Towards an Integration of Syntactic and Temporal Annotations in Estonian

Siim Orasmaa

Institute of Computer Science, University of Tartu
J. Liivi Str 2, 50409 Tartu, Estonia
siim.orasmaa@ut.ee

Abstract

We investigate the question how manually created syntactic annotations can be used to analyse and improve consistency in manually created temporal annotations. Our work introduces an annotation project for Estonian, where temporal annotations in TimeML framework were manually added to a corpus containing gold standard morphological and dependency syntactic annotations. In the first part of our work, we evaluate the consistency of manual temporal annotations, focusing on event annotations. We use syntactic annotations to distinguish different event annotation models, and we observe highest inter-annotator agreements on models representing "prototypical events" (event verbs and events being part of the syntactic predicate of clause). In the second part of our work, we investigate how to improve consistency between syntactic and temporal annotations. We test on whether syntactic annotations can be used to validate temporal annotations: to find missing or partial annotations. Although the initial results indicate that such validation is promising, we also note that a better bridging between temporal (semantic) and syntactic annotations is needed for a complete automatic validation.

Keywords: temporal annotation, dependency syntax, Estonian

1. Introduction

Knowing temporal structure of text (events, temporal expressions and relations between these entities) supports many Natural Language Processing applications, including question answering, text summarization and machine translation. In recent years, the problem of automatically detecting the temporal structure of text has gained a lot of attention, resulting in development of annotation frameworks TimeML (Pustejovsky et al., 2003) and ISO-TimeML (ISO/TC 37/SC 4/WG 2., 2007). Following these frameworks, proof-of-concept annotated corpora like English TimeBank (Pustejovsky et al. 2003), French TimeBank (Bittar, 2010) and Italian TimeBank (Caselli et al., 2011) have been created. However, there has been little attention on integrating and comparing TimeML annotations with other manually created linguistic annotations, such as syntactic annotations. While both practical applications (machine learning) and research can benefit from consistency between manually created annotations, it is an open question how and to what extent this consistency can be achieved. Current work investigates this question by manually adding temporal annotations to a corpus containing gold standard morphological and syntactic annotations.

The first part of our work focuses on event annotations. The temporal annotations in TimeML framework are largely based on the notion of "event", as "events" are the main units participating in temporal relations. However, TimeML framework provides rather general "event" definition (*event* is "a cover term for situations that happen or occur," and for "states and circumstances in which something obtains or holds true"), and leaves many linguistic details of "event" annotation to be specified at the language level. We study the problem of TimeML-based temporal annotation in Estonian – a language which does not have a strong linguistic basis for the notion of "event" from previous studies – and we note difficulties on establishing consistent "event" annotation. In order to analyse the problem empirically, we use gold standard syntactic annotations to distinguish different event annotation models, and we show that on specific syntactic constructions, there is a high inter-annotator agreement on event annotation.

The second part of our work investigates how to improve consistency between syntactic and temporal annotations. We test whether syntactic annotations can be used to validate temporal annotations: to find missing or partial annotations. Although the initial results indicate that such validation is promising, we also note that a better bridging between temporal (semantic) and syntactic annotations required for a complete automatic validation.

2. Temporal annotations in TimeML-framework

2.1. Overview

In the TimeML framework (ISO/TC 37/SC 4/WG 2., 2007), the annotation of temporal structure is divided into two layers: 1) the layer of annotated entities: event and temporal expressions; 2) the layer of relations between these entities. Although an event (or an event mention in text) can have rather complex linguistic structure which describes event's participants, temporal and spatial circumstances, TimeML proposes to mark up only the word that best represents the event – usually a verb or a noun (Xue and Zhou, 2010). In case of more complex syntactic structures, a general rule is to mark only the head of a construction as an event (Saurí et al., 2009). For example, in the construction *did not disclose* (as in *Kaufman did not disclose details of the deal*), only the verbal head *disclose* is marked as an event.

Annotated events are classified into 7 classes: `ASPECTUAL` (example verbs: *start, continue, finish, ...*), `I_ACTION` (*deny, allow, hinder, ...*), `I_STATE` (*want, believe, like, ...*), `PERCEPTION` (*hear, see, watch, ...*), `REPORTING` (*say, quote, state,...*), `STATE` (*be* + ADJ) and `OCCURRENCE` (the events not belonging to any other classes) (Bittar, 2010). Classes `REPORTING`,

PERCEPTION, ASPECTUAL, I_ACTION and I_STATE refer to events that have a second event as an argument (explicitly present in the sentence) (Saurí et al., 2009). TimeML guidelines also state that these events *select for* a second event as an argument, referring to the semantic nature of this predicate-argument relation. For example, in sentence *John finished building the house*, the event *finished* has the event *building* as an argument.

In addition to events, temporal expressions are also annotated in text. Temporal expressions (*timexes*) are divided into four types: dates (examples: *next Wednesday; June 11, 1989*), times (*at 18.00 o'clock; in the morning*), durations (*five days; last six months*), and sets of times (*on every year; on Wednesdays*). Temporal expressions are annotated at full extent, including all the words modifying temporal semantics of the expression (such as *last, next, every* etc).

The layer of relations between entities consists of three types of relations: temporal relations (TLINKs), aspectual relations (ALINKs) and subordination links (SLINKs). Although finding temporal relations between all the entities is the final goal of the research, the full manual annotation is often infeasible (given the large number of temporal entities in text) and also unnecessary (as some of the relations can be inferred from others) (Xue and Zhou, 2010). So, in practice, the annotation of temporal relations is divided into subtasks, which follow text segmentation and also rely on syntactic relations between entities, e.g. an intrasentential temporal relation is drawn between two events if one event syntactically dominates the other event (TimeML Working Group, 2009). Other two types of links (ALINKs, SLINKs) are based on syntactic relations between entities of certain types. ALINK marks the relation between an aspectual event and its a subordinated event (e.g. relation between *finished* and *eating* in *John **finished** eating*). SLINK marks relations which can change the truth conditions or degree of certainty of event-denoting propositions (Bittar, 2010). An example of such relation is the relation between a perception event and its subordinated event, as between the verb *saw* and the noun *accident* in *John **saw** the accident*.

2.2. Adapting TimeML to Estonian

Estonian-specific annotation guidelines were created adopting from the TimeML specification. In this subsection, we discuss main problems that were encountered during the adoption and our current solutions to these problems, focusing on event annotation.

On creating Estonian specific event annotation guidelines, we tried to follow the example of English event annotation guidelines (Saurí et al., 2009) as much as possible. However, there were two general problems, which we found difficult to address in a comprehensive manner: 1) **decomposition problem**: how complex (multiword) events should be decomposed into markable units (which units should be considered as markables and which can be systematically skipped)?, 2) **problem of "background events"**: in which cases events appearing as "background information" to the main event of the clause (non-verb events and verb events not part of the predicate of the clause) should be considered as markables?

The decomposition problem arises when the main verb of a clause is purely grammatical (such as *olema* 'be') or semantically weak (such as verbs *tegema* 'to make', *andma* 'to give'), so that some additional expression must be added to the verb to express the full meaning of the event (e.g. *olema valmis* 'be ready', *tegema erandi* 'make an exception'). In case of grammatical verb *olema* 'be', we decided that it should be annotated as a multiword expression when it appears with a state denoting noun, adjective or adverb (such as *olema õnnelik* 'be happy') and left unannotated when it appears with an infinite verb in compound tense. We also allow multiword event annotations in light verb constructions (semantically weak main verb + event denoting noun/adjective/adverb) if it is difficult to distinguish the two events as distinct. For example, in the expression *pidas läbirääkimisi* 'held negotiations', it is difficult to interpret the finite verb 'held' as a stand-alone event.

The event decomposition problem also arises with modal verbs. In case of English, modal verbs are considered auxiliary verbs accompanying the main verb of the sentence and are not annotated with EVENT tag. In Estonian, modal verbs can be conjugated in mood, voice, number, person and tense. So, similarly to Italian (Caselli et al., 2011) and French (Bittar, 2010) annotation projects, we decided to annotate modal verbs with EVENT tag and assign a special TimeML class value: MODAL. Modal verbs are considered as EVENTs selecting for an EVENT argument.

The problem of "background events" most frequently arises in case of noun and adjective event mentions (which do not function as arguments of an event selecting for event-denoting argument in TimeML). For example, in sentence *Esimeses geimis pääses meeskond ette ja hoidis edu tänu heale servile* 'The team took the lead in the first game and maintained the lead because of the good serve', it is relatively clear that main verbs *pääses* 'took' and *hoidis* 'maintained' should be annotated as events; however, it is difficult to decide whether nouns *geimis* 'game' and *servile* 'serve' should be annotated as events or whether they should be considered as a "background information" which can be left unannotated. We decided that background events should be annotated: a) if they are governing an annotated temporal expression, or b) if they are directly governed by a verb annotated as event and they appear more than once in the text, thus more likely have important relations with annotated events. However, the criterion b) is still problematic, e.g. one needs to decide whether synonymous references to the background event should also be counted while counting its occurrences.

Verbs can also be considered as "background events" if they are not part of the syntactic or semantic predicate of the clause. For example, in sentence *Medali võitnud sportlane võeti soojalt vastu* 'The athlete who had won the medal was warmly welcomed', the verb participle *võitnud* '(had) won' functions as attribute of the subject of the clause ('athlete') and thus can be considered as a background information for the main event of the clause ('(was) welcomed')[1]. We decided that "background event" verbs should be annotated.

[1] In case of Estonian, *medali võitnud sportlane* (which literally means 'medal won athlete') does not form a separate clause, so the whole sentence forms a single clause.

3. Dependency annotations

3.1. The Constraint Grammar formalism

In current work, the underlying syntactic annotations are based on manually corrected output of syntactic analyser of Estonian (K. Müürisep et al., 2003). Syntactic analyser of Estonian is based on Constraint Grammar (CG) formalism (Karlsson et al., 1995) and its latest version uses VISL CG-3 format and software[2]. It outputs syntactic tags (subject, object etc) for each word-form and dependency structure for each sentence.
Following is an example of dependency structure of the English sentence *John finished building the house*[3] (Estonian annotations have similar structure, although the linguistic categories differ):

```
John [John] N S NOM @SUBJ> #1->2
finished [finish] <mv> V IMPF @FS-STA #2->0
building [build] <mv> V PCP1 @ICL-<ACC #3->2
the [the] ART S/P @>N #4->5
house [house] N S NOM @<ACC #5->3
. [.] PU @PU #6->0
```

Dependency relations can be read from tags `#x->y`, where `x` marks the number of current token and `y` marks its syntactic head (e.g. in previous example, `#3->2` marks that the word 3 (*building*) has the word 2 (*finished*) as its syntactic head.

3.2. Integration with TimeML

In principle, syntactic dependency relations should show how temporal expressions are connected to event expressions and which events function as arguments of other events. However, because TimeML aims to capture semantic relations between entities, it is not always straightforward how these relations can be mapped to dependency syntactic relations. In this subsection, we list the problems that arise when dependency relations are used to find argument events for events requiring argument in TimeML.[4]
Firstly, in some cases **a dependency relation must be reversed** in order to find an argument for the event requiring argument in TimeML. In Estonian dependency annotations, following cases can be distinguished:

1. Some aspectual and modal finite verbs are systematically annotated as dependents of the accompanying infinite verbs. E.g. in *John **hakkas** maja ehitama* 'John began to build the house', the aspectual verb *hakkas* 'began' is a dependent of its argument *ehitama* 'to build'.
2. An event requiring argument in TimeML can function as a syntactic attribute of its TimeML argument. E.g. in ***Taaspuhkenud** vägivald tuleb lõpetada.* 'Reinitiated violence must be stopped', the verb participle *Taaspuhkenud* 'reinitiated' (TimeML `ASPECTUAL` event) is syntactically governed by the event noun *vägivald* 'violence'.
3. Some `REPORTING`, `I_ACTION` and `I_STATE` events are expressed by adverbials which are governed by their TimeML argument - main verb of the clause. E.g. in *Korraldaja **sõnul** toimub üritus detsembris* 'According to the organizer, the event will take place in December' the adverbial phrase *Korraldaja sõnul* 'According to the organizer' is governed by the main verb *toimub* '(will) take place'.

Secondly, an event requiring argument in TimeML can have **multiple dependent events**; however, it is possible that not all of the syntactic dependents are arguments according to the TimeML *class* of the event. E.g. in *Eilsel valitsuse istungil **lubas** peaminister maksu vähendada* 'At yesterday's goverment meeting, the prime minister promised to reduce the tax ' the main verb *lubas* 'promised' has two dependent events: *istungil* '(at the) meeting' and *vähendada* 'to reduce'; however, only *vähendada* is the actual argument according to the TimeML class `I_ACTION`.
Thirdly, the required argument can be **indirect dependent** of the event requiring argument in TimeML, so it must be reached via a path of dependency relations. This mostly happens if the event requiring argument in TimeML is part of a periphrastic verb construction and its non-verb part is not annotated as EVENT; however, the non-verb part syntactically governs the TimeML argument event. E.g. in *Nad **teevad** ettepaneku viimane otsus üle vaadata* 'They will make a proposal to check over the last decision', only the verb *teevad* 'make' is annotated as an EVENT in the periphrastic expression *teevad ettepaneku* 'will make a proposal'; however, it's EVENT argument *(üle) vaadata* 'check over' is dependent of the word *ettepaneku* 'proposal'.

4. The annotation project

The annotation project involved three annotators and one judge. In the main annotation process, each text was annotated by two annotators and assigned to the judge for disagreement resolution.[5] We decided to do double annotation because of the difficulty of the task and because double annotation provides better basis for studies of inter-annotator agreement.
Before the main annotation process, a pilot annotation experiment was made, where all the annotators were provided the guidelines and 5 newspaper articles for annotation. According to the results of the pilot experiment, the guidelines were further elaborated before the main annotation process.

[2] VISL project homepage at the Institute of Language and Communication, University of Southern Denmark: `http://beta.visl.sdu.dk/`

[3] Produced automatically with Machine analysis tool: `http://beta.visl.sdu.dk/visl/en/parsing/automatic/dependency.php`

[4] We refer to events from classes `REPORTING`, `I_ACTION`, `I_STATE`, `ASPECTUAL`, `PERCEPTION` and `MODAL` as *events requiring argument in TimeML*. In TimeML literature, these events are also referred as *events selecting for an event-denoting argument*.

[5] The annotators had background in computational linguistics, but no previous experience with TimeML annotations. The judge had previous experience on adapting the TimeML guidelines to Estonian and on doing some corpus annotation experiments in TimeML.

The source files for the main annotation process were selected articles from three Estonian newspapers: Maaleht, Postimees, and SL Õhtuleht. Each file contains a single article and there are total 80 files consisting of approx. 22 000 tokens (including punctuation).
The main annotation process was separated into 4 iterations. Each iteration was divided into two stages: at the first stage, events and temporal expressions were marked in text, and at the second stage, temporal relations (TLINKs) were annotated between events and between events and temporal expressions.
The first stage was performed manually in a text file containing dependency syntactic annotations. Along with determining the extent of the event and time expressions, annotators were also asked to choose the class of the event and determine temporal expression's type and calendrical value. After the first stage, text files were processed with a script which checked initial validity of the annotation (e.g. detected typos and cases where the annotation did not follow the specified format) and then annotations were manually checked by the judge.
The second stage was performed using Brandeis Annotation Tool (Verhagen, 2010). Following the TempEval-2 guidelines (TimeML Working Group, 2009) on annotation of temporal links, the annotation process was divided into 4 tasks:

1. determine relations between events and temporal expressions;
2. determine relations between events and document creation time;
3. determine relations between main events of two consecutive sentences;
4. determine relations between events in same sentence (intrasentential relations);

Like in TempEval-2, we used a simplified set of temporal relations: `BEFORE`, `BEFORE-OR-OVERLAP`, `SIMULTANEOUS`, `IS_INCLUDED` and `INCLUDES`, `OVERLAP-OR-AFTER`, `AFTER`, `VAGUE` and `IDENTITY`. The elaborate relations `SIMULTANEOUS`, `IS_INCLUDED` and `INCLUDES` were used instead of the general relation `OVERLAP` (which was used in TempEval-2), because in the pilot annotation experiment, annotators often found that the general relation `OVERLAP` was confusing and needed elaboration.

5. Evaluation

5.1. Overall inter-annotator agreements

In this work, we focus on evaluation of inter-annotator agreements on entity (TIMEX, EVENT) extent and on entity attribute filling.
Detailed results for all annotator pairs on entity extent are shown in Table 1. The three annotators are marked as A,B,C, and the judge annotator is marked as J. It can be noted that not all agreements are on similar level: annotators A and B had generally higher agreement among each other than with the annotator C. This trend is also confirmed if judge annotations are considered: A and B had higher agreement with the judge than C.

Layer	AB	AC	BC	JA	JB	JC
EVENT	0.86	0.74	0.77	0.92	0.90	0.75
TIMEX	0.82	0.72	0.80	0.88	0.88	0.76

Table 1: Inter-annotator agreements (F1-scores) on entity extent.

In Table 2, entity extent agreements are aggregated over annotator pairs and, additionally, agreements on entity extent along with the attribute assigning are reported.

Layer	avg F1-score (AB,AC,BC)	avg F1-score (JA,JB,JC)
EVENT-extent	0.793	0.86
TIMEX-extent	0.784	0.842
EVENT with *class*	0.511	0.686
TIMEX with *type*	0.578	0.71
TIMEX with *value*	0.44	0.63

Table 2: Aggregate inter-annotator agreements on entity extent and on entity extent with attributes.

5.2. A study of inter-annotator agreement on EVENT annotation

As described in subsection 2.2., there are various general problems on establishing elaborate EVENT annotation guidelines, and this is also reflected in rather poor inter-annotator agreements on EVENT annotation. To investigate the problem of empirically, we used the gold standard syntactic annotations and attempted to find a subset of EVENT annotations in which the disagreement was minimal.
Based on available syntactic annotations, we hypothesized that a prototypical EVENT: 1) is a verb, and 2) is part of the syntactic predicate of the clause. It was expected that on prototypical EVENTs inter-annotator agreement would be higher than on non-prototypical EVENTs.
In order to test these hypotheses, we made experiments where EVENT annotations were filtered based on morphological and syntactic annotations. EVENT annotations not meeting the filtering criteria were deleted[6] along with all the associated TLINK relations. After the deletion, inter-annotator agreements were measured on remaining annotations. Only annotations of the three annotators were used in the experiment; annotations belonging to the judge were excluded as these are highly dependent on underlying annotations.
Table 3 shows results of experiments, reporting how the deletion affects the EVENT and TLINK coverage, and EVENT extent annotation agreement (average F1 score over annotator pairs AB, AC, BC) on the corpus. Model

[6] In case of multiword EVENTs, an EVENT gets deleted only if its header token (the token with EVENT *class* attribute) does not meet the criteria. Typically, a verb was marked as a header token.

Model	Description	EVENT coverage	TLINK coverage[7]	EVENT extent F1
0	initial (no EVENT filtering)	4550 (100.0%)	16858 (100.0%)	0.793
1a	verbs	2974 (65.36%)	12802 (75.94%)	0.945
1b	verbs and nouns	4273 (93.91%)	16180 (95.98%)	0.824
1c	verbs and adjectives	3193 (70.18%)	13368 (79.3%)	0.907
1d	verbs, adjectives and nouns	4490 (98.68%)	16814 (99.74%)	0.8
2a	EVENTs that are part of the predicate of clause	2608 (57.32%)	11246 (66.71%)	0.984
2b	2a + direct verb dependents of the predicate	2889 (63.49%)	12470 (73.97%)	0.954
2c	2a + direct non-verb dependents of the predicate	3634 (79.87%)	13469 (79.9%)	0.864
2d	2a + clause members not directly dependent of the predicate	3243 (71.27%)	13151 (78.01%)	0.897

Table 3: Annotation coverage and inter-annotator agreement results for different EVENT filtering models. A filtering model specifies which EVENT annotations are preserved in the manually annotated corpus; all the other EVENT annotations are deleted and TLINK annotations covering the deleted EVENTs are also removed.

0 is the initial annotation where no EVENT filtering is applied. Models 1a-1d explore, how part-of-speech affects the inter-annotator agreement, and models 2a-2d explore how belonging to the syntactic predicate affects the agreement. Model groups 1 and 2 are constructed in a following way: a prototypical case is taken as the base model (1a - keep only EVENT verbs; 2a - keep only EVENTs in syntactic predicates) and other models (b-d) are created by extending the base model.

Results of models 1a-1d confirm the hypothesis that verbs are prototypical candidates for EVENT: the highest inter-annotator agreement (0.945) is observed if only EVENT annotations on verbs are preserved. The results also show that the most problematic part-of-speech for EVENT annotation is noun: adding EVENT-noun annotations (model 1b) reduces the agreement to 0.824. Adjectives are less problematic than nouns and this can be explained by their lesser frequency and by Estonian-specific decisions in syntactic annotation. In Estonian, verbal participles are similar to adjectives and are systematically marked as adjectives when appearing in specific positions (Muischnek et al., 1999). We observe that the majority of the annotated adjective EVENTs are past particles functioning syntactically as attributes or predicatives.

Models 2a-2d require that the syntactic predicate is automatically detected for each clause, based on the syntactic tags of words. In Estonian, syntactic predicate has following structure. A finite verb is always part of the predicate and if the finite verb governs all other members of the clause, this is also the only member of the predicate. In cases when the finite verb has grammatical function in the clause (e.g. in case of modal verbs or compound tenses), members of the clause are governed by the infinite verb, so the infinite verb is also included in the predicate. The infinite verb forms the predicate also in cases when there is no finite verb at all (e.g. in case of negation, which is formed using the negative particle *ei* and an infinite form of the verb).

Results of the models 2a-2d (in the Table 3) confirm the second hypothesis: the highest inter-annotator agreement (0.984) is achieved when only members of syntactic predicate are allowed to be annotated as EVENTs. The agreement remains relatively high (0.954) when verbs that are direct dependents of the predicate are additionally kept as EVENTs. This mostly indicates cases of a catenative verb where an infinite verb or a gerundive verb is governed by the predicate. However, when non-verbs are allowed to be annotated as EVENTs in subject, object and adverbial positions, agreement decreases to 0.864. Indirect dependents of the predicate (model 2d) cause smaller decrease in agreement and this can be explained by their smaller frequency amongst the EVENT annotations.

Bethard et al. (2012) observed highest inter-annotator agreement in temporal annotation when direct speech, modal, negated, hypothetical and aspectual events were omitted from the timeline. We also made some experiments on filtering EVENTs by TimeML *class* in order

[7] In case of counting EVENT coverage, each token with unique position in text was counted once, regardless how many different annotators had annotated it. In TLINK coverage, all TLINKs were counted, including TLINKs between same entities suggested by different annotators.

to test whether specific EVENTs requiring arguments in TimeML along with their arguments are more problematic in EVENT annotation. However, because of the problems in fully integrating dependency annotations with argument structure of TimeML EVENTs (as discussed in section 3.2.) and because of the low inter-annotator agreement on EVENT classification, results of these experiments may not be reliable and are not reported here.

6. Syntax-based validation of the temporal annotations

As temporal annotation is difficult task for humans, it is important that created annotations are consistent with underlying syntactic annotations. If certain level of consistency is achieved, temporal annotations can be decomposed into linguistically motivated substructures and systematically analysed in order to provide more elaborate annotation guidelines, which in turn will foster more consistent temporal annotation creation (Marşic, 2012).

In order to check consistency between temporal annotations and syntactic annotations, we used gold standard dependency annotations to find missing or incomplete parts in temporal annotation. Here, we give an overview which inconsistencies were sought, what are the results and what problems remained unsolved.

6.1. Finding missing argument EVENTs

According to TimeML, EVENTs belonging to classes `REPORTING`, `I_ACTION`, `I_STATE`, `ASPECTUAL`, `PERCEPTION` (and `MODAL`) require an EVENT argument in the same sentence. Dependency syntactic annotations can be used to check whether the EVENTs that require argument actually have an argument: an argument EVENT should be a syntactic dependent of the EVENT that requires the argument. Though, as discussed in section 3.2., there are cases when finding the actual argument(s) is not straightforward, e.g. in cases of reversed dependency relations and multiple argument candidates.

We checked for existence of required EVENT arguments in the last version of the corpus (annotations corrected by the judge). For each EVENT that required argument, we first checked whether its direct dependents were annotated as EVENTs; if not, then indirect dependents were checked; and finally, if no EVENT was found amongst the descendants, it was checked whether direct parent of the EVENT was an EVENT (the case of reversed dependency). Table 4 shows the results of the checking procedure: how EVENTs requiring argument are distributed over argument structures suggested by dependency annotations.

For majority of EVENTs requiring argument (79.2%), a single EVENT argument was found. However, there was a large proportion of cases (24.9 %), where the argument was found via reversed dependency relation, so there was mismatch between TimeML argument structure (semantic argument structure) and the argument structure suggested by dependencies (syntactic argument structure). In case of EVENTs with multiple arguments (20.1% of all EVENTs requiring arguments), an extra effort is required to choose the correct arguments among all found arguments and we did not attempt to do this automatically. And finally, in small fraction of cases (7 EVENTs) the arguments were not found via syntactic relations. However, a closer inspection revealed that in all of these cases the actual argument EVENTs were present in sentence. The argument EVENTs were not found due to various limitations in currently implemented logic (e.g an argument was not found if it was governed by a coordinate of the argument demanding EVENT).

Description of argument structure	Frequency	Proportion
one EVENT argument	807	79.2%
via dependency link	546	53.58%
via dependency path	7	0.69%
via reversed dependency	254	24.93%
multiple EVENT arguments	205	20.12%
no EVENT arguments	7	0.68%
Total	1019	

Table 4: Distribution of EVENTs requiring arguments in TimeML over the EVENT argument structures suggested by dependency syntactic annotations. In case of EVENTs with one argument, following cases are distinguished: *via dependency link* - the argument is a direct child of the EVENT, *via dependency path* - the argument is an indirect child of the EVENT, and *via reversed dependency* – the argument governs the EVENT itself via a single dependency relation.

6.2. Finding missing TLINK annotations

According to TempEval-2 TLINK guidelines (TimeML Working Group, 2009), intrasentential EVENT-TIMEX and EVENT-EVENT temporal relations should be annotated in cases where one entity syntactically dominates other entity (and in cases when TIMEX and EVENT occur in the same noun phrase). Following these guidelines, we searched for missing TLINKs in cases where a TIMEX is governed by an EVENT or an EVENT is governed by other EVENT.

In the last version of the corpus, there are 490 cases of a dependency relation between an EVENT and the head of a TIMEX phrase. We found that only in 17 of these cases, TLINK was not provided by the judge. Manual inspection revealed that 6 of these 17 cases were actual missing relations. The remaining cases were problematic, such as temporal expressions used in comparison (e.g. 'As in last year, the festival program **includes** several highly nominated movies') or coordinated temporal expressions (e.g. 'The show will **take place** tomorrow in Türi and at Wednesday in Viljandi'). If the event associated with the temporal expression is not explicitly present in text, like in last two examples, one possible solution is to create an empty EVENT tag (representing implicit EVENT) and to link the temporal expression to the new EVENT. However, this would still be problematic regarding the syntactic dependency structure, as the new EVENT would not be connected with the dependency tree of the sentence.

Problematic were also cases where the temporal expression was syntactically governed by the finite verb; how-

ever, semantically it would have been more appropriate to be dependent of the infinite verb (a subordinate of the finite verb). E.g. in *Ta **kavatseb** tagasi **jõuda** tuleva aasta märtsis* 'He intends to return in March next year' the temporal expression was governed by the finite verb *kavatseb* 'intends' instead of the semantically more appropriate infinite verb *(tagasi) jõuda* 'to return'.

In the last version of the corpus, there are total 2144 cases where one EVENT syntactically governs other EVENT in a sentence. While validating whether TLINK relations were marked between these EVENTs, we found 121 cases of missing TLINKs. In a majority of cases (73 of 121), TLINKs were missing between EVENTs from different clauses, which indicates that intraclausal dependency relations are more likely to be missed by annotators.

We inspected manually temporal relations missing inside a clause and found that they were frequently between EVENTs forming a periphrastic verb construction or a catenative verb. In case of periphrastic verb constructions (such as in 'Half of the production goes to export'), this indicated problems in underlying EVENT annotation: a single multiword EVENT should have been formed or, alternatively, only the verb should have been annotated as an EVENT. TLINKS missing between EVENTs in a catenative verb indicated difficulties on determining the temporal relation, such as in case of negation e.g 'The specialist did not want to predict the outcome' or in case of usage of modal verb e.g. 'The problem must be solved'.

7. Discussion

7.1. A study on EVENT inter-annotator agreement

In this article, we have shown that there is a high inter-annotation agreement (F-score 0.98) on EVENT annotations that are part of the syntactic predicate of a clause, and the agreement decreases if the EVENT annotations are extended to involve event-denoting words outside the syntactic predicate.

The finding supports the intuition that verbs being part of the syntactic predicate are prototypical "events", on which high inter-annotator agreement can be reached. The agreement remains relatively high (0.95), if, in addition to EVENTs in predicates, verb EVENTs functioning as subjects, objects or adverbials of clause are considered. Such high-agreement annotations covered 64% of the EVENT annotations, and supported 74% of the TLINK annotations. However, remaining 36% of the EVENT annotations (which mostly were EVENT nouns) were problematic in terms of achieving high-inter-annotator agreement.

One of the limitations of our study is that the annotators had limited previous experience with TimeML annotations. It can be argued that experienced annotators would have performed better, especially on difficult noun EVENT annotations. However, detailed analysis of pairwise annotation agreements (Table 1) shows that two of the annotators adapted the guidelines rather well, reaching relatively high inter-annotator agreements, despite having limited previous experience. The systematic bias introduced by the third annotator indicates that the EVENT guidelines as a whole did not provide a common ground upon which all annotators agreed.

Another limitation of our study is that multiword EVENT annotations were allowed. While the multiword annotations can be a convenient way to annotate constructions based on semantically weak and grammatical finite verbs, detecting these constructions is difficult as it requires sophisticated linguistic knowledge. A more "common ground" solution would have been to annotate only the finite verb parts of these constructions, and high inter-annotator agreement on syntactic predicates also seems to support such annotation choice.

In conclusion, the results show that syntactic annotations can be used to establish a "common ground" on EVENT annotations (annotations with high inter-annotator agreement). This also suggests that the EVENT annotation process can divided into tasks requiring different levels of linguistic expertise: annotations on the syntactic predicates and on verbs are relatively easy and can be done with high inter-annotator agreement, but the annotations involving EVENT nouns and adjectives require more sophisticated linguistic expertise, and thus are better to be assigned to expert annotators.[8]

In future work, we will investigate how different syntactic constructions affect inter-annotator agreements on temporal relation (TLINK) annotations. In addition, we investigate how the obtained inter-annotator agreement results can be integrated with the final version of the corpus, so the corpus users can distinguish "low-agreement" and "high-agreement" subcorpora.

7.2. Syntax based validation of TimeML annotations

Our results on syntax based validation of TimeML annotations suggest that if temporal annotations are systematically checked based on manual dependency syntactic annotations, incomplete and problematic parts of the temporal annotation can be revealed.

In previous works, TimeML annotations have been validated for annotation format consistency (Derczynski et al., 2013), and for temporal link logical consistency and sufficient coverage (Derczynski and Gaizauskas, 2010). While a syntactic validation is also desirable because annotation guidelines often use syntactic specifications, it is more difficult to formalise and it is sensitive to the mismatches between syntactic and semantic structures.

Our study highlighted the mismatches between syntactic event argument structure (defined by syntactic dependency relations between events) and TimeML event argument structure (defined by events selecting for argument events). Checking the last version of the corpus, we found that 79% of the events selecting for event argument can be associated with a single event argument via dependency relations. However, finding event arguments for remaining events selecting for argument requires that additional knowledge is encoded in annotations, e.g. in case of multiple argument candidates, obligatory arguments must be distinguished from optional arguments. TimeML ALINK and SLINK relations should serve this purpose, and in future work, we plan to investigate whether these relations can be created semi-automatically from dependency relations.

[8] We want to thank anonymous reviewer for pointing out to this idea.

Our results on validating temporal relations were more promising and revealed cases of inconsistencies between temporal and syntactic annotation. However, it must be noted that our validation does not ensure sufficient coverage of temporal relations, nor does it indicate that the coverage is insufficient, because temporal relations that do not follow the dependency relations are not considered. In future work, we plan to use the validation tool CAVaT (Derczynski and Gaizauskas, 2010) to also validate our corpus for temporal link coverage and logical consistency.

In conclusion, our results indicate that manual syntactic annotations can be used to validate manual temporal annotations to a large extent. However, complete automatic validation also requires better bridging between syntactic and temporal (semantic) annotations.

8. Conculsions

We have introduced an annotation project for Estonian, where temporal annotations in TimeML framework were manually added on top of gold standard linguistic annotations (morphological and dependency syntactic annotations). On analysing the consistency of temporal annotations, we focused on event annotations and we showed that on specific syntactic constructions, there is a high inter-annotator agreement.

We also experimented on syntax-based validation of temporal annotations. Although the initial results indicate that such validation is promising, we also note that the complete automatic validation requires additional work on bridging the gap between TimeML annotations and syntactic annotations.

9. Acknowledgements

This work was supported by Estonian IT Academy program, by the European Regional Development Fund through the Estonian Centre of Excellence in Computer Science (EXCS), by Estonian Ministry of Education and Research (grant IUT 20-56 "Computational models for Estonian"), and by European Social Funds Doctoral Studies and Internationalisation Programme DoRa, which is carried out by Foundation Archimedes.

10. References

Steven Bethard, Oleksandr Kolomiyets, and Marie-Francine Moens. 2012. Annotating Story Timelines as Temporal Dependency Structures. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may.

A. Bittar. 2010. *Building a TimeBank for French: a Reference Corpus Annotated According to the ISO-TimeML Standard*. Ph.D. thesis, Université Paris Diderot, Paris, France.

Tommaso Caselli, Valentina Bartalesi Lenzi, Rachele Sprugnoli, Emanuele Pianta, and Irina Prodanof. 2011. Annotating Events, Temporal Expressions and Relations in Italian: the It-Timeml Experience for the Ita-TimeBank. In *Linguistic Annotation Workshop*, pages 143–151. The Association for Computer Linguistics.

Leon Derczynski and Robert J. Gaizauskas. 2010. Analysing Temporally Annotated Corpora with CAVaT. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*. European Language Resources Association.

Leon Derczynski, Hector Llorens, and Naushad UzZaman. 2013. TimeML-strict: clarifying temporal annotation. *CoRR*, abs/1304.7289.

ISO/TC 37/SC 4/WG 2. 2007. *Language Resource Management - Semantic Annotation Framework (SemAF) - Part 1: Time and events*.

K. Müürisep, T. Puolakainen, K. Muischnek, M. Koit, T. Roosmaa, and H. Uibo. 2003. A New Language for Constraint Grammar: Estonian. In *International Conference Recent Advances in Natural Language Processing RANLP 2003*, pages 304–310, Borovets.

F. Karlsson, A. Anttila, J. Heikkilä, and A. Voutilainen. 1995. *Constraint Grammar: a Language Independent System for Parsing Unrestricted Text*. Mouton de Gruyter.

G. Marşic. 2012. Syntactically Motivated Task Definition for Temporal Relation Identification. *Special Issue of the TAL (Traitement Automatique des Langues) Journal on Processing of Temporal and Spatial Information in Language - Traitement automatique des informations temporelles et spatiales en langage naturel*, vol. 53, no. 2:23–55.

K. Muischnek, K. Müürisep, and T. Puolakainen. 1999. Automatic Analysis of Adjectives in Estonian. In *Workshop in TALN99 (6eme Conference Annuelle sur le Traitment Automatiques des Langues Naturelles)*, TALN99, pages 108–114.

James Pustejovsky, José Castaño, Robert Ingria, Roser Saurí, Robert Gaizauskas, Andrea Setzer, and Graham Katz. 2003. TimeML: Robust specification of event and temporal expressions in text. In *Fifth International Workshop on Computational Semantics (IWCS-5)*.

Roser Saurí, Lotus Goldberg, Marc Verhagen, and James Pustejovsky. 2009. *Annotating Events in English TimeML Annotation Guidelines*. `http://www.timeml.org/tempeval2/tempeval2-trial/guidelines/EventGuidelines-050409.pdf`.

TimeML Working Group. 2009. *TLINK Guidelines*. `http://www.timeml.org/tempeval2/tempeval2-trial/guidelines/tlink-guidelines-081409.pdf`.

M. Verhagen. 2010. The Brandeis Annotation Tool. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*. European Language Resources Association.

Nianwen Xue and Yuping Zhou. 2010. Applying Syntactic, Semantic and Discourse Constraints in Chinese Temporal Annotation. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 1363–1372, Stroudsburg, PA, USA. Association for Computational Linguistics.