A Framework for Constructing Temporal Models from Texts*

Enrique Alfonseca§†
§Departamento de Ingeniería Informática,
Universidad Autónoma de Madrid
28049 Madrid, Spain
Enrique.Alfonseca@ii.uam.es

Suresh Manandhar†
†Computer Science Department,
University of York
YO10 5DD York, U.K.
suresh@cs.york.ac.uk

Abstract

We describe here the algorithm and criteria we have used for the identification of events in a text, and for finding the temporal relations between them. Some of our *ad hoc* heuristics provide very good results, with precision and recall values around 90%, that are able to provide us a fairly good ordering of events from different kinds of texts. There are still some difficult cases not covered by the heuristics, and we are working in annotating a bigger training set so we can learn rules for them.

Introduction

A story usually consists on one or more episodes, each one of which consists of a set of **events**, which may be **actions** or **states**. The text usually provides useful information about events, such as the agent or patient, the objects involved, and the time and place settings (Bell, 1998). Interestingly, episodes and events are not usually provided in chronological order. Brewer (1985) distinguishes between the order in which a story is told (the *discourse structure*) and the chronological ordering of the events (the *event structure*). If we want to understand the meaning of a text, it is important to be able to reconstruct the order in which events happened. Studies by linguists show that discourse structure and event structure are usually different, and this applies to different domains. As as example, Bell (1998) provides several single-sentence newswire stories each of which contains up to five events with many different reorderings.

In order to understand a text it is important to be able to reconstruct its *event structure*. An automatic procedure for doing this has potentially many useful applications, such as multi-document summarisation or question answering systems (Mani and Wilson, 2000). We describe here a framework that extracts the temporal order of events from a text. Our algorithm introduces several changes with respect to previous approaches (Filatova and Hovy, 2001), such as the use of the WordNet ontology (Miller, 1995) for the identification of events; and the relative temporal ordering of the events in the text even when the date of a story is unknown.

*This work has been partially sponsored by CICYT, project number TIC2001-0685-C02-01.

Related work

In the Sixth and the Seventh Message Understanding Conference (MUC6, 1995) (MUC7, 1998), the task called *Scenario Template* included identifying time expressions, and assigning a calendar time only to the scenario event types (joint ventures in MUC-6, and rocket launchings in MUC-7). However, the scores were low. SRA, which scored best at that task, found only around 35% of the start and end times of events; and more than half the answers it provided for this task were incorrect.

Mani and Wilson (2000) extended the interpretation of time expressions in MUC to include expressions that are relative to the time in which the document was written, such as *today* or *three days ago*. Filatova and Hovy (2001) describe a method that looks for events in newswire articles, and assigns calendar times to them whenever possible.

Framework

We aim at finding events in texts, together with temporal expressions. Times have to be placed in the right places in the timeline, and they have to be attached to the events they modify. During initialisation, the system creates a time line where any absolute time point or time interval can be placed. The steps are:

1. Preprocessing of the texts: chunk parsing, word-sense disambiguation and identification of narrative events (section).
2. Identification of temporal expressions in the text, and resolution in the timeline (section 2).
3. Find temporal relationships between the events (section 4).

Detection of events in a text

As far as we know, most systems that attempt to resolve temporal information from texts do not extract events (Mani and Wilson, 2000) (Koen and Bender, 2000); or they only consider events denoted by verbs heading syntactic clauses (Filatova and Hovy, 2001). We have extended the definition of event to include:

- Complements of predicative verbs such as *to be* or *to become*, which usually represent states, e.g. (1a).

Corpus	Events (human)	Events (system)	Correct	Recall	Precision
MUC-3	66	65	59	89.39%	90.77%
LOTR	67	66	59	88.06%	89.39%
WSJ	73	77	67	91.78%	87.01%
IBM	78	73	68	87.17%	93.15%
Total	284	281	253	**89.08%**	**90.03%**

Table 1: Results (precision and recall) for the identification of **events** in the sample texts.

- Verbs with lexical information, i.e. excluding auxiliary, modal and predicative verbs; these verbs represent actions or states, e.g. (1b).
- Nouns that, in the WordNet taxonomy, are hyponyms (specifications) of one of the three concepts <*act*>, e.g. *arrival* in (1c); <*event*>, e.g. *accident* in (1c); or <*state*>, e.g. *danger* in (1d).

(1) a. He was **strong** at the time.
b. The train **arrived** on time.
c. The **accident** was **known** before his **arrival**.
d. The general was in **danger**.

Because some nouns may have some senses that represent events and some that don't, it is necessary to use a word sense disambiguation procedure that decides with which sense a noun is used in a given context. In our preliminary experiments we have simply used a baseline procedure that assigns, to every noun, the sense with which it is more frequent in the SEMCOR sense-tagged corpus (Landes et al., 1998), but accuracy could be increased with a better algorithm.

Results detecting events

We have applied those criteria to four small texts from different domains: one newswire article from the MUC-3 training data, about terrorism; the first sentences from the Wall Street Journal (WSJ) corpus; and sentences from a corpus with IBM program instructions, all of them obtained from the Penn Treebank version II (Marcus et al., 1993); and the first sentences from the first chapter of *The Lord of the Rings* (LOTR) (Tolkien, 1968). These texts were processed using the TnT part-of-speech tagger (Brants, 2000) and our own cascade of chunk parsers based on transformation lists (see (Ramshaw and Marcus, 1995)). The resulting events were compared to the annotation produced by a human.

The metrics used are for measuring its accuracy are **recall**, the fraction of correct events that were found in the test corpus; and **precision**, the fraction of the events proposed by the system that were correct.

The overall results detecting events in the text are shown in Table 1. As a comparison, we can cite that Filatova and Hovy (2001) obtained a recall of 60.76% and a precision of 55.81%. However, we are not using the same corpora and we do not know which were their annotation guidelines, so we do not think that our respective results are comparable.

The program attains similar results in the first three documents, most of the errors being due to part-of-speech tagging and parser mistakes. The recall of the IBM manual is lower because most of the verbs in the section titles appeared capitalised, and the part-of-speech tagger mistagged them all as proper nouns.

Finding information about events

Our chunk parser, when possible, extracts subject-verb and verb-object relationships from the texts. With this information, we could find the agents of roughly one third of the total events identified in the texts. To find the agent of a nominal event we also looked at genitive constructions, such as that in sentence (2).

(2) I waited for Peter's arrival

Anchoring events in time

When a writer narrates a story, the events that he or she is telling can be classified in three groups: past events, present events and future events. Every event in the first group will be told in past tense, every current event will be told in present tense, and every future event will be told in future tense. Occasionally, some verbal times can be used in other forms, such as those in sentences (3a) and (3b). Usually, when this phenomenon happens, there are explicit time expressions (e.g. *in 1492* and *tomorrow*) that show that the tense of the verb is not being used in the usual way.

(3) a. After many days navigating, Colon finally **discovers** America in 1492.
b. Tomorrow I **go** to see my parents.

This can be further complicated by the fact that we can embed narrations inside narrations, for instance, when a person in a novel speaks. In that case, the verb tenses in the person's utterances are relative to the time at which that person speaks. We have called **anchor_time** the time at which the narration happens, and **anchor_rules** the criteria used for finding temporal relationships from a verb and the anchor_time using its tense. If, while processing a text, we find an event of *speaking* that introduces a new narrative context, we use the time of that event as the new anchor_time for that context.

The **anchor_rules** are necessary because, when the embedded narration is not quoted literally, but stated as a subordinate clause introduced by the complementiser *that*, then present tenses and future tenses are transformed into past and conditional tenses, so we can recognise which verb tenses refer to the time when the utterance was produced and which refer to now. Table 2 shows when the event of *going* is interpreted to happen for each of the sentences in (4).

Sentence	*go* happens after...
(4a)	the event of saying
(4b)	the event of telling
(4c)	the event of saying
(4d)	now
(4e)	the event of saying

Table 2: Temporal relation between *going* and other events in the example sentences in (4).

(4) a. Peter told me yesterday, 'John said that he would go'.
b. Peter told me yesterday, 'John said that he will go'.
c. Peter told me yesterday that John said 'I will go'.
d. Peter told me yesterday that John said that he will go.
e. Peter told me yesterday that John said that he would go.

The algorithm is displayed in Figure 1. Initially, the document has to be placed in the timeline, at the initial anchor point. If the document contains a header specifying its date, then we can place it at a fixed point in the timeline; otherwise, we just assign to it an unknown time point. Every event that is described in past-tense will be placed before the document's time. And, inversely, future-tense verbs will describe events that happen after the writing of the document. The anchor_time will also help in locating in time expressions such as *today* or *three months ago*.

Next, time expressions are identified in the document, and the events they modify are placed accordingly in the time line. After that, prepositions and conjunctions introducing event clauses are used to find the ordering of the remaining events. Finally, if there is any event of narrating, it is processed, but the anchor point for the events inside the narration will be the time when the narration occurred.

Identifying time expressions

Writers place events in time using **Time-denoting expressions** (Móia, 2001), such as `three days ago`. These expressions refer to intervals in time. We have compiled a list of regular expressions in a FLEX file to recognise absolute (e.g. dates) and relative (e.g. *today*) time stamps. This expressions can be easily used to place the events modified by them in the timeline. In our experiments, a total of 22 expressions were found, giving a total recall of 87.5%, and a precision of 95.45%.

After *time-denoting expressions* have been found, we use prepositions and conjunctions to translate them into absolute or relative time-intervals. Some time expressions, such as *today* or *tomorrow*, or time prepositions such as *ago* are relative to the anchor time; while other time expressions such as *before* or *after* are relative to the time of the event in the main clause. All this is taken into account.

In our sample texts, the simple heuristic of assigning the time to the nearest event in the same syntactic clause was 100% correct, for the 21 expressions found in the text.

Finally, Table 3 displays the number of temporal relationships that were discovered using the anchor rules. The first column shows the total number of relationships discovered,

Corpus	Found	Expanded
MUC-3	61	911
LOTR	51	183
WSJ	95	1226
IBM	32	1083

Table 3: Results for finding temporal relations between events, using the verb tenses and the temporal expressions.

and the second column shows the number inferences from the previous, by applying the symmetry of the *simultaneous* relationship and the transitivity of the *preceding* relationship:

$$t_1\ Simultaneous\ t_2 \Leftrightarrow t_2\ Simultaneous\ t_1$$
$$(t_1 < t_2) \wedge (t_2 < t_3) \Rightarrow (t_1 < t_3)$$

In the LOTR corpus there were few inferences, because most of the verbs were in past tense (preceding the anchor time) and there were few temporal expressions with which place the events in the timeline.

Reordering the events

Consecutive sentences often convey chronological events. When that is not the case, prepositions and conjunctions provide the information to correctly interpret the sentences. The following are several examples of ways to express temporal relationships, such as preceding events, in (5a); consecutive events, in (5b); simultaneous, in (5c); subevent relationships, in (5d); or overlapping events, as in (5e).

(5) a. Unfortunately, I *left* before your *arrival*.
b. Several witnesses *cried* after the *accident*.
c. While I *was waiting* for you, I *read* the whole magazine.
d. While he *was driving* home, he *passed* a red light.
e. While I *was parking* my car, it started *raining*.

Although the temporal relationships found already provide us with a good overview of the order of the events in the texts, we can still find many events whose position in the timeline is not known, but we think that temporal prepositions and conjunctions, together with semantic information from the verbs could be useful to find it. However, preliminary experiments using a Maximum Entropy model to find these remaining relations suggest that our training set is too small to hold a representative number of the many ways that language provides to express temporal events, so we probably need to annotate a bigger amount of data or use an *ad hoc* solution.

Conclusions and Future Work

We describe here a procedure for identifying and resolving time expressions in texts, and for reconstructing the chronological ordering of events from a text. Our algorithms seem to be general enough to be used with documents from different domains, such as political news, financial news, novels and user manuals.

The method we have built for identifying events has two advantages over previous approaches: WordNet is used to

Process(document)
1. Assign the document's date to a point in the timeline (the anchor_time).
2. anchor_rules ← relations between each verb tense and the anchor_time.
2. Find_tenses(anchor_time, anchor_rules, document)

Find_tenses(anchor_time, anchor_rules, document)
1. Find temporal expressions in the text.
2. Use these temporal expressions to place a few events in time.
    2.1. If they are absolute expressions, place the events in the timeline.
    2.2. If they are relative to the anchor_time, place them accordingly.
3. Use the anchor_rules to find relations between the verbal events and the anchor_time.
4. Find temporal relations between events, using features such as prepositions (*before*, *after*, etc.).
5. For every *narration event*,
    5.1. fragment ← portion of the document with the narration.
    5.2. new_anchor_time ← the time when the narration event takes place.
    5.3. Create the new_anchor_rules for the fragment:
    5.4. If the narration is a literal citation, tenses are relative to new_anchor_time.
    5.5. Else,
        5.5.1. Past and conditional tenses are relative to new_anchor_time, as present and future.
        5.5.2. Present and future tenses refer to the current anchor_time.
    5.6. Find_tenses(new_anchor_time, new_anchor_rules, fragment)

Figure 1: Pseudocode of the algorithm for finding temporal relations between events.

identify common nouns that can represent events, and we allow different events to be in the same clause, even if they happened at different times. Therefore, we do not need a robust parser for our experiments.

Next, we describe a new algorithm for resolving the time expressions, regardless of whether they are absolute calendar dates, relative to the time when the text was written, or relative to the event in the main clause. The advantage of our algorithm is that it deals easily with embedded narrations, by updating a set of *anchor rules* that keep track of the semantic meaning of verb tenses in each case. This is specially useful for processing newswire articles and novels, because both include speeches quite often.

Finally, the extension of the events with information about the agents and objects of the action events is useful for answering questions such as (6).

(6) a. *How many times did Frodo visit Lorien?*

b. *How often has the current president of Portugal travelled abroad?*

The following are a few open lines and improvements that we may be addressing in the future:

- Detecting event's coreference, so two references to the same event are merged in one.
- Improve modules such as the word-sense disambiguator or the parser.
- Consider people's ages as time expressions.

Because many of the systems to find temporal expressions in the MUC conferences attained high results using regular expressions, we have tried the same approach, with very good results. However, some of these expressions have still to be disambiguated. Mani and Wilson (2000) used a decision tree classifier for distinguishing senses such as the two of the word *today*: the general (similar to *nowadays*) and the specific one. The same approach could be followed to decide whether a present-tense verb refers to a universal truth or to an event happening right now.

References

A. Bell. *The Discourse Structure of News Stories*, chapter 3, pages 64–104. Allan Bell and Peter Garret, eds., Blackwell Publishers, 1998.

T. Brants. *TnT - A Statistical Part-of-Speech Tagger*. User manual, 2000.

W. F. Brewer. *The story schema: universal and culture-specific properties*, pages 167–94. Cambridge University Press, 1985.

E. Filatova and E. Hovy. Assigning time-stamps to event-clauses. In *Workshop on Temporal and Spatial Information Processing, ACL-2001*, Toulouse, France, 2001.

D. B. Koen and W. Bender. Time frames: Temporal augmentation of the news. *IBM Systems Journal*, 39(3/4), 2000.

S. Landes, C. Leacock, and R. Tengi. *Building Semantic Concordances. In (C. Fellbaum (Ed.) WordNet: An Electronic Lexical Database*. MIT Press, 1998.

I. Mani and G. Wilson. Robust tempral processing of news. In *38th Anual Meeting of the ACL (ACL'2000)*, Hong Kong, 2000.

M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz. Building a large annotated corpus of english: the penn treebank. *Computational Linguistics*, 19(2):313–330, 1993.

G. A. Miller. WordNet: A lexical database for English. *Communications of the ACM*, 38(11):39–41, 1995.

T. Móia. Telling apart temporal locating adverbials and time-denoting expressions. In *Workshop on Temporal and Spatial Information Processing, ACL-2001*, Toulouse, France, 2001.

MUC6. *Proceedings of the 6^{th} Message Understanding Conference (MUC-6)*. Morgan Kaufman, 1995.

MUC7. *Proceedings of the 7^{th} Message Understanding Conference (MUC-7)*. Morgan Kaufman, 1998.

L. A. Ramshaw and M. P. Marcus. Text chunking using transformation-based learning. In *Third ACL Workshop on Very Large Corpora*, pages 82–94. Kluwer, 1995.

J. R. R. Tolkien. *The Lord of the Rings*. Allen and Unwin, 1968.