University, Columbus, Ohio.

Rohrer, C. (1986). Indirect discourse and "consecutio temporum". In V.L. Cascio & C. Vet, eds., *Temporal structure in sentence and discourse*, 79–97, Foris, Dordrecht.

Sag, I.A., Wasow, T. & Bender, E.M. (2003). *Syntactic Theory–A Formal Introduction*. CSLI Publications, Stanford.

Saquete, E., Martínez-Barco, P., Muñoz, R. & Vicedo, J.L. (2004). Splitting complex temporal questions for question answering systems. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04)*, 566–573, Barcelona, Spain.

Saurí, R. & Pustejovsky, J. (2009). TimeML in a nutshell. Manuscript, http://www.timeml.org/tempeval2/tempeval2-trial/guidelines/introToTimeML-052809.pdf (retrieved October 19, 2012).

Saurí, R., Littman, J., Knippen, B., Gaizauskas, R., Setzer, A. & Pustejovsky, J. (2006). TimeML annotation guidelines: Version 1.2.1. Manuscript, http://www.timeml.org/site/publications/timeMLdocs/annguide_1.2.1.pdf (retrieved October 19, 2012).

Saurí, R., Goldberg, L., Verhagen, M. & Pustejovsky, J. (2009). Annotating events in English: TimeML annotation guidelines. Manuscript, http://www.timeml.org/tempeval2/tempeval2-trial/guidelines/EventGuidelines-050409.pdf (retrieved October 19, 2012).

Schilder, F. (1997). *Temporal Relations in English and German narrative discourse*. Ph.D. thesis, University of Edinburgh, Edinburgh, UK.

SCHILDER, F. & HABEL, C. (2001). From temporal expressions to temporal information: Semantic tagging of news messages. In *Proceedings of the 2001 ACL Workshop on Temporal and Spatial Information Processing*, 1–8, Association for Computational Linguistics, Toulouse, France.

SCHLENKER, P. (2004). Sequence phenomena and double access readings generalized (two remarks on tense, person and mood). In J. Lecarme & J. Guéron, eds., *The Syntax of Time*, MIT Press, Cambridge, Mass.

SETZER, A. (2001). *Temporal Information in Newswire Articles: an Annotation Scheme and Corpus Study*. Ph.D. thesis, University of Sheffield.

SETZER, A. & GAIZAUSKAS, R. (2000a). Annotating events and temporal information in newswire text. In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC 2000)*, European Language Resources Association (ELRA), Athens.

SETZER, A. & GAIZAUSKAS, R. (2000b). Building a temporally annotated corpus for information extraction. In *Proceedings of the Information Extraction Meets Corpus Linguistics Workshop at the 2nd International Conference on Language Resources and Evaluation (LREC 2000)*, European Language Resources Association (ELRA).

SETZER, A. & GAIZAUSKAS, R. (2001). A pilot study on annotating temporal relations in text. In *ACL 2001 Workshop on Temporal and Spatial Information Processing*.

SHADBOLT, N., HALL, W. & BERNERS-LEE, T. (2006). The Semantic Web revisited. *IEEE Intelligent Systems*, 21(3):96–101.

SIEGEL, E.V. & MCKEOWN, K. (2000). Learning methods to combine linguistic indicators: Improving aspectual classification and revealing linguistic insights. *Computational Linguistics*, 24(4):595–627.

SIEGEL, M. & BENDER, E.M. (2002). Efficient deep processing of Japanese. In *Proceedings of the 3rd Workshop on Asian Language Resources and International Standardization. Coling 2002 Post-Conference Workshop*, 31–38, Taipei, Taiwan.

Silva, J., Branco, A. & Gonçalves, P. (2010). Top-performing robust constituency parsing of Portuguese: Freely available in as many ways as you can get it. In N. Calzolari, K. Choukri, B. Maegaard, J. Mariani, J. Odijk, S. Piperidis, M. Rosner & D. Tapias, eds., *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, 1960—-1963, European Language Resources Association (ELRA), Valetta, Malta.

Silva, J.R. (2007). *Shallow Processing of Portuguese: From Sentence Chunking to Nominal Lemmatization*. Master's thesis, Faculdade de Ciências da Universidade de Lisboa, Lisbon, Portugal.

Smith, C.S. (1997). *The Parameter of Aspect*. Kluwer Academic Publishers, Dordrecht.

Stowell, T. (1993). Syntax and tense. Manuscript.

Strassel, S., Przybocki, M., Peterson, K., Song, Z. & Maeda, K. (2008). Linguistic resources and evaluation techniques for evaluation of cross-document automatic content extraction. In N. Calzolari, K. Choukri, B. Maegaard, J. Mariani, J. Odjik, S. Piperidis & D. Tapias, eds., *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, European Language Resources Association (ELRA), Marrakech, Morocco.

Tao, C., Solbrig, H.R., Sharma, D.K., Wei, W.Q., Savova, G.K. & Chute, C.G. (2010). Time-oriented question answering from clinical narratives using semantic-web techniques. In *Proceedings of the 9th International Conference on the Semantic Web*, vol. 2, 241–256, Berlin.

Tatu, M. & Srikanth, M. (2008). Experiments with reasoning for temporal relations between events. In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING '08)*, vol. 1.

Toutanova, K., Manning, C.D., Flickinger, D. & Oepen, S. (2005). Stochastic HPSG parse selection using the Redwoods corpus. *Journal of Research on Language and Computation*, 3(1):83–105.

TSANG, E.P.K. (1987). The consistent labeling problem in temporal reasoning. In *Proceedings of the Sixth National Conference on Artificial Intelligence*, 251–255, Americal Association for Artificial Intelligence, Menlo Park, CA, USA.

UZZAMAN, N. & ALLEN, J. (2011). Temporal evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 351–356, Association for Computational Linguistics, Portland, Oregon, USA.

UZZAMAN, N. & ALLEN, J.F. (2010). TRIPS and TRIOS System for TempEval-2: Extracting temporal information from text. In K. Erk & C. Strapparava, eds., *SemEval 2010—5th International Workshop on Semantic Evaluation—Proceedings of the Workshop*, 276–283, Uppsala University, Uppsala, Sweden.

VAN EYNDE, F. (1994). Auxiliaries and verbal affixes: A monostratal cross-linguistic analysis. Habilitation thesis, Katholieke Universiteit Leuven. Leuven, Belgium.

VAN EYNDE, F. (1998). Tense, aspect and negation. In F. Van Eynde & P. Schmidt, eds., *Linguistic Specifications for Typed Feature Structure Formalisms. Studies in machine Translation and Natural language Processing*, vol. 10, 209–280, Luxembourg.

VAN EYNDE, F. (2000a). A constraint-based semantics for tenses and temporal auxiliaries. In R. Cann, C. Grover & P. Miller, eds., *Grammatical interfaces in HPSG*, 231–249, CSLI Publications, Stanford University.

VAN EYNDE, F. (2000b). A constraint-based semantics for tenses and temporal auxiliaries. In R. Cann, C. Grover & P. Miller, eds., *Grammatical interfaces in HPSG*, 231–249, CSLI Publications.

VELLDAL, E. (2007). *Empirical Realization Ranking*. Ph.D. thesis, University of Oslo, Oslo.

VENDLER, Z. (1957). Verbs and times. *The Philosophical Review*, 66:143–160.

VENDLER, Z. (1967). Verbs and times. In *Linguistics in Philosophy*, 97–121, Cornell University Press, Ithaca, New York.

Verhagen, M. (2005). Temporal closure in an annotation environment. In *Language Resources and Evaluation*, 39, 211–241.

Verhagen, M. & Pustejovsky, J. (2008). Temporal processing with the TARSQI Toolkit. In *COLING 2008: Companion Volume: Demonstrations*, 189–192.

Verhagen, M., Mani, I., Sauri, R., Knippen, R., Jaeng, S.B., Littman, J., Rumshisky, A., Phillips, J. & Pustejovsky, J. (2005). Automating temporal annotation with TARSQI. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Ann Arbor, USA, demo Session.

Verhagen, M., Gaizauskas, R., Schilder, F., Hepple, M. & Pustejovsky, J. (2007). SemEval-2007 Task 15: TempEval temporal relation identification. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, 75–80, Association for Computational Linguistics, Prague, Czech Republic.

Verhagen, M., Gaizauskas, R., Schilder, F., Hepple, M., Moszkowicz, J. & Pustejovsky, J. (2009). The TempEval challenge: identifying temporal relations in text. *Language Resources and Evaluation*, 43(2):161–179, Special Issue: Computational Semantic Analysis of Language: SemEval-2007 and Beyond.

Verhagen, M., Saurí, R., Caselli, T. & Pustejovsky, J. (2010). SemEval-2010 task 13: TempEval-2. In K. Erk & C. Strapparava, eds., *SemEval 2010—5th International Workshop on Semantic Evaluation—Proceedings of the Workshop*, 51–62, Uppsala University, Uppsala, Sweden.

Verkuyl, H.J. (1972). *On the Compositional Nature of the Aspects*. Reidel, Dordrecht.

Verkuyl, H.J. (1993). *A Theory of Aspectuality: The interaction between temporal and atemporal structure*. Cambridge University Press, Cambridge.

Vilain, M., Kautz, H. & van Beek, P. (1990). Constraint propagation algorithms for temporal reasoning: A revised report. In J. de Kleer & D. Weld, eds., *Readings in Qualitative Reasoning about Physical Systems*, 373–381, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Webber, B.L. (1988). Tense as discourse anaphor. *Computational Linguistics*, 14(2):61–73.

Witten, I.H. & Frank, E. (1999). *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, San Francisco.

Witten, I.H. & Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, San Francisco, second edition.

Xue, N. & Zhou, Y. (2010). Applying syntactic, semantic and discourse constraints to Chinese temporal annotation. In *Proceedings of COLING 2010*, 1363–1372, Beijing, PRC.

Yoshikawa, K., Riedel, S., Asahara, M. & Matsumoto, Y. (2009). Jointly identifying temporal relations with Markov logic. In *Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP*.

Yoshimoto, K. & Mori, Y. (2002). A compositional semantics for complex tenses in Japanese. In F. Van Eynde, L. Hellan & D. Beermann, eds., *The Proceedings of the 8th International Conference on Head-Driven Phrase Structure Grammar*, 300–319, CSLI Publications, Stanford.

Zhao, X., Jin, P. & Yue, L. (2010). Automatic temporal expression normalization with reference time dynamic-choosing. In *COLING '10 Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, 1498–1506, Association for Computational Linguistics, Stroudsburg, PA, USA.

Zhou, L. & Hripcsak, G. (2007). Temporal reasoning with medical data—a review with emphasis on medical natural language processing. *Journal of Biomedical Informatics*, 40(2):183–202.

Zhou, Y. & Xue, N. (2011). Discourse-constrained temporal annotation. In *Proceedings of the 5th Linguistic Annotation Workshop (LAW V '11)*, 161–169, Association for Computational Linguistics, Stroudsburg, PA, USA.