An Exercise in Reuse of Resources: Adapting General Discourse Coreference Resolution for Detecting Lexical Chains in Patent Documentation

Nadjet Bouayad-Agha[1], Alicia Burga[1], Gerard Casamayor[1], Joan Codina[1], Rogelio Nazar[1], Leo Wanner[1,2]

[1]NLP Group, Department of Information and Communication Technologies, Universitat Pompeu Fabra,
[2]Catalan Institute for Research and Advanced Studies (ICREA)
[1]C/ Roc Boronat, 138, 08018 Barcelona, Spain
{firstname.lastname}@upf.edu

Abstract

The Stanford Coreference Resolution System (StCR) is a multi-pass, rule-based system that scored best in the CoNLL 2011 shared task on general discourse coreference resolution. We describe how the StCR has been adapted to the specific domain of patents and give some cues on how it can be adapted to other domains. We present a linguistic analysis of the patent domain and how we were able to adapt the rules to the domain and to expand coreferences with some lexical chains. A comparative evaluation shows an improvement of the coreference resolution system, denoting that (i) StCR is a valuable tool across different text genres; (ii) specialized discourse NLP may significantly benefit from general discourse NLP research.

Keywords: Coreference resolution, lexical chain, Stanford Coreference Resolution System, patents, domain adaptation

1. Introduction

The development of high quality large scale resources for NLP (e.g., treebanks, lexical databases/dictionaries, or tools) is a very time consuming and laborious work. Therefore, it is not surprising that an increasing number of such resources is available off-the-shelf for free use by the community. One of these resources is the *Stanford Deterministic Coreference Resolution System*[1] (henceforth, StCR). It is the last stage of a pipeline of linguistic processing modules in the Stanford CoreNLP platform, and uses the results of the preceding modules (e.g., tokenization, (named) entity recognition and syntactic dependency analysis).

In our work, we adapt StCR to the patent domain and expand its functionality to detect chains instantiated by lexical relations that go beyond strict identity. We have chosen StCR for two reasons. First, because of its quality: it scored best in the CoNLL 2011 Shared Task on general discourse coreference resolution; see (Raghunathan et al., 2010; Lee et al., 2011; Lee et al., 2013). Second, because it is rule-based rather than statistical, i.e., machine learning-based. The effort for the annotation of patents for training a machine learning-based lexical chain-recognition model is very high due to the very complex linguistic structures of the patent material, considerably higher than the creation of a set of rules by a linguist experienced in patent analysis.

Due to their linguistic idiosyncrasies, patents represent a real endurance test for any general discourse technique. The success of the adaptation of StCR to patent processing shows that (i) StCR is a valuable tool across different text genres, and, more generally, that (ii) specialized discourse NLP may significantly benefit from general discourse NLP research.

As in general discourse (Halliday and Hasan, 1976; Morris and Hirst, 1991), a lexical chain in a patent is defined as a sequence of lexical entities between which relations of identity, near-identity or non-identity hold (Recasens et al., 2010). However, the complex syntactic structures of patents and the potential distribution of the mentions of an entity across the entire document make the detection of these sequences significantly more complex. In this paper, we show how this can be done.

The remainder of the paper is structured as follows. In Section 2., we summarize an empirical study we carried out on linguistic phenomena in patents related to coreference and other lexical chains. Section 3 presents the original StCR and analyzes its application to patent documentation. Section 4 describes the adaptation of StCR to the patent domain. Section 5 sketches some general guidelines for the adaptation of StCR to a specific domain and lists the phenomena that should be taken into account during the adaptation. Section 6, finally, provides some conclusions and outlines the directions of our future work in the area of lexical chain identification in the patent domain.

2. An Empirical Study of Patents

Patent texts are notoriously difficult to read and comprehend due to their abstract vocabulary and very complex linguistic constructions. In this section, we present the results of an empirical study on the linguistic phenomena in patent texts related to coreference and other frequent types of lexical chains.

2.1. Coreferences

Let us briefly summarize the most important idiosyncrasies of patent with respect to coreferences.

Sentence length. One of the most prominent traits of patent texts is that the average length of their sentences is much longer than in general discourse. This holds particularly for the *claims* section of the patent document, where, in accordance with international patent writing regulations,

[1]http://nlp.stanford.edu/software/dcoref.shtml

each claim must be rendered as a single sentence. As a consequence, sentences often exceed 250 words; some of them reach 500 or even 900 words. These long sentences often contain multiple references to the same element of the invention or method in question that in other domains are typically distributed among sevearl separate sentences. This needs to be taken into account for coreference resolution since sentence boundaries and ordering of candidates are determining factors in many coreference resolution strategies.

NP repetition. Due to the legal nature of patents, writers of patents tend to avoid ambiguity as much as possible. This often leads to the repetition of full nominal phrases (NPs) across the text instead of pronouns once the referent has been introduced. Modifiers of the introduced referents are also kept. The frequent use of multiple modifiers in NPs, as well as the observed nominal nature of the claims, result in complex NPs with multiple levels of embedded NPs; cf. for illustration the citation (a) below. Thus, it is crucial to use the right level of NP embeddedness as the base for coreference detection. Even NPs that do not contain prepositional phrases as modifiers can be very complex and contain nouns as pre-nominal modifiers (as in (b) below), making it difficult to accurately detect the head.
NPs in the patent domain can also contain punctuation symbols other than commas (as illustrated in (c)), for which linguistic processing tools should be adapted.

(a) [[a control means]$_{\mathrm{NP}}$ for rotating [the first motor]$_{\mathrm{NP}}$ in [a given rotating direction]$_{\mathrm{NP}}$ with [a given ratio of [rotating speed]$_{\mathrm{NP}}$]$_{\mathrm{NP}}$]$_{\mathrm{NP}}$

(b) a recording [media]$_{\mathrm{N}}$ [storage]$_{\mathrm{N}}$ and player [unit]$_{\mathrm{N}}$

(c) [A device comprising: B; C; D; E; F]$_{\mathrm{NP}}$.

Due to NP repetition, pronouns occur less often than in general discourse. In particular, personal pronouns different from *it* are inexistent, and the use of referential *it*s is very limited. On the other hand, adverbial pronouns such as *wherein* are by far more frequent in patents than in general discourse and must be taken into account as candidates for coreference resolution:

> [The multi-recording media storage and player unit of claim 21]$_{\mathrm{i}}$, [wherein]$_{\mathrm{i}}$ said expanded information comprises links to web addresses ...

NP definiteness. In general discourse, the definiteness of an NP marked by a determiner constitutes a clear hint during the detection of the appropriate antecedent: an NP introduced by a definite determiner can corefer with a previous NP introduced by an indefinite NP that shares the same head. Opposite to that, two indefinite NPs cannot corefer, even if they are identical or share the same head. In the patent domain, however, our study yielded numerous examples of coreferring indefinite NPs:

(cl.1) [**A** battery remaining capacity indicating apparatus for detecting a remaining battery capacity of a charge and discharge battery]$_{\mathrm{i}}$ [...]

(cl.2) [**A** battery remaining capacity indicating apparatus]$_{\mathrm{i}}$ further comprising: [...]

(cl.3) [**A** battery remaining capacity indicating apparatus]$_{\mathrm{i}}$ further comprising: [...]

Sense idiosyncrasy. Another important trait of patent texts is that some words do not share the same meaning and grammatical function (and consequently the same PoS) with their instances in general discourse. Notably, the word *means*, which in general discourse can be either a verb or a noun, depending on the context, always behaves as a noun in patents, and therefore should be considered as a potential candidate for coreference resolution. Another prominent example is the word *said*, which in general discourse behaves as a verb yet in patents stands for a definite determiner.
Furthermore, some words that act as nominal heads in patent texts have a very abstract meaning and do not actually corefer with other expressions in the text, as is the case of the nouns *claim*, *apparatus*, or *unit*. In contrast, bare NPs (nominal groups not introduced by an article) and some numerical NPs may corefer, and therefore cannot be excluded as candidates by a coreference resolution strategy. Thus, in the example below, the bare noun *batteries* is generic and, in principle, should not corefer. However, given that there is a definite NP (*the batteries*) with the same head later in the text, and both NPs are embedded within the same larger NP, *batteries* and *the batteries* need to corefer:

> [A battery charging device (...) for continuing supply of air to [**batteries**]$_{\mathrm{bare\_NP}}$, comprising: a charge current controlling portion for judging whether an abnormal condition is stored in the memory means incorporated in [**the batteries**]$_{\mathrm{NP}}$]$_{\mathrm{NP}}$

Coreferences with multiple antecedents. In patents, it is very common to find coreferences with more than one antecedent. The last element of this type of coreference can be a relational pronoun (as illustrated by (a) below), or an NP (as illustrated by (b)):

(a) if [**said battery-side end voltage**] and [**said device-side end voltage**] do not correspond to [**each other**]

(b) The electric circuit wherein each of the DC-to-AC converters comprises [**a first switch (...)**]$_{\mathrm{i-1}}$ and [**a second switch (...)**]$_{\mathrm{i-2}}$. The electric circuit wherein [**the first and second switches**]$_{\mathrm{i}}$ are field effect transistors.

In the case of relational pronouns, the heads of the two antecedents (being pronominal or not) can be the same or different, but both antecedents are within the same clause as the pronoun, which facilitates the coreference resolution. In the case of pure NPs, though, it is very difficult to find patterns that detect the nominal elements involved in this kind of coreference. However, we observed that all the elements involved in the coreference chain share the same head (although the last one has to be in plural), and that the last mention necessarily appears after the two antecedents.

Position of the coreference elements. The position of each coreference element within the document structure is also very relevant in the patent domain, and should be taken into account. The claims section, for instance, contains a set of claims that are related to each other through dependency relations. A claim is dependent to another claim if it makes explicit reference to the latter. The dependencies between claims form a directed graph, which can be explored to limit the scope of coreference candidates for a given expression, so that only candidates in claims that are superordinated to the claim in which the expression in question occurs are considered.

2.2. Lexical Chains

For various patent processing applications (among them, e.g., entity extraction or summarization), it is essential to capture not only coreferences, but also other types of lexical chains. In what follows, we summarize the set of lexical relations that are, according to our study, the most frequent relatins in patents: 'part-whole', 'entity in process', 'set-member' and 'class'.

Part-whole: This lexical relation is very important in patents because it relates the patented object (or method) with its different components (or steps). Thus, it is present in every single patent. The composition of a device/method has to appear in the claims, but can be repeated in the description.
This relation is a $1 : n$- relation since in patents, first the patented object/method (or one of its components) is introduced in terms of a single NP and then its components (steps or (sub)components) as n further NPs. The components/steps are most of the times coordinated among themselves (by a semicolon or the conjunction *and*); cf. an example:

> [**each**]$_{\text{head}}$ having [**a DC-side arranged to connect across positive and negative terminals of cells received by the circuit**]$_{\text{comp\_1}}$ and [**an AC-side for carrying an AC voltage converted from the DC-side**]$_{\text{comp\_2}}$

Entity in process: This lexical chain marks the relation between references to a single entity, but in different phases of a process. In order to detect this lexical chain, it is necessary to go beyond the NP itself, and also look at the verbal information around (which is somehow included in the last mention). Although this relation is, in principle, a $1 : 1$ relation, due to its potential recursion it may become $1 : 1 : 1$ $\ldots$:

> [$\ldots$] a temperature detection device for detecting [**a current temperature of the battery**]$_{\text{entity\_process1}}$; a temperature rise output device for obtaining the temperature rise from [**the temperature detected by said temperature detection device**]$_{\text{entity\_process2}}$ [$\ldots$]

Set-member: This semantic relation links singular NPs (members) to plural NPs (set). This relation is a $1 : n$ relation. The single NP is the one that represents the entire collection and n the different NPs that represent the members. It is possible that more than one member is explicitly mentioned (in this case, adjectives such as *first*, *second*, etc. are commonly used). However, often there are just two NPs connected, given that the singular NP represents any member of the collection. Consider an illustration of the 'set-member' relation:

> a plurality of DC-to-AC converters each having a DC-side arranged to connect across positive and negative terminals of cells received by the circuit and [**an AC-side for carrying an AC voltage converted from the DC-side**]$_{\text{member}}$ and an inductive coupling between [**the AC-sides**]$_{\text{collection}}$ of the DC-to-AC converters to provide $\ldots$

Class: The 'class' lexical relation links an NP that refers to a specific object with another NP that refers to the class of this object (as a generic unit). Therefore, it is a $1 : 1$ relation. Both NPs can appear either in singular or in plural; the NP that refers to the class appears with no article (if in plural) or with an indefinite article that is NOT followed by a definite NP (if in singular).

> [**An electric circuit for receiving a battery of cells in series**]$_{\text{COREF/object}}$, comprising : a plurality of DC-to-AC converters (24A-24H) [$\ldots$]
> [**The electric circuit**]$_{\text{COREF}}$ wherein the inductive coupling comprises transformer windings (26A-26H) .
> An electrically powered device including [**an electric circuit**]$_{\text{class}}$ $\ldots$

3. Stanford Coreference Resolution System

Let us first give an overview of the StCR and then briefly analyze the applicability of its original configuration to patents.

3.1. Overview of StCR

The central idea behind the StCR's deterministic approach to coreference resolution is the application of successive independent coreference models (sieves) of decreasing precision, so that coreference matches for which the system has greater confidence are detected first, and further matches are detected on the basis of the former. Instead of basing the decision on whether two mentions corefer on the whole set of features extracted from the text, the system chooses to separate lower precision features from higher precision features into different sieves.
The general architecture of StCR consists of three stages:

(1) *Candidate detection*: Detection of linguistic expressions that are candidates for coreference matching, using a high-recall algorithm. The initial list is filtered out to exclude undesirable expressions such as impersonal pronouns, partitives, numericals, bare NPs, etc.

(2) *Coreference resolution*: Application of sieves from highest to lowest precision to all candidate mentions selected in (1), to obtain clusters of related entities.

(3) *Post-processing*: Elimination of singleton clusters.

A sieve in (2) starts with the first mention of the first sentence and, moving forward one mention at a time, assigns the current mention m_i an antecedent m_{i-1} from a list of candidates. These are ordered (i) from left-to-right, traversing breadth-first the syntactic dependency tree when m_{i-1} is in the same sentence as m_i (thus favoring subjects), (ii) from right-to-left, breadth-first when the candidate is an NP in a sentence preceding that of m_i (thus favoring syntactic salience and proximity), and (iii) from left-to-right if the candidate is in pronominal form in a sentence preceding that of m_i.
Each sieve traverses the candidate list until a corefering antecedent is detected, or the end of the list is reached. In case of a match, the mention m_i, its antecedent and all their coreferring mentions are grouped into the same cluster which shares the common features of all mentions. Only first mentions in textual order of the clusters are considered when searching for new matches, following the intuition that earlier mentions are more informative and have less potential candidates (and are therefore less likely to be mismatched).
The original StCR system applies 12 sieves for coreference resolution; cf. (Raghunathan et al., 2010) and (Lee et al., 2011) for detailed presentations:

> 1. Discourse Processing, 2. Exact String Match, 3. Relaxed String Match, 4. Precise Constructs, 5.–7. Strict Head Match, 8. Proper HeadWord Match, 9. Alias, 10. Relaxed Head Match, 11. Lexical Chain, 12. Pronouns.

The features that these sieves use include the mention string, shallow linguistic traits, deep linguistic analysis and semantic features. Of particular interest to our work is the Lexical Chain sieve, which in the original StCR uses WordNet to detect hypernymy or synonymy relations and which we replace with an implementation tailored to detect some of the types of lexical chains in patents discussed in Section 2.2.

3.2. Analysis of the application of the original StCR

Due to patent idiosyncrasies, the original constellation of the StCR has indeed a limited performance on patents. In particular:
• NPs are wrongly included or excluded as mentions. For instance, many impersonal pronouns are included because the filter for excluding impersonal pronouns fails:

> **It** is then judged whether the current value is not more than a specified value.

• Bare NPs are always excluded, even if they are needed as antecedents for a posterior NP:

> A battery charging device [...] for continuing supply of air to [**batteries**] [...] comprising: a charge current controlling portion for judging whether an abnormal condition is stored in the memory means incorporated in [**the batteries**]

• Smaller mentions within larger mentions with the same head (a typical phenomenon in patents where multiple levels of NP-embedding is common) are excluded:

> **[[a charge controlling portion]**$_{\text{mention\_excluded}}$ for charging the batteries at the current value that has been retrieved by the current value retrieving portion**]]**$_{\text{mention\_included}}$

• The ordering of candidate antecedents of nominal mentions within the same sentence (left-to-right) is not adequate in the patent domain because claim sentences are often too long. In general, a right-to-left order is more adequate, not only within the same sentence, but also across the text, as there is a lot of repetition of terms that may actually refer to different objects:

> Fig. 1 is a perspective view of the battery charging device according to one form of embodiment of the present invention.
> Fig. 2 is a perspective view of a battery package according to the one form of embodiment of the present invention [...] Fig. 15 is an explanatory view showing a theory for charging of the battery charging device of the third embodiment$_{\text{i}}$. [...]
> In the illustrated embodiment$_{\text{i}}$ [...]

• The hypernymy and synonymy lexical relations detected by the original StCR system are only of limited use for the highly specialized terminology of patent texts. Furthermore, truly relevant lexical chains such as those described in Section 2.2. are not detected:

> [**A battery charging device (...)**]$_{\text{Head}}$, comprising : [**a judging portion for (...)**]$_{\text{comp\_1}}$, and [**an abnormality indicating portion for (...)**]$_{\text{comp\_2}}$

4. Adapting StCR to Patents

To adapt StCR to patent material, we (1) substituted the Stanford CoreNLP pipeline by our own pipeline, and (2) tuned all three stages of the StCR.

4.1. Substitution of the Stanford CoreNLP Pipeline

Stanford's original general discourse CoreNLP pipeline performed poorly on patent material. The pipeline includes the Stanford Parser, on which the coreference detection sieves rely heavily to extract features from both its phrase structures and, to a lesser degree, from its converted dependency trees. Unfortunately, the Stanford Parser was unable to cope with the long sentences found in patent documents, and, therefore, a parser retraining did not make sense. For this reason we decided to replace it with Bohnet (2010)'s dependency parser, which was better suited to handle very long sentences (Burga et al., 2013).[2]
Our pipeline is composed of a modified version of GATE's Annie tokenizer (Cunningham, 2011), a PoS tagger, a

[2]To the best of our knowledge, no off-the-shelf NLP techniques are available for the patent domain.

lemmatizer and a parser, the last three components taken from Bohnets parsing environment. In order to replace the phrase-structure input to the coreference engine, we developed a hierarchical chunker specifically designed for the patent domain and the peculiar syntactic structures of patent sentences. The chunker output contains multiple level of embedded annotations from which we derive the constituency structures used then by StCR. We decided to include only NPs, a decision which is justified by the fact that StCR operates mostly on NPs when determining the coreference chain elements (Lee et al., 2013). The resulting phrase-based structure is thus a flat sequence of hierarchally embedded NPs.

Finally, a GATE (Cunningham, 2011) plug-in was created as a wrapper around the StCR code. The plug-in integrates the StCR into our patent processing pipeline by converting our annotations into a format accepted by StCR and the lexical chains delivered by StCR into GATE annotations.

4.2. The adaptation procedure

The following adaptations have been performed in the individual stages of the StCR.

In the candidate detection stage, we introduced patent-specific filters to exclude, e.g., simple NPs with common abstract head nouns (such as *claim*, *part*, *method*), inadequate mentions that contain ":" or ";", or mentions with over 30 tokens (to palliate chunking errors). Unlike in the original StCR, we kept smaller mentions within larger mentions with the same head and bare NPs.

In the coreference resolution stage, nominal and pronominal antecedent ordering was adapted to fit patent idiosyncrasies. From the twelve sieves that compose the StCR, except the sieve for detecting the antecedent of deictic *I/you*, all sieves were included in the same order, albeit with some modifications. In addition, a new sieve was created that detects elements related via copula. Furthermore, the following specific adaptations have been implemented:

• Sieve 1 (Discourse Processing Sieve): This sieve was removed, as patents do not include the conversational text that is the target of this sieve.

• Sieve 2 (Exact String Match): This sieve showed a high precision and was left as it is in the original configuration.

• Sieve 3 (*Relaxed String Match*: Bare NPs are linked when they occur at the beginning of sentences.

• Sieve 4 (*Precise Constructs*): The detection of appositives has been disabled; in addition, relative pronouns are assumed to refer back to the nearest antecedent mention, whilst relational pronouns such as *one another* or *each other* are assumed to refer back to the nearest plural antecedent mention.

• Sieve 5 (*Strict Head Match*): This sieve is kept as it was, but the list of stop words was adapted to the patent domain.

• Sieves 5, 6, and 7 (*Variants of Strict Head Match*): The mention and its antecedent are not required to match in number so as to allow set-member relations.

• Sieve 8 and 9 (*Proper Head Word Match and Alias Match*): These sieves were disabled, given that neither proper nouns nor aliases appear in patents.

• Sieve 10 (*Relaxed Head Matching*: Is kept as it was.

• Sieve 11 (*Lexical Chain*): The lexical chain sieve was modified so as to use elaborate patterns for detection of 'part-whole' relations and 'instance-of' relations between mentions within the same sentence.

• Sieve 12 (*Pronominal Coreference resolution*): This sieve was modified according to the occurrence of the pronouns in patents. Most personal pronouns were excluded and adverbial pronouns were included.

The new copula sieve distinguishes between attributive relations where the copulative construction indicates a property of the subject (see (a) below for illustration) and identity relations (see (b) for illustration):

(a) [the present invention] **is** [a wind turbine generator]attribute

(b) [the difference of the wind direction deviation]$_i$ **is** [the error of the anemoscope 6 due to drift wind]$_i$

Unlike StCR, in which copulative constructions (<NP_1> *is* <NP_2>) are detected in the precise pattern-based construct, in patents, this construct needed its own sieve, applied towards the end of the sieve application sequence. This is because both mentions NP_1 and NP_2 can be involved in separate lexical chain relations such that an early merge into a single chain would make the subject NP_1 the antecedent of the chain while NP_2 would become unavailable for further linking.

The post-processing stage was adapted to post-process the obtained clusters. First, clusters that contain a mixture of reference relations have been divided into different chains. Second, if compatible mentions (i.e., same head, same attributes, and one smaller mention included in the other larger one) are used to establish different cohesion relations, their clusters are merged into one.

4.3. Performance of the adapted StCR

For the evaluation of the performance of the adapted StCR compared to the original StCR, we follow COALA (Andrews et al., 2007), which compares the coreference annotations of two different algorithms presenting the differences in a format suitable for manual evaluation. We developed a similar tool that, given two annotations, computes all pairs of chains with at least one mention in common and determines the complete context for all mentions involved.

In our evaluation, we focused so far on identity relations, i.e., strict coference (since the original StCR deals only with them), leaving aside other types of lexical chains. Both systems, the original StCR and our adaptation, were run within our pipeline. A set of 100 pairs of aligned chains and their contexts produced by both systems was selected at random, excluding pairs with a 100% overlap. Each chain pair was assessed manually with respect to how many of the mentions in each chain are correct and how many are incorrect. For mentions found in only one of the chains, the evaluators assessed whether their inclusion into other chains is correct or not. In 61%, the output of the adapted version of the StCR had a higher hit rate; in 24%, this was the original StCR, and in 15% the versions were equal.

5. How to adapt StCR to other domains

The exercise of adapting StCR to the patent domain raises an interesting question on how it can be adapted to other domains. Taking as a starting point the case of patents, we outline in this section the steps to take and the issues to consider in order to achieve an appropriate adaptation.

As aleady presented above, the stages involved in the StCR tool are: (i) Preprocessing (tokenization, PoS tagging, lemmatization, chunking and parsing), (ii) Candidate Detection, (iii) Coreference Resolution and (iv) Postprocessing. During adaptation, we need to identify which stage has to be adapted to capture a specific linguistic phenomenon encountered as different in the new domain. Below, we describe the linguistic considerations that need to be taken into account for each stage during the adaptation.

5.1. Preprocessing stage

In order to evaluate whether it is sufficient to use the original preprocessing pipeline included in the StCR, or whether one or more subcomponents should be replaced, the capability of the original pipeline to process the data in the domain in question needs to be assessed with respect to stability and quality. If all data can be processed and the behavior of the pipeline is stable, the output has to be evaluated with respect to its quality in order to decide whether (certain) subcomponents can be retrained with the data of the new domain, or whether it is more appropriate to use external tools.

In the light of our experience with long and complex sentence structures in patent material, for high quality preprocessing it is important to pay particular attention to: (i) the average length of the sentences, (ii) the level of the embeddedness of NPs, (iii) the possibility of the NPs at different levels to be involved in coreference/lexical chains, (iv) the potentially diverging PoS of frequent lexical items. In the case of demanding requirements with respect to (i)–(iii), the use of an external dependency parser and/or chunker is to be considered.

5.2. Candidate detection stage

For an adequate adaptation of the parameters assigned in the candidate detection stage, the following linguistic issues have to be carefully analyzed in the domain to which the StCR will be applied:

• Length of the sentences: The StCR algorithm uses sentence length to establish ordering restrictions when prioritizing candidates for coreference (especially for pronominal antecedents). Thus, it is necessary to adjust these corresponding parameters depending on the average length of the sentences in the new domain.

• Levels of NPs' embeddedness: As already mentioned, the original StCR excludes the base NP (pronominal modifiers + head) as candidate for coreference if it is included in larger NPs (for instance, when prepositional phrases modify the NP). However, in some domains, smaller NPs are enough for establishing coreference, and therefore they need to be kept during the candidate stage. Thus, in general terms, it is necessary to choose the relevant embeddedness level for candidate detection in the new domain. For efficiency purposes, the chosen NPs should be as short as possible, but they should contain all the information that is needed for their recognition as NPs distinct from other NPs with the same head. Once the optimal embeddedness level for choosing the candidates has been identified, it is also necessary to decide whether higher and/or lower levels should be included during the candidate search. Although theoretically all levels should be considered as possible locations of candidates, it is necessary to also take into account the velocity and efficiency of the algorithm when applying the possible coreference matches, as well as the process of merging the coreference chains.

• Adequate inclusion/exclusion of items or types of NPs: Each domain contains its own nominal heads that have the characteristics of stop words (i.e., very abstract, generic or empty of sense), which need to be filtered and thus excluded from the candidate list of possible heads.

Bare NPs and numerical NPs should also be evaluated in each domain. While in general discourse they cannot be coreferential, there are domains in which they can. Thus, it is necessary to study thoroughly the domain to decide on the "candidate status" of different kinds of NPs. At the same time, it is also necessary to adjust the inclusion/exclusion of pronouns detected as candidates for coreference. For instance, while in general discourse personal pronouns are very prominent, they are of minor importance in some domains and therefore should be filtered out.

• Detection of other types of phrases apart from NPs that can corefer: Although in general discourse only NPs can be included within coreference chains, it is possible that in specialized domains other types of phrases could also corefer. For instance, in patents, gerund phrases (which express steps of a method) such as *determining a voltage change* can be coreferential. Therefore, they must be considered as potential candidates.[3]

5.3. Coreference resolution stage

The coreference resolution stage is the core of the StCR, given that it is in charge of relating the elements selected in the previous stage by applying a succession of independent coreference models (sieves). In order to make appropriate adaptations of this stage of the tool, the following issues need to be taken into account:

• Internal structure of the text: In general discourse, the structure of the text is flat, but in certain specialized domains, the text possesses a hierarchical structure. The position of each instance within a given structure can then be very relevant, such that it must be taken into account when establishing coreference. For instance, in patents, the section of claims contains a tree-like structure (there are dependent and independent claims), which limits the scope of possible coreference matches. Thus, if the claims 1 and 5 are independent, the claims 2, 3 and 4 depend on 1, and the claims 6 and 7 depend on 5, an instance in the claim 7 cannot be coreferential with an instance in the claim 2, even if

[3] Obviously, the inclusion of gerund phrases implies a more complex pre-processing stage. Due to the limitations in that stage, our adaptation did not include those phrases as candidates.

they are identical or share the same head.[4]

• Ordering of prioritized candidates: StCR assigns the "left-to-right" order to prioritize candidates within the same sentence for coreference antecedents. However, this parameter should be adjusted for each domain, especially with respect to the average length of the sentences.

• Use of definiteness (through determiners): In general discourse, the definiteness of the NPs serves as a parameter during the search for coreference matches. Thus, an indefinite NP can be the antecedent of a definite NP (identical or sharing the same head), but not of another indefinite NP (even if both are identical). However, in some domains, it is possible to have coreference chains composed of two indefinite NPs (as we saw in patents).

• Other restrictions on an antecedent: The restrictions used when allowing a mention to become an antecedent can change from one domain to another. As mentioned in the previous subsection, in some domains such as patents, bare and numerical NPs can belong to a coreference chain. Thus, it is important not only to detect them as candidates, but also to adjust the parameters assigned to possible matches.

• Limits of relaxation for relaxed matching sieves: StCR can also detect coreferential matches between two non-identical NPs. Even if it is necessary to keep this option, it is essential to know well the linguistic characteristics of the new domain in order to adjust appropriately the parameters assigned to the relaxation. The relevant relaxation parameters (e.g., position/distance of the node with respect to the head, PoS, lexical restriction, etc.) depend on each domain and should be correctly detected. Otherwise, the quality of the output risks to suffer greatly.

• Ambiguity of syntactic configurations: There are syntactic constructions that can imply coreference or some other lexical relations. Thus, it is necessary to evaluate in the new domain what those "ambiguous syntactic constructions" tend to express, in order to decide how we treat them in the process of coreference resolution.

• Multiple coreference: StCR is not designed for resolving coreferences with more than one antecedent.[5] Given that this kind of coreference can be very frequent in some domains (as, e.g., in patents), and that the two antecedents can be distant between each other and not necessarily related by a coordinative conjunction, it is crucial to adapt the tool for being able to cover these cases. In order to do that, three steps are necessary: (i) allow mentions to belong to more than one chain (to achieve this, the merge of chains should also be adjusted accordingly); (ii) adjust the pronominal sieve to address plural and/or distributional pronouns; (iii) tune relaxed match sieves to consider linking multiple singular antecedents to a single plural NP.

5.4. Post-processing stage

Our adaptation of the StCR to the patent domain required to foresee the possiblity that mentions are accomodated in more than one chain because we considered cases of multiple antecedents and because we detected additional types of lexical chains that can overlap with coreference chains. An empirical study of the domain to which StCR is to be adapted should reveal whether these linguistic phenomena are relevant or the original approach to assign each mention to a single chain is sufficient.

[4]This factor was not included in our adaptation because of time limitations.

[5]Even if the original algorithm includes plural personal pronouns (which potentially can have two antecedents), it looks for a single plural NP as antecedent.

6. Conclusions and Future Work

We have shown that the adaptation of a general discourse coreference resolution system such as StCR to the patent domain is feasible and that the adapted version performs considerably better than the original. Some modifications required just a change of the code of mention detection or sieves, whilst others (such as ordering or clustering) were more deeply ingrained. But, in general, our previous experience with the adaptation of general discourse tools to the patent domain was confirmed: it is much less costly to adapt an available general discourse tool than to develop one specifically for the patent domain. We hope that the short outline of how StCR can be adapted to other domains will be found useful for other works.

In the future, we plan to broaden the range of identified semantic relations and to label the recognized relations in the chains.

7. References

Andrews, B., Fan, J., Murdock, J., and Welty, C. (2007). Coala: A tool for inter-document coreference resolution evaluation. In *Proceedings of the AAAI Spring Symposium: Machine Reading*, pages 11–16.

Bohnet, B. (2010). Very high accuracy and fast dependency parsing is not a contradiction. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 89–97, Stroudsburg, PA, USA. Association for Computational Linguistics.

Burga, A., Codina, J., Ferraro, G., Saggion, H., and Wanner, L. (2013). The challenge of syntactic dependency parsing adaptation for the patent domain. In *Proceedings of the ESSLI 2013 Workshop on Extrinsic Parse Improvement (EPI)*.

Cunningham, H. (2011). Text Processing with GATE (Version 6). Technical report, University of Sheffield Department of Computer Science.

Halliday, M. and Hasan, R. (1976). *Cohesion in English*. Longman, London.

Lee, H., Peirsman, Y., Chang, A., Chambers, N., Surdeanu, M., and Jurafsky, D. (2011). Stanford's Multi-Pass Sieve Coreference Resolution System at the CoNLL-2011 Shared Task. In *Proceedings of the CoNLL-2011 Shared Task*.

Lee, H., Chang, A., Peirsman, Y., Chambers, N., Surdeanu, M., and Jurafsky, D. (2013). Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Computational Linguistics*, 39(4).

Morris, J. and Hirst, G. (1991). Lexical cohesion, the thesaurus, and the structure of text. *Computational Linguistics*, 17(1):211232.

Raghunathan, K., Lee, H., Rangarajan, S., Chambers, N., Surdeanu, M., Jurafsky, D., and Manning, C. (2010).

A Multi-Pass Sieve for Coreference Resolution. In *Proceedings of the EMNLP-2010 Conference*.

Recasens, M., Hovy, E. H., and Martí, A. (2010). A Typology of Near-Identity Relations for Coreference (NIDENT). In *Proceedings of the LREC 2010*.