

# XTM: A Robust Temporal Text Processor

Caroline Hagege<sup>1</sup>, Xavier Tannier<sup>2</sup>

<sup>1</sup> Xerox Research Centre Europe, 6 Chemin de Maupertuis, 38240 Meylan ,France  
Caroline.Hagege@xrce.xerox.com

<sup>2</sup> LIMSI, 91403 Orsay, France  
Xavier.Tannier@limsi.fr

**Abstract.** We present in this paper the work that has been developed at [hidden name] to build a robust temporal text processor. The aim of this processor is to extract events described in texts and to link them, when possible, to a temporal anchor. Another goal is to be able to establish temporal ordering between the events expressed in texts. One of the originalities of this work is that the temporal processor is coupled with a syntactico-semantic analyzer. The temporal module takes then advantage of syntactic and semantic information extracted from text and at the same time, syntactic and semantic processing benefits from the temporal processing performed. As a result, analysis and management of temporal information is combined with other kinds of syntactic and semantic information, making possible a more refined text understanding processor that takes into account the temporal dimension.

## 1 Motivation

Although interest in temporal and aspectual phenomena is not new in NLP and AI, temporal processing of real texts is a topic that has been of growing interest in recent years (see [5]). The usefulness of temporal information has become clear for a wide range of applications like multi-document summarization, question/answering systems (see for instance [10]) and information extraction applications. For presenting search results, Google also offers now, in an experimental way, a timeline view to provide results of a search (see [www.google.com/experimental](http://www.google.com/experimental)). Temporal taggers and annotated resources such as TimeBank ([7]) have been developed. An evaluation campaign for temporal processing has also been organized recently (see [11]).

But still, it remains a challenge to associate automatically with a temporal anchor, all the events denoted in texts, and to be able to compute in many cases temporal relations holding between the different events.

Some reasons for this difficulty are:

- Temporal information is conveyed by a wide range of different sources (lexical semantic knowledge, grammatical aspect, morphological tenses) that have to be combined in order to resolve the temporal value.
- Extra-linguistic knowledge is necessary to process temporal ordering properly (e.g. in “*He opened the door and went out*”, world-knowledge tells us that *opening the door* occurred just before *going out* while in “*He ate and drank*” is an assertion of a general level and no temporal order can be stated here.
- Some reasoning is necessary (e.g. if an event occurred before another event which is simultaneous to a third event, then it is possible to state that the first event happened before the third one).

The work we perform concerning temporal processing of texts is part of a more general text understanding process. Temporal processing is integrated into a more general tool, XIP, which is a general purpose linguistic analyzer [2]. Temporal analysis is thus intertwined with syntactico-semantic text processing including deep syntactic analysis and determination of thematic roles [4].

In the first part of this paper, we present our temporal processor. Details on how we perform our three-level temporal processing are then given. Then, we present the results obtained by our system in the context of the TempEval campaign [11]. As a conclusion, we give some directions for future work.

## 2 XTM a Temporal Module for Robust Linguistic Processing

Our temporal processor, called XTM (for XIP Temporal Module), is an extension of XIP [2]. 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 general thematic roles between a predicate (verbal or nominal) and its arguments. For syntactic relations, long distance dependencies are taken into account and arguments of infinitive verbs are handled. See [3] for details on deep linguistic processing using XIP.

Temporal processing is first performed in parallel with incremental linguistic processing and then in an independent way for temporal inference and calculations. We

will first give a brief reminder of XIP and explain why it is an advantage to consider linguistic and temporal processing simultaneously.

## 2.1 XIP – A General Purpose Deep Syntactic Analyzer

XIP is rule-based and its architecture can be roughly 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 includes a Named Entity Recognition (NER) process.
- A deeper processing performs first a generic syntactic dependency analysis (detection of main syntactic relations as “subject”, “direct object”, “determination” etc.) and then, based on the result of this generic stage, a deeper analysis (some thematic roles, clause embedding, etc.)

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.

## 2.2 Intertwining Temporal Processing and Linguistic Processing

Temporal processing is integrated into XIP. We consider that temporal processing is one step 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 handling of these complex expressions avoids possible errors in general chunking and dependency computation. For instance, chunking a complex temporal expression like “*2 days before yesterday*” as a single unit in a sentence like “*They met 2 days before yesterday*” allows us to avoid having an erroneous adjunct *two days* attached to *met*.

We will detail in sections 3.2.1 and 3.2.2 how low-level (i.e. sentence level) temporal processing is combined with the rest of general purpose linguistic processing. But a temporal annotation that aims at ordering events appearing in text along a time line cannot be performed only at the sentence level. In section 3.2.3 we detail how we perform temporal processing at the level of the whole document, and how temporal calculations and inference are done.

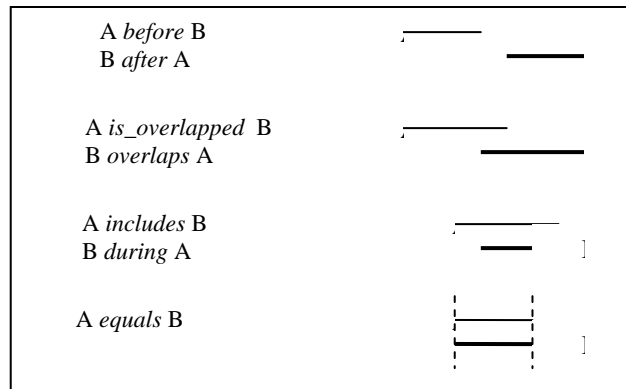
### 3 Details on Temporal Processing

Before entering into details on how temporal processing is handled in XTM, some preliminary definitions are necessary. More precisely, because one of the final goals is to be able to time stamp and to order chronologically events denoted in the text, we have to clarify what we consider as “temporal relations” and as “events”.

#### 3.1 Preliminary Definitions

##### Temporal Relations

The set of temporal relations we use is the following: AFTER, BEFORE, DURING, INCLUDES, OVERLAPS, IS\_OVERLAPPED and EQUALS (see Figure 1). They are defined as equivalent to or disjunctions of Allen’s 13 relations [1]. They are simpler than Allen’s relations, which makes sense in most fuzzy natural language situations, but they preserve the basic properties of Allen algebra, such as mutual exclusivity, exhaustivity, inverse relations and the possibility to compose relations. This choice is explained in more details in [6].



**Fig. 1.** Temporal relations used in XTM

## Events

Temporal expressions are attached to, and temporal ordering applies to, events. It is not straightforward to define what is an event. The question of how to consider stative verbs (temporally annotable or not), as well as deverbal nouns like *destruction* or *birth*, is a difficult one. In our approach, we decided to consider as events (that can be temporally anchored) the following linguistic elements:

- Any verb (expressing either an action or a state<sup>1</sup>)
- Any deverbal noun, when there is a clear morphological link between this noun and a verb (e.g. “interaction” is derived from the verb “interact”).
- Any noun which is not a deverbal noun and that can be either:
  - An argument of preposition *during* (e.g. *during the war*)
  - A subject of verbs *to last*, *to happen* or *to occur*, when these verbs are modified by an explicit temporal expression (e.g. *the siege lasted three days*).

We call this last class of nouns “time span nouns”. Examples of such nouns are words like *sunrise* or *war*, which intuitively correspond to nouns denoting events of certain duration. A list of these nouns (which may not be exhaustive) has been obtained by applying the above-mentioned heuristics to the Reuters corpora collection at NIST and by removing all deverbal nouns from the obtained list.

### 3.2 A Three-Level Temporal Analysis

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 (section 3.2.1)
- Attaching these expressions to the corresponding events they modify and ordering events appearing in the same sentence (section 3.2.2)
- Ordering events in the whole document. (section 3.3.3)

---

<sup>1</sup> Although stative verbs and action verbs have different semantic properties that may impact temporal inference as stated in [5].

### 3.2.1 Local Level

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

The first question that is raised concerns the definition of boundaries (tokenization) of complex temporal expressions. Should complex temporal expressions like *10 days ago yesterday*, or *during 10 days in September* be considered as a whole or should they be split into different tokens?

In the standard TimeML [9], signals (prepositions “in”, “during”, “after”, adverb “ago”, etc.) are not included in temporal expressions, so these kinds of tokens are generally split. But this is not our approach. Indeed, our aim is to produce temporal tokens that are semantically consistent, and that can be associated with a normalized representation.

We consider the following criteria, which are syntactically and semantically motivated:

A complex temporal expression has to be split into minimal temporal tokens if:

1. each minimal temporal token is syntactically valid when attached to the modified event
2. each combination event + minimal temporal expression must be logically implied by the combination event + complex temporal expression.

Here are some examples illustrating this definition:

*each week* in “*We met each week*”

The expression *each week* cannot be split into *each* and *week* as condition 1 is not satisfied (The expression *We met each* is not syntactically valid).

*twice each week* in “*We met twice each week*”

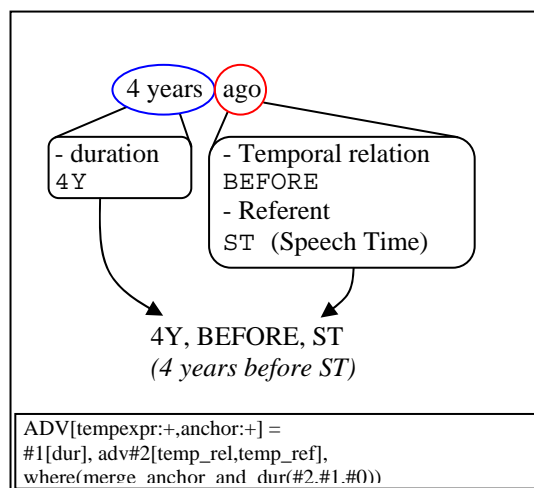
This expression could be split into two minimal expressions *twice* and *each week* according to condition 1. However, condition 2 is not satisfied as *we met twice* is not implied by *we met twice each week*. For this reason, this expression has to be considered as a whole.

*10 days in September* in “*We traveled 10 days in September*”

This expression has to be split into two minimal temporal tokens (*10 days* and *in September*). Both condition 1 and 2 are verified (*we traveled 10 days in September* implies both *we traveled 10 days* and *we traveled in September*).

Having defined these criteria for determining precisely what a minimal temporal token is, we perform recognition of temporal expressions 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. Some actions are associated with the contextual rewriting rules. These actions are meant to attribute a value to the resulting temporal expression (left hand side of the rule). Technically, these actions are calls to Python functions that can be executed directly from the parser [8].

Figure 2 illustrates this stage with an example rule for a simple anchor date. The rule builds an ADV (adverbial) node with associated Boolean features (on the left hand side 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” whose parameters are three linguistic nodes (#0 represents the resulting expression on the left hand side of the rule).



**Fig. 2.** Local level processing, anchor date

### 3.2.2 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 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 the same

sentence. Verbal tenses are also explicitly extracted at this stage by using morphological information coming from the pre-processing stage. Furthermore, at this stage, some underspecified normalization is performed at a local level.

#### ***Attaching temporal expressions to events***

As a XIP grammar is applied in an incremental way, in a first stage, any prepositional phrase (PP), including 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 sentence “*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 gives 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 verbs, what is the sequence of tenses), some intra-sentential temporal ordering of events 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 following two 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 relationships.

*ORDER[before](suspended, comes)*

*DELTA(suspended, comes, a month)*



They express that the event *suspended* is before the event *comes* with an interval of *a month* (analyzed as a duration whose value has been calculated at the local level, see section 3.1).

In the second example:

*After ten years of boom, they're talking about layoffs.*

*boom* is embedded in the *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 inflection 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 absence of sufficient context.

### **3-2-3 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 [6].

Dates which are relative to speech time can be calculated from the document creation time (DCT), 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- Evaluation

Our temporal processor has been evaluated in the context of the first evaluation campaign for temporal relation TempEval, which has been organized in 2007, within the scope of SemEval (Verhagen et al., 2007). The participants were proposed three tasks:

- Task A: identifying temporal relations holding between time and event expressions within the same sentence
- Task B: identifying temporal relations holding between event expressions and Document Creation Time (DCT)
- Task C: identifying temporal relations holding between main events of adjacent sentences.

For each task, events and temporal expression boundaries were provided to the participants together with information about tense and verbal aspects for the events. The conversion of linguistic temporal expressions into absolute dates was also provided. We chose not to use this information as we had our own event linking (integrated into the parser) and also our own processing for temporal expression normalization. We also decided to rely on our own morphological information about tense and aspect. In this way, we also indirectly evaluated the capability of our module to extract events and temporal expressions, to link them and to normalize them. We simply mapped our results to the TempEval framework afterwards. We participated in the three tasks and obtained the following results:

**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).

**Task C** 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 a default of OVERLAP for each unfound relation. The result was equal precision and recall of 0.58, which was the second best score.

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

Full results are given in the TempEval overview paper [11].

## 5- Conclusion and Further Work

We have developed a temporal processing module integrated within a more general tool for syntactic and semantic analysis. This module has been evaluated in the context of the TempEval initiative and we feel that the results are encouraging considering that we obtained good results and that we do not use all of the information that was provided for the competition. Furthermore, preliminary tests have shown that our system can also handle texts from any style and genres producing the same kind of results. One of the advantages of our approach is that temporal processing and syntactico-semantic processing can benefit from each other (linking temporal expressions and events is a special case of syntactic attachment and at the same time an early and correct chunking and characterization of temporal expression avoids errors in syntactic analysis (e.g. temporal noun phrases are generally neither subject nor direct object of a verbal predicate).

Another advantage of our incremental approach (three levels of processing) is that we can, according to different application needs, tune our module so that we can have a partial temporal processing (From a simple linking of events to full temporal inference).

However, many problems remain. Some of them are typical problems of temporal processing, others are more general but their solution should be beneficial for a proper temporal treatment:

- How to determine the temporal focus ? (i.e. the temporal reference changes according to the discourse).
- Anaphora between events is not detected by our system. If this were done, we would be able to use time anchor of one event to determine the time anchor of the co-referent event.

We hope to be able in the future to address some of these problems in order to have a more and more refined time processor able to take into account rich semantic information.

## References

1. Allen, J.: Toward a general theory of action and time. *Artificial Intelligence* 23 (1984) 123-154.
2. Ait-Mokhtar, S., Chanod, J.P., Roux, C.: Robustness beyond Shallowness: Incremental Deep Parsing. *Natural Language Engineering*, 8 (2002) 121-144
3. Brun, C., Hagège, C.: Normalization and Paraphrasing using Symbolic Methods, 2nd Workshop on Paraphrasing, In Proceedings of ACL 2003, Sapporo, Japan (2003)
4. Hagège, C., Roux, C.: 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 (2003)
5. Mani, I., Pustejovsky, J., Gaizauskas, R. (ed.): *The Language of Time A reader*. Oxford University Press (2005)

6. Muller, P., Tannier, X.: Annotating and measuring temporal relations in texts. In Proceedings of the 20th International Conference on Computational Linguistics (COLING'04), Geneva, Switzerland (2004) 50-56
7. Pustejovsky, J., Hanks, P., Saurí, R., See, A., Gaizauskas, R., Setzer, A., Sundheim, B.: The TIMEBANK Corpus. Corpus Linguistics. Lancaster, U.K (2003)
8. Roux, C.: Coupling a linguistic formalism and a script language. CSLP-06, Coling-ACL, Sydney, Australia (2006)
9. Saurí, R., Littman, J., Knippen, B., Gaizauskas, R., Setzer, A., Pustejovsky, J.: TimeML Annotation Guidelines (2006)
10. Schilder, F., Habel, C., Versley, Y.: Temporal information extraction and question answering: deriving answers for where-questions. 2nd CoLogNET-ElsNET Symposium, Amsterdam. (2003)
11. Verhagen, M., Gaizauskas, R., Schilder, F., Hepple, M., Katz, G., Pustejovsky, J.: SemEval-2007 – Task 15: TempEval Temporal Relation Identification. SemEval workshop in ACL (2007)