A Framework for Temporal Resolution

Georgiana Puşcaşu

School of Humanities, Languages and Social Sciences
University of Wolverhampton
United Kingdom
E-mail: georgie@wlv.ac.uk

Abstract

The ability to identify and analyse temporal information is important for a variety of natural language applications, such as information extraction, question answering and multi-document summarisation. This paper discusses the automatic annotation of temporal information in newswire texts. The temporal tagger adheres to a framework initially designed for anaphora resolution and recently adapted for temporal resolution. It achieves an accuracy of 83.66% in identifying and labelling temporal expressions.

1. Introduction

Access to the temporal information conveyed in a text can improve the performance of many NLP applications. This paper discusses the automatic annotation of temporal expressions (TEs) in newswire texts. This recent research is being pursued with the aim of automatically identifying temporal information for subsequent use in a Question Answering system. The fact that approximately 20% of the TREC-11 main task test questions have had a temporal expression as an answer proves the relevance of this research, especially as this percentage does not include questions which required a time scale ordering of the events described in the documents, questions where the answer was the duration of intervals, or questions which contained TEs as clues for the expected answers. In order to discover the benefits that temporal annotation would bring to the QA task, a temporal annotation scheme and a temporal tagger are needed.

We adopted the TIDES annotation scheme presented in (Ferro et al., 2001), which is compatible with the time ontology defined by Stanford Knowledge Systems Lab (KSL-Time, 1999) and also with the DAML-Time Ontology (Hobbs, 2002). Therefore it can support advanced inferential capabilities based on information extraction from text.

Using this annotation scheme, we developed a temporal tagger able to identify both self-contained TEs, which are tagged with their value, and indexical TEs, which, depending on their semantics, receive a value based on a temporal function having as argument the time that they are relative to, which we will call its *anchor*. In the process of building an automatic temporal tagger, we adopted a three-layered architecture capable of accommodating different models for temporal resolution, a model corresponding to a specific approach to solving this problem.

2. Markable temporal expressions

Before going into details regarding the annotation process, a description of what is considered to be a temporal expression (TE) is needed. A TE is usually signalled by one or more reserved time words, called lexical triggers. The main terms that indicate a time expression are:

* nouns: *day, month, year, minute, second, weekend, midnight, century, millennium, era, semester, future, past*
* proper names: *Sunday, April, Christmas, Valentine's Day*
* specialized time patterns: *9:00, 26/12/2002, '80s*
* adjectives: *past, current, future, next, medieval, monthly*
* adverbs: *currently, next, then, weekly*
* time nouns/adverbs: *today, yesterday, now*
* numbers: *4th* (as in *He arrived on the 4th.*)

Words like *when, eventually, lately* or *frequent* are not considered triggers. Many time words and many time phrases containing time words can be used in an idiomatic sense (”*the last minute*”, ”*the eleventh hour*”, ”the order of *the day*”). The temporal expressions in such idioms are to be tagged, but are not to be assigned a value. Names of holidays or other occasions that are celebrated in different cultures are tagged, but for the time being they are not assigned a value, as this value usually comes from cultural and world knowledge.

The full extent of a TE should either be a noun, adjective, adverb or corresponding phrases (noun, adjectival or adverbial phrases). The temporal expression cannot be a prepositional phrase or a clause, so it cannot start with a preposition or a subordinating conjunction (*after **Friday**, before the meeting on **Monday*** are disallowed as temporal expressions). Premodifiers of temporal expressions such as determiners and postmodifiers such as prepositional phrases or subordinate clauses should be included in the time expression. The appositives that may appear after a TE are not to be included in the expression's tag, but, if they contain trigger words, they are to be tagged separately.

In the case of temporal range expressions (*from **1990** to **1999***), and conjunction (***today** and **tomorrow morning***) or disjunction (***six months** or **a year from now***) of time expressions, the points are tagged separately, even if they share modifiers.

More than one lexical trigger appear in some cases within the same TE, so in contexts where more indicators are present, the number and full extent of the corresponding TEs are determined using the following rules:

* **one TE** is created if there are no intervening tokens between temporal terms that qualify a unit of time

(<*twelve o'clock midnight*>, <*November 1999*>), if the terms are connected with the preposition *of* (<*the evening of December, 31*>) or if the prepositions *to, till, after, in* are used in expressing a certain point of time in a day. In these cases, but also in the case of ”month day, year” format, the expression containing all the terms will be tagged as a single unit.

* **multiple TEs with embedding** appear in two cases. One is when the larger TE denotes an offset to another TE included in it. In this case two tags are created with the one corresponding to the anchoring phrase contained within the extent of the tag of the complete phrase (<*two weeks from* <*next Tuesday*>>). The second case is characterized by the larger TE as a possessive construction. If both the possessive phrase and the phrase that it modifies are time-denoting expressions, then two tags are created, and the possessive phrase tag is contained within the extent of the complete phrase tag (<<*This year*>*'s spring*>).
* **multiple TEs without embedding** are created in cases other than those described above, meaning that temporal phrases appearing in close proximity (like appositive phrases, range expressions, and conjoined expressions) are tagged as independent phrases. Although tagged independently in terms of the extent, there is a dependency in terms of the value. The expression with finer granularity inherits the value of the coarser-grained expression. This inheritance happens regardless of the relative ordering of the two expressions (<*8.00 pm*> *on* <*Friday*>).

3. Description of the annotation scheme

The tag element used to represent time expressions is TIMEX2 and its attributes are the same as those described in the TIDES annotation guidelines (Ferro et al., 2001): VAL, MOD, SET, GRANULARITY, PERIODICITY and NON_SPECIFIC. The timeline used by the annotation scheme is derived from the ISO 8601 standard (ISO 8601, 1998) format for representing time values. The tag attributes are presented below together with their use.

The VAL attribute is used for any expression that indicates a point or interval on a calendar/clock or that can be identified as an unanchored duration. The placeholder character, ”X”, is used when parts of the value are unknown. VAL can have the following formats for values:

POINTS IN TIME	
VAL= ”YYYY-MM-DDThh:mm:ss” <TIMEX2 VAL=”2004-02-23T15:00”>3 p.m. Monday</TIMEX2>	Anchored expressions T = ISO time-of-day designator
VAL=”YYYY-WOY-D” <TIMEX2 VAL=”2004-W10”>next week</TIMEX2>	Week-based format
VAL=”token” <TIMEX2 VAL=”PRESENT_REF”>now</TIMEX2>	Tokens that replace the entire value of VAL
VAL=”YYYY-*token*” <TIMEX2 VAL=”2003-FA”>Fall 2003</TIMEX2> **VAL=”YYYY-MM-DDT*token*”** <TIMEX2 VAL=”2004-02-24TMO”>Tuesday morning</TIMEX2> **VAL=”WOY-*token*”** <TIMEX2 VAL=”W09-WE”>this weekend</TIMEX2>	Tokens that replace particular positions in the value of VAL
DURATIONS	
VAL=”PnYnMnDTnHnMnS” <TIMEX2 VAL=”P1H”>one hour long</TIMEX2> **VAL=”PnW”** <TIMEX2 VAL=”P3W”>three weeks</TIMEX2>	Expressions answering the question *how long*

Table 1: Possible formats for the value of VAL

There are certain tokens relevant in the representation of time points and durations that can occupy the whole value of VAL and tokens covering only parts of the value.

TOKENS COVERING THE WHOLE VALUE OF *VAL*		
Token	**Markable expressions**	**Non-markable expressions**
PAST_REF	past yesterday former lately long ago medieval	before previously earlier beforehand once
PRESENT_REF	now today current, currently present, presently nowadays (at) this (point in) time (at) the present time (at) the present moment	immediately instantly forthwith
FUTURE_REF	future tomorrow	ahead after soon, sooner shortly later eventually subsequent
TOKENS OCCUPYING ONLY ONE POSITION IN *VAL*		
Token	**Expressions**	**Position**
MO MI AF DT EV NI	morning midday afternoon daytime or working hours evening night	Hour
WE	weekend	Day
SP SU FA WI Q*n* H1 H2	spring summer fall, autumn, fall term/semester winter *n*-th quarter (n = 1..4) first half (of year) second half (of year)	Month

Table 2: Tokens that may appear in the value of VAL

The MOD attribute is used together with other attributes when the time expression includes a modifier that changes or clarifies the interpretation of VAL in some way. MOD captures the semantics of quantifier modifiers (*approximately, no more than*) and lexicalized aspect markers (*early, start of*), but not the semantics of prepositions or other terms outside the temporal expression. The tokens representing possible values for MOD, together with expressions that trigger them are presented below.

Type of expressions	**Token**	**Expressions**
POINTS IN TIME	BEFORE AFTER ON_OR_BEFORE ON_OR_AFTER	more than ... ago less than ... ago no less than ... ago no more than ... ago
DURATIONS	LESS_THAN MORE_THAN EQUAL_OR_LESS EQUAL_OR_MORE	less than ... (long), nearly more than ... (long) no more than at least
POINTS AND DURATIONS	START MID END APPROX	early, dawn, start, beginning middle, mid- end, late about, around, approximately

Table 3: Tokens that may appear in the value of MOD

The SET attribute is used in the representation of expressions denoting sets of time, i.e., times that recur regularly or irregularly (*every Tuesday, numerous weeks, some Thursdays*) and its only value is YES.

The GRANULARITY attribute is an explicit representation for how precise a TE is (the granularity for *Monday* is at day-level, while for *two years ago* is at year-level). It receives an explicit value only when the TE denotes a set of recurring times or has an indefinite value within a known set. The value is a string having on the first position the letter *G*, followed by a number or the placeholder *X* representing the number of time units, and having in the last position a character or token indicating the granularity unit (*W*-week, *NI*-night etc.).

The PERIODICITY attribute applies only to regular sets of times denoted by expressions including words like *always*, *every* or *each* or by expressions explicitly indicating a repetition (*daily*, *monthly*, *yearly*). The value of the attribute is a string containing the letter *F* (frequency) on the first position, number or *X* on the second position and a letter or token on the last (e.g. <TIMEX2 SET="YES" PERIODICITY="F1M" GRANULARITY="G1M">every month</TIMEX2>).

The NON_SPECIFIC attribute is used to represent TEs where no specific time is indicated. These expressions include NPs used generically (*She likes **December**.*), indefinites (***a Thursday** in May*) and other non-specific uses of NPs. The only value of this attribute is YES.

4. Temporal resolution

Temporal resolution is the whole process carried out in order to identify the values of the attributes attached to every temporal expression. This process starts by evaluating the expression either to a value or to a temporal function and, in the case of underspecified expressions, continues by finding the anchor that will, in the end, fill the unknown slots of the value. The anchor is an already resolved temporal expression. In the sentence *In **1999** he started his work in **August**.* the temporal expression *1999* is the anchor for *August* and will fill in the year slot in its final value.

4.1. The architecture

The architecture used for temporal resolution is three layered, analogous with the one presented in (Cristea and Dima, 2001) for anaphora resolution. Temporal references need to be resolved for many time expressions, a process which is similar to anaphora resolution, thus justifying the choice of such an architecture. This three-layered representation includes the text layer, the projection layer and the semantic layer. The text layer contains the temporal expressions from the surface text. On the projection layer, the temporal expressions from the text layer are projected together with temporary feature structures characterizing them. The semantic layer contains the fully resolved temporal expressions represented as feature structures which, beside the attribute-value pairs from the projection layer, include the final value and the ID of the anchor used to compute this value. Using these additional attributes it is possible to follow the succession of computations employed in solving a certain temporal expression, information that can be exploited in solving subsequent TEs. In the following example, the year that is attached to *December* is 1997, due to the tense of the verb *learned*. Then *Aug. 30* would incorrectly be assigned the same year as the preceding TE containing a year (1997), if the information saying that the year for *December* was computed by reference to another TE would not be available. But this information helps in getting the correct year from the fully-specified TE (*1998-06-01*).

(Document creation time: **1998-06-01**) *But in **December** she learned that the district had decided to start school on **Aug. 30**, before her family was to return.*

4.2. Model for temporal resolution

The architecture described above is capable of accommodating different models for solving time references. A model is seen as a quadruple: a set of primary attributes characterising the TEs, a set of knowledge sources (capable of filling in values for the primary attributes), a set of heuristics/rules (that co-operate in order to project the current investigated TE onto the semantic layer as a resolved TE or in order to find its anchor) and a set of heuristics/rules able to filter and order the list of already resolved TEs from which the anchor will be extracted.

Component 1 consists of a set of primary attributes that fill the projection layer and the semantic layer. These attributes will all contribute to the resolution of the final value of the TE they are associated with. The attributes currently employed by our temporal tagger are:

* *ID* = unique reference number attached to every TE.
* *tempVal* = the value contained in the expression itself.
* *finalVal* = the final value computed after the anchor is found.
* *temporalFunction* = true for underspecified TEs and false for fully specified ones (it indicates whether or not further resolution is needed).
* *valueFromFunction* = the function that will give the final value of this TE. This function receives the anchor as its argument.
* *anchorTimeID* = the ID of the resolved TE that serves as anchor for the current TE.
* *mod, granularity, set, periodicity* = corresponding to the TIMEX2 tag attributes described in section 3.
* *specificity* = true if the TE has temporal usage, false if it has a generic or metaphoric one.
* *verbTense* = the tense of the neighbouring verb.

Component 2 includes a set of knowledge sources that fill in values of the primary attributes during text processing. At least three knowledge sources should be included in every model of temporal resolution: a part of speech tagger, a shallow parser able to recognise the full extent of the temporal expression and a module which extracts the value or the temporal function associated with a temporal expression.

These three modules are employed by our temporal tagger. Within our tagger, the shallow parser that identifies the full extent of the TE is a rule-based module. The module that extracts the value or the temporal function associated with a TE employs hand-written rules to fill in the values of the attributes: *tempVal, temporalFunction, valueFromFunction, mod, granularity, set and periodicity*. In the case of fully specified TEs the value of the attribute *tempVal* is filled starting from the year level and down to the level of granularity of that time expression (for example *Monday, 1st of March, 2004* has the granularity at day scale, while *Summer of '69* or *January 2005* have month-level granularity). For these expressions the attribute *temporalFunction* will be false, as there will be no resolution needed to solve their final value (*finalVal = tempVal*) and the values for *mod, set and periodicity* will be established on the basis of the signal words or expressions that trigger a certain value (as described in section 3). For underspecified TEs (either expressions containing a value, where the value is not known up to the maximum level of granularity that is the year level, or expressions conveying a temporal function relative to a certain point in time) the attribute *temporalFunction* is true meaning that further resolution will be needed in order to obtain the full value. The *valueFromFunction* is filled only for the second type of underspecified TEs with the corresponding function or combination of functions.

Other useful knowledge sources which we plan to develop as resources become available are: a disambiguator for expressions which can have both temporal and generic or metaphoric usage (*the **fall** of the empire*, ***today**'s youth*), as described in (Mani and Wilson, 2000), and a module capable of identifying events, their anchoring in time, and the temporal relations that hold between them in order to be able to identify values of event-dependent TEs.

Component 3 is a set of rules or heuristics that decide if a TE is ready to go from the projection layer to the semantic layer and, if it is not ready, which expression already on the semantic layer is its anchor (the anchor is the expression that will fill the unknown parts of the final value, it is also the argument of the temporal function).

A TE is ready to go to the semantic layer if it is fully resolved or if there is no possible anchor in the previous text (this is not applicable to newswire texts, as a possible anchor can always be the document creation time). If there are possible anchors in the previous text, but their scores are lower than a certain threshold, then the TE under resolution remains on the projection layer until further text is processed, and if a certain quantity of text is processed and no better anchor is found, then the rules decide whether the anchor with the highest score among all the possible ones is chosen or whether the value is not to be specified.

The anchor of a TE is chosen from the already resolved TEs that have a value in the unknown slots. If more than one resolved TE can fill a specific slot, the relative position to the current TE, their type (whether they were fully specified or underspecified) and the function that computed their value (for underspecified ones) are used in choosing the anchor.

Component 4 contains a set of rules which establish the TEs that are able to be anchors and order them according to the probability of their being referred in the subsequent text. This component is not implemented at present in our temporal tagger, but we are planning to implement it when we have a deeper understanding of how transitions in temporal focus can supply information on the probability of a TE to be referred to in the future. It is obvious that at a certain position in a text, there can be cases where previously mentioned TEs are closed from being anchors for further TEs, but further research is needed to see what triggers this closure.

5. Evaluation and conclusions

The temporal tagger embodying the previously described architecture and containing only the indispensable modules in the first three components of the model has been evaluated on a set of 20 news articles from the AQUAINT Corpus of English News Text (the set of documents involved in the TREC-11 track). The tagger only receives as input the content extracted from the <DATE_TIME>, <HEADLINE> and <TEXT> tags. The text is processed sequentially, adding the TEs identified by the TE-extent identification module in Component 2 onto the text layer. Then on the projection layer, for every TE from the text layer, the value-extracting module (also from Component 2) fills the values of most of the attributes in Component 1. Then Component 3 decides which TEs on the projection layer are ready to go to the semantic layer and, for those not ready, it returns the anchor that will give their unknown slots values. This anchor is chosen from the set of resolved TEs from the semantic layer, configured by Component 4 which eliminates from the set the TEs closed from being referred to (in our tagger this component is not yet implemented, so this set contains all the TEs on the semantic layer).

The temporal tagger described in this paper identifies the TEs extent with an accuracy (F-measure) of 93.98% and the values of the attached attributes with an accuracy of 83.66%.

The errors made by the TE-extent identifying module are under recognition (some TEs were missed out), over recognition (some TEs were annotated in addition) and wrong extent (the identified text span was either shorter or longer than it should be).

The errors made in attaching values to the identified TEs are due to choosing an incorrect anchor and also due to the fact that no disambiguator was used to separate specific and generic usages of certain time words.

This framework is capable of accommodating events, which can also be placed on the semantic layer and their relations to other events or TEs can be solved with additional event specific modules.

6. References

Cristea, Dan and Gabriela Dima, 2001. An integrating framework for anaphora resolution. In *Information Science and Technology*, volume 4. Bucharest: Romanian Academy Publishing House.

Ferro, Lisa, Inderjeet Mani, Beth Sundheim, and George Wilson, 2001. Tides temporal annotation guidelines. Technical Report MTR 01W0000041, The MITRE Corporation.

Hobbs, Jerry, 2002. A DAML Ontology of Time. http://www.cs.rochester.edu/ ferguson/daml/daml-time-nov2002.txt.

ISO 8601, 1998. ftp://ftp.qsl.net/pub/g1smd/8601v04.pdf.

KSL-Time, 1999. http://www.ksl.Stanford.edu/ontologies/ time/1999.

Mani, Inderjeet and George Wilson, 2000. Robust temporal processing of news. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL'2000)*. Hong Kong.