XRCE-T: XIP Temporal Module for TempEval Campaign

Caroline Hagège
XEROX Research Centre Europe
6, chemin de Maupertuis
38240 MEYLAN, FRANCE
Caroline.Hagege@xrce.xerox.com

Xavier Tannier
XEROX Research Centre Europe
6, chemin de Maupertuis
38240 MEYLAN, FRANCE
Xavier.Tannier@xrce.xerox.com

Abstract

We present the system we used for the TempEval competition. This system relies on a deep syntactic analyzer that has been extended for the treatment of temporal expressions. So, together with the temporal treatment needed for TempEval purposes, further syntactico-semantic information is also calculated, making thus temporal processing a complement for a better general purpose text understanding system.

1 General presentation and system overview

Although interest in temporal and aspectual phenomena is not new in NLP and IA, temporal processing of real texts is a topic that has been of growing interest in the last years (Mani et al. 2005). The usefulness of temporal information became clear for a wide range of applications like multi-document summarization and question/answering systems (see for instance (Schilder et al. 2003)). For this reason, temporal taggers and annotated resources such as TimeBank (Pustejovsky et al., 2003) have been developed recently.

The work we perform concerning temporal processing of texts is part of a more general process in text understanding. Temporal processing is integrated into a more generic tool, XIP, which is a general purpose linguistic analyzer (Aït et al., 2002). Temporal analysis is thus intertwined with syntactico-semantic text processing including deep syntactic analysis and determination of thematic roles (Hagège and Roux 2003).

In a first part of this article, we present briefly our general purpose analyzer XIP and explain how temporal processing is integrated into it. Then we go into further details and, after some preliminary important comments, we show how we perform our three-level temporal processing. TempEval experiments of our system are finally described; we expose the results we obtained together and investigate the errors that we get. This study will drive the future improvements of the system.

1.1 XIP – a general purpose deep syntactic analyzer

Our temporal processor, called XTM (for XIP Temporal Module), is an extension of XIP (Xerox Incremental Parser (Aït Mokhtar et al., 2002)). XIP performs robust and deep syntactic analysis. *Robust* means here that any kind of text can be processed by XIP (including output of an OCR system or ill-formed input). And *deep* means that linguistic information extracted by the parser can be of a subtle nature and not necessarily straightforward. XIP extracts not only superficial grammatical relations in the form of dependency links, but also basic thematic roles between a predicate (verbal or nominal) and its arguments. For syntactic relations, long distance dependencies are computed and arguments of infinitive verbs are handled. See (Brun and Hagège 2003) for details on deep linguistic processing using XIP.

XIP is rule-based and its architecture can roughly be divided into the three following parts:

- A pre-processing stage is integrated into XIP and handles tokenization, morphological analysis and POS tagging.
- A surface syntactic analysis stage consists in chunking the input. This stage also in-

cludes a Named Entity Recognition (NER) process.

- A deeper syntactic analysis performs first a simple syntactic dependency analysis and then a deep analysis.

Further extensions to the core XIP analysis tool, dealing for example with pronominal co-reference or metonymy of named entities, have been developed and can be plugged in.

Figure 1 synthesizes the general architecture of XIP.

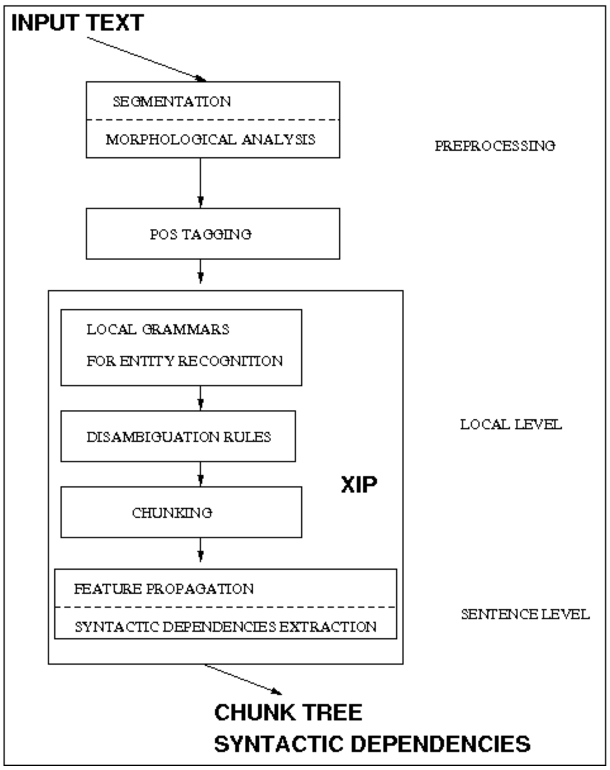

Figure 1: XIP general architecture.

1.2 Intertwining temporal processing and linguistic processing

Temporal processing is integrated into the general linguistic processing engine. The underlying idea is that temporal processing is one of the necessary steps in a more general task of text understanding. For this reason, all temporal processing at the sentence level is performed together with other tasks of linguistic analysis. Association between temporal expressions and events is considered as a particular case of the more general task of attaching thematic roles to predicates (the TIME and DURATION roles). On the other hand, a proper tagging of temporal expressions is beneficial to the task of parsing, because the proper treatment of these complex expressions avoids possible errors in general chunking and dependency calculus. We will detail in sections 3.1 and 3.2 how low-level temporal processing is combined with the rest of the linguistic processing.

1.3 Specific temporal treatment

Low-level temporal processing is not sufficient to deal with all the problems that are raised when one wants to order chronologically events in a whole document, or in the context of multi-document processing. Reference resolution of temporal expressions needs a context that is wider than the sentence level. Temporal calculus and inference also need special treatments that are not of the responsibility of a syntactico-semantic processor. We detail in section 3.3 this document level temporal processing.

2 Preliminary definitions

We define here more precisely what we mean by “temporal relations” and by “events”.

2.1 Temporal relations

The set of temporal relations we use is the following: AFTER, BEFORE, DURING, INCLUDES, OVERLAPS, IS_OVERLAPED and EQUALS (see Figure 2). They are defined as equivalent or disjunctions of Allen’s 13 relations (Allen, 1984). They are simpler than them, which hardly make sense in most fuzzy natural language situations, but they keep basic properties of Allen algebra, as mutual exclusivity, exhaustivity, inverse relations and the possibility to compose relations. This choice is explained in more details in (Muller and Tannier, 2004).

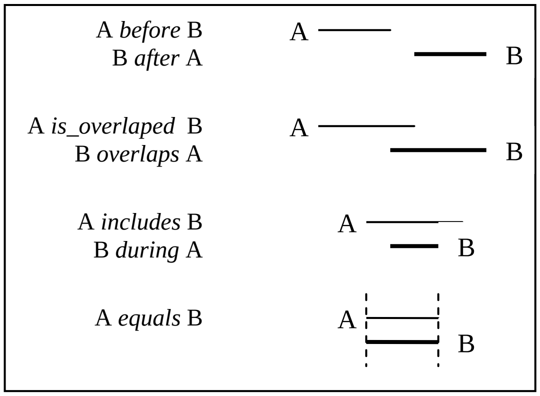

Figure 2: Temporal relations used in XTM

2.2 Events

Temporal expressions attachment and temporal ordering hold for events. Definition of what is an event is not straightforward. In our approach, we decided to consider as events (that can be temporally marked) the following linguistic elements:

- Any verb (expressing either an action or a state)
- Any deverbal noun, when there is a clear morphological link between this noun and a verb (e.g. "interaction" is a derivation of the verb "interact")
- Any noun which is not a deverbal noun and that can be either:
    - an argument of preposition "during"
    - a subject of verb "to last"
    - a subject of verbs "to happen" or "to occur", when these verbs are modified by an explicit temporal expression.

We called these nouns time span nouns. Examples of such nouns are words like "sunrise" or "war", that intuitively correspond to nouns denoting an event of certain duration.

Obtaining this list is done by applying the above-mentioned heuristics to the collection Reuters corpora at NIST and by removing all deverbal nouns from the obtained list.

3 Three levels of temporal processing

We distinguish in our system three main levels during the processing of temporal expressions. This temporal processing has the following purposes:

- Recognizing and interpreting temporal expressions
- Attaching these expressions to the corresponding events they modify
- Ordering these events using a set of temporal expressions we present above.

3.1 Local level

At this level, the main task is the recognition of temporal expressions and the attribution of a value to these expressions.

Recognition of temporal expressions is performed by local rules to which optional left and right contexts can be added. This is done using the XIP formalism and this processing stage occurs just before general chunking rules. Together with contextual rules, some actions are associated. These actions are meant to attribute a value to the resulting temporal expression (left hand side of the rule). Technically, these actions are represented by calls to Python functions that can be executed directly from XIP (Roux, 2006). Figures 3 and 4 illustrate this stage with two examples of rules, the first one for a simple anchor date and the second one for sets. The first one builds an ADV (adverbial) node with associated Boolean features (on the left of the "=" symbol) from linguistic expressions such as "4 years ago" (which matches the right hand side of the rule between "=" and the keyword "where"). Note that there is a call to function "merge_anchor_and_dur" which parameters are three linguistic nodes (`#0` represents the resulting expression on the left hand side of the rule). The second rule is of the same kind. The representation of the values is close to TimeML format (Saurí et al, 2006).

3.2 Sentence level

The sentence level corresponds roughly to the post-chunking stage in a XIP grammar. Once chunks and local grammar expressions have been delimited, relations between linguistic nodes are

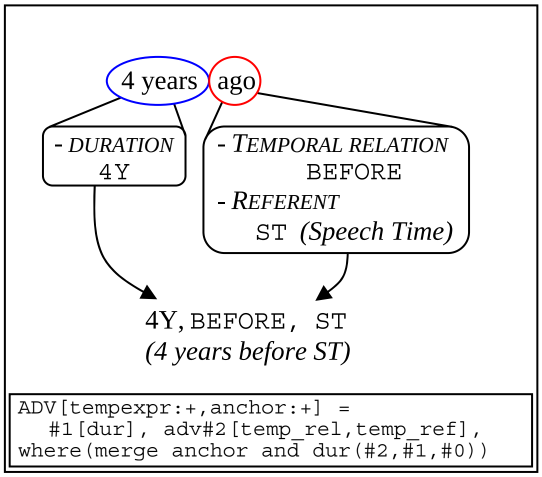

Figure 3: Local level processing, anchor date.

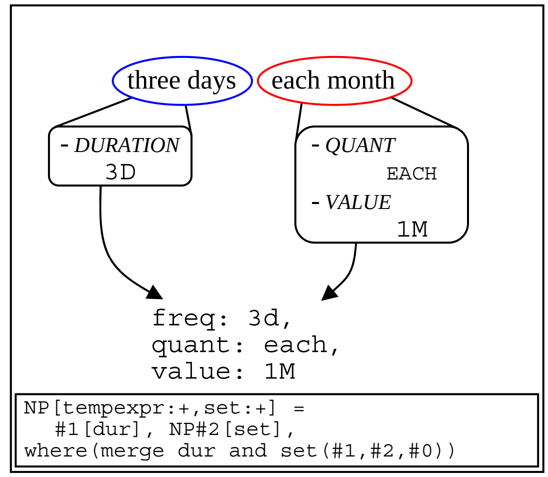

Figure 4: Local level processing, set.

established. These relations represent syntactic and semantic dependencies between linguistic elements. For instance, the grammatical relation SUBJECT is established between the head of a subject noun phrase (NP) and the verb. This is the natural place where some links between temporal expressions and the events they modify are established, as well as temporal relations between events in a same sentence. Verbal tenses are also explicitly extracted at this stage by using morphological information coming from the pre-processing stage.

Attaching temporal expressions to events

As a XIP grammar is developed in an incremental way, at a first stage, any prepositional phrase (PP), included temporal PP, is attached to the predicate it modifies through a very general MOD (modifier) dependency link. Then, in a later stage, these dependency links are refined considering the nature and the linguistic properties of the linked constituents.

In the case of temporal expressions, which have been previously recognized at the local level, a specific relation TEMP links each temporal expression to the predicate it is attached to.

For instance, in the following sentence (extracted from trial data):

> People began gathering in Abuja *Tuesday* for the *two day* rally.

The following dependencies are extracted:

```
TEMP(began, Tuesday)
TEMP(rally, two day)
```

"Tuesday" being recognized as a date and "two day" as a duration.

Temporal relations between events in the same sentence

Using the results of the linguistic analysis, which is able to detect the structure of a sentence (i.e. what is the main verb, where are the embedded clauses depending on this main verb, what kind of subordination holds between the verb, what is the sequence of tenses), some intra-sentential temporal ordering is possible.

Using the temporal relations presented above, the system can detect in certain syntactic configurations if predicates in the sentence are temporally related and what kind of relations exist between them. When it is explicit in the text, a temporal distance between the two events is also calculated.

The two following examples illustrate these temporal dependencies:

> This move comes *a month after* Qantas suspended a number of services.

In this sentence, the clause containing the verb "suspended" is embedded into the main clause headed by "comes". These two events have a temporal distance of one month which is expressed by the expression “a month after”. We obtain the following dependencies:

```
ORDER[before](suspended, comes)
DELTA(suspended, comes, a month)
```

Expressing that the event “suspended” is before the event “comes” with an interval of “a month” (analyzed as a duration which value has been calculated at local level, see section 3.1).

In the second example:

After ten years of _*boom*_*,* they’re _talking_ about layoffs.

“boom” is embedded into “talking” clause, and an ordering can be inferred, as well as a duration of the event “boom”:

```
ORDER[before](boom, talking)
TEMP(boom, ten years)
```

Verbal tenses and aspect

Morphological analysis gives some information about tenses. For instance, the form “said” bears the feature “past:+” indicating that this form is a past tense. However this information is not enough because it is only attached to a single lexical unit.

As verbal forms appear very often as a combination of different lexical units (auxiliaries, past-participles, gerunds, bare infinitives etc.) together with morphological flexion on the finite forms, we have to take all these elements into account in order to decide what the final tense of the whole verb chain is. This final tense may be underspecified in the lack of context.

For instance, for the chain “has been taken”, we extract the following information: “take” is the semantic head of the verbal chain, the aspect is perfective and the tense of the auxiliary “has” is present, which is expressed the following way:

```
VDOMAIN(have,take)
VASPECT_PERFECT(take)
VTENSE_PRESENT(take)
   and “have” bearing feature
   [pres:+].
```

From this information, we deduce that this form is either in present or in past. This is expressed the following way:

```
PRES-OR-PAST(taken).
```

If no other information is obtained for this particular event, the inferred TempEval relation with the document creation time (DCT) will be BEFORE-OR-OVERLAP.

3.3 Document level

Beyond sentence-level, the system is only at the first stage of development. We are only able to complete relative dates in some cases, and to infer new relations with the help of composition rules, by saturating the graph of temporal relations (Muller and Tannier, 2004).

Dates which are relative to speech time can be calculated from the document creation time, when available. We use a fine-grained but fuzzy temporal calculus module. For example, considering a DCT on March 30, 2007, the expression “2 years ago” rarely refers to March 30, 2005 (unless explicit adverbs like “exactly”).

Each unit of time has a “fuzzy granularity”. For example, for minutes:

- “17 minutes ago” means “exactly 17 minutes ago”, not 16 or 18
- “15 minutes ago” or “20 minutes ago” can be understood as fuzzy, because the “fuzzy granularity” (FG) of minutes is 5 minutes.

For years, the FG is also 5 (cf “17 years ago” versus “20 years ago”).

4 Adapting XTM to TempEval specifications

The TempEval track consists of three different tasks. In task A, the aim is to identify temporal relations between some given events and *all* temporal expressions of the sentence. In task B, the same events must be related to the Document Creation Time (DCT). Finally, task C deals with the temporal ordering of consecutive events. See (Verhagen et al., 2007) for more details about the different tasks.

TempEval guidelines present several differences with respect to our own methodology. These differences concern definitions of relations and events, as well as choices about linking.

4.1 TIMEX3 definition

TimeML definition of a temporal expression (TIMEX3) is slightly different from what we consider to be a temporal expression in XTM:

- First, we incorporate signals (in, at ...) into temporal expressions boundaries. But, as TIMEX3s are provided in the test collection, a simple mapping is quite easy to perform.
- The different tokenization for complex expressions is more problematic. We consider that a minimal temporal expression (a temporal token) is the smallest part of a phrase:
    - that is syntactically valid
    - whose meaning, when associated to the predicate it modifies, is true if the whole temporal expression associated to the same predicate is also true.

  For example, “ten days ago yesterday" is a single temporal expression, because no smaller part (e.g. “ten days ago” and “yesterday”) is true in the same context. On the other hand, "during 10 days in December" should be split into "during 10 days" and "in December", both syntactically and semantically valid.

The two following examples illustrate the difference from TimeML:

- TimeBank/TempEval: Hernandez was kidnapped at **<TIMEX3>***10 p.m. Wednesday*<**TIMEX3>**
  XTM: Hernandez was kidnapped [*at 10 p.m.*] [*Wednesday*].

- TimeBank/TempEval: He's expected to meet with Tariq Aziz **<TIMEX3>***later this afternoon***</TIMEX3>**.
  XTM: He's expected to meet with Tariq Aziz [*later*] [*this afternoon*].

4.2 TIMEX3 linking

XTM does not handle temporal relations between events and durations. In our temporal model, an event can have duration. However, this is not represented by a temporal relation, but by an attribute of the event. Durations included in a larger temporal expression (like in “two days later”) introduce an interval for the temporal relation: AFTER(A, B, interval: two days). Here again no temporal relation is attributed with respect to the duration.

Therefore, we had to adapt our system so that it is able to infer at least some relations between events and durations. We used two ways to do so:

- An event having an explicit duration attributed by XTM gets the relation OVERLAP with this duration.
- An event occurring, for example, “two days after another one” (resp. “two days before”) gets the relation AFTER (resp. BEFORE) with this duration.

Other relations are found (or not) by composition rules.

4.3 TIMEX3 values

TempEval test collection provides a "value" attribute for each TIMEX3. However we did not use this value, because we wanted to obtain an evaluation as close as possible to a real world application. The only value we used was the given Document Creation Time, since this particular date was often impossible to guess from the text without an *ad hoc* origin-dependent processing (for example, "CNN19980126.1600.1104", being the only reference to the DCT in a file).

4.4 EVENTs mapping

Event lists do not match either between TempEval corpus and our system analysis. Unfortunately, when a TempEval EVENT is not considered as an event by XTM, we did not find any successful way to map this EVENT to another event of the sentence.

4.5 Temporal relation mapping

Obtaining TempEval relations from our own relations is straightforward:

- AFTER and BEFORE are kept just as they are.
- OVERLAPS, IS_OVERLAPED, DURING, INCLUDES or disjunctions of these relations are turned into OVERLAP.
- Disjunctions of relations containing AFTER (resp. BEFORE) and OVERLAP-like relations are turned into OVERLAP-OR-AFTER (resp. BEFORE-OR-OVERLAP).

5 Results

Provided trial, training and test sets of documents were both subsets of the annotated TimeBank corpus. For each task, two metrics are used. The strict measure considers, for example, that attributing a BEFORE-OR-OVERLAP relation, while the gold-standard suggests BEFORE, is totally wrong. The relaxed measure (see also (Muller and Tannier, 2004)) awards a weighted score to such partly correct relations.

Our rule-based analyzer is designed to favor precision. As the general purpose of our system is text understanding and information extraction, we consider that the most important is to avoid as many erroneous temporal relations as possible. That is why, at least for tasks A and B, we do not assign a temporal relation when the parser does not find any trustworthy link. For the same reason, in our opinion, the strict measure is not as valuable as the relaxed one. Since disjunctive relations are allowed, a strict binary metric is senseless.

Full results are given in the TempEval overview paper (Verhagen et al., 2007). We give a few more information in this section, as well as a short investigation into our system failures.

5.1 Results

- Tasks A and B

Tasks A and B were evaluated together. We obtained the best precision for relaxed matching (0.79 for task A, 0.82 for task B), but with a low recall (respectively 0.50 and 0.60). Strict matching is not very different. Another interesting figure is that less than 10% of the relations are totally incorrect (e.g.: BEFORE instead of AFTER). As we said, this was our main aim.

Note that if we choose a default behavior (OVERLAP for task A, BEFORE for task B, which are respectively the most frequent relations) for every undefined relation, we obtain precision and recall of 0.69, which is lower than but not far from the best team results.

- Task C

This task was more exploratory. The document-level stage of our system is not fully developed yet. Even more than for task AB, the fact that we chose not to use the provided TIMEX3 values makes the problem harder. Our gross results are quite low, and we used an OVERLAP default for each unfound relation and finally got equal precision and recall of 0.58, which is the second best score.

However, assigning OVERLAP to all 258 links of task C led to precision and recall of 0.508; no team managed to bring a satisfying trade-off in this task.

5.2 Error analysis

Most of the errors we get are consequences of errors or underspecification of our linguistic processor.

As we strongly rely on the quality of our parser to perform the task, some missing or erroneous information in the parsing chain have consequences for the final results. Those errors can occur at any level of the processing chain. A POS tagging error can impede to detect a potential event and as a consequence this event will not be temporally annotated, an erroneous PP attachment can lead to mark events in a wrong way.

Many events that could not have been linked to TIMEX3, and especially to the DCT, are nouns. The system lack lexical semantic knowledge concerning nouns and the verbs they modify. The tense of the verb, without any semantic information, is useless. For example, in “we saw an explosion”, the explosion is before DCT because of the simple past verb; but in “he accepted a treaty”, the treaty can be after DCT. Only more knowledge will allow making progress.

6 Conclusion and further work

We described in this paper the system that we adapted in order to participate to TempEval 2007 evaluation campaign. This rule-based system performed quite well for assigning temporal relations between events and temporal expressions in the text. Especially, we obtained a good precision score and a very low rate of incorrect relations, which makes the tool robust enough for information extraction applications.

For the system to be used in real-life applications like information extraction, question-answering, summarization, further work needs to be achieved in two main directions:

- For linking the events to temporal expressions, resorting to a better semantic processing seems to be the only way to go further the current results. Especially, the event classification (between states, processes, etc.), for verbs as well as for nouns, is an important challenge.
- For ordering all the events beyond sentence level, which was the purpose of task C in which no participant has been successful, a more general discourse analysis is needed. This aspect is much more ambitious.

Acknowledgement

We would like to thank the TempEval task organizers for their efforts in organizing this event and annotating the corpora.

References

James Allen, 1984. Toward a general theory of action and time. *Artificial Intelligence*, 23:123-154.

Salah Aït-Mokhtar, Jean-Pierre Chanod and Claude Roux. 2002. *Robustness beyond Shallowness: Incremental Deep Parsing*. Natural Language Engineering, 8 :121-144

Caroline Brun and Caroline Hagège, 2003. *Normalization and Paraphrasing using Symbolic Methods*, 2nd Workshop on Paraphrasing, In Proceedings of ACL 2003, Sapporo, Japan.

Caroline Hagège, Claude Roux. 2003. *Entre syntaxe et sémantique: Normalisation de l'analyse syntaxique en vue de l'amélioration de l'extraction d'information*. In Proceedings of TALN 2003, Batz-sur-Mer, France.

Inderjeet Mani, James Pustejovsky and Robert Gaizauskas (ed.) 2005. *The Language of Time A reader.* Oxford University Press..

Philippe Muller and Xavier Tannier 2004. *Annotating and measuring temporal relations in texts.* In Proceedings of the 20th International Conference on Computational Linguistics (COLING'04), Geneva, Switzerland, pages 50-56.

James Pustejovsky, Patrick Hanks, Roser Saurí, Andrew See, Robert Gaizauskas, Andrea Setzer and Beth Sundheim. 2003. The TIMEBANK Corpus. *Corpus Linguistics.* Lancaster, U.K.

Claude Roux. 2006. *Coupling a linguistic formalism and a script language.* CSLP-06, Coling-ACL, Sydney, Australia.

Roser Saurí, Jessica Littman, Bob Knippen, Robert Gaizauskas, Andrea Setzer and James Pustejovsky. *TimeML Annotation Guidelines.* 2006.

Frank Schilder, Christopher Habel and Yannick Versley, 2003. *Temporal information extraction and question answering: deriving answers for where-questions* .2nd CoLogNET-ElsNET Symposium, Amsterdam., 2003.

Marc Verhagen, Robert Gaizauskas, Frank Schilder, Mark Hepple, Graham Katz and James Pustejovsk. 2007. *SemEval-2007 Task 15: TempEval Temporal Relation Identification*. SemEval Workshop in ACL 2007, Prague, Czech Republic.