The DANTE Temporal Expression Tagger

Paweł Mazur[*]

[*]Institute of Applied Informatics
Wrocław University of Technology
Wyb. Wyspiańskiego 27,
50-370 Wrocław, Poland
Pawel.Mazur@pwr.wroc.pl

Robert Dale[†]

[*,†]Centre for Language Technology
Macquarie University,
NSW 2109, Sydney, Australia
[†]rdale@ics.mq.edu.au
[*]mpawel@ics.mq.edu.au

Abstract

In this paper we present the DANTE system, a tagger for temporal expressions in English documents. DANTE performs both recognition and normalization of these expressions in accordance with the TIMEX2 annotation standard. The system is built on modular principles, with a clear separation between the recognition and normalisation components. The interface between these components is based on our novel approach to representing the local semantics of temporal expressions. DANTE has been developed in two phases: first on the basis of the TIMEX2 guidelines only, and then on the ACE 2005 development data. The system has been evaluated on the ACE 2005 and ACE 2007 data. Although this is still work in progress, we already achieve highly satisfactory results, both for the recognition of temporal expressions and their interpretation (normalisation).

1. Introduction

The task of temporal expression recognition and normalisation involves identifying, within texts, expressions that refer to points or periods of time, and re-expressing these temporal references in a standard format which (a) precisely describes the semantics of the expressions, (b) disambiguates dates and times from different time zones, and (c) makes it easier to determine the sequencing of events described in these texts.

The time expression normalisation task is an interesting and challenging one because, while some temporal references appear in well-defined formats, others are expressed using a wide range of natural language constructions, and are often ambiguous, requiring analysis of the surrounding text in order to arrive at an interpretation. Of course, there are cases where information external to a document—perhaps contained in another document, or best considered part of world knowledge—is required in order to interpret a temporal expression; such cases are not considered here.

There have always been sections of the linguistics, philosophy and natural language processing communities that have been interested in temporal referring expressions. However, interest in the recognition and interpretation of these expressions has grown significantly as a result of the DARPA-sponsored competitions in named entity recognition from the mid-1990s onwards. In contrast to earlier work in the area, these competitions and related exercises introduced a rigorous evaluation paradigm, whereby success or failure was measured in terms of the ability of software systems to replicate human 'gold standard' annotations of the scope and interpretation of temporal referring expressions.

Undoubtably, the key events and exercises that have played a role in this growth have been the Message Understanding Conferences (MUCs) in 1996 and 1998, and three workshops associated with the Automatic Content Extraction (ACE) program[1] in 2004, 2005 and 2007. While both MUC evaluations covered only recognition of two types of temporal expressions (dates and times), there has been a significant increase in the level of task difficulty in the ACE competitions. The fundamental move forward here was the addition of a normalisation task to the recognition task: annotations were provided for the interpretation of dates and times by using TIMEX2, a slightly modified version of ISO 8601, as the standard for the representation of normalized dates and times. The introduction of TIMEX2 also influenced the recognition task, as the range of temporal expressions to be recognised was broadened significantly as compared to the MUC-6 and MUC-7 task definitions.

Subsequently, the TIMEX2 standard has evolved through a number of versions, partially due to the wide interest it has received in the community, and the existence of the ACE program and similar competitions. This has also resulted in quite a large number of temporal expression taggers being constructed by the participants in these competitions. Details of the current, and most likely final, version of the standard are provided in (Ferro et al., 2005).

In this paper we present the DANTE (Detection And Normalisation of Temporal Expressions) system, which, as its name suggests, performs both recognition and normalisation of temporal expressions. Currently, the system works only for English texts; however, its extension to other languages is facilitated by its modular architecture, where some components are language independent. In January 2007, DANTE participated in the ACE Time Expression Recognition and Normalization (TERN) task.

The rest of this paper is organized as follows. First, in Section 2 we describe related work, briefly presenting other existing temporal expression taggers. Then, in Section 3, DANTE's system architecture and development process is discussed. Section 4 provides information on DANTE's performance both in terms of recognition and

[1]See http://www.nist.gov/speech/tests/ace.

normalisation results, and in terms of resource consumption and execution time. In Section 5 we discuss the most problematic cases for DANTE that give rise to errors in the current version. Conclusions and future work are described in Section 6.

2. Related Work

The earliest approaches, typical of work undertaken for MUC-6, were based on the construction of handcrafted rules using a grammatical formalism that would match both fixed-format dates and times, and a range of expressions in natural language within the scope defined in the guidelines. For MUC-7, there were both solutions based on transducers, such as those described in (Mikheev et al., 1998) and (Krupka and Hausman, 1998), and also other techniques, such as hidden Markov models as used in IdentiFinder (Miller et al., 1998).[2] In both MUC competitions, the results achieved for TIMEX recognition by the best systems were high:

- at MUC-6, Recall of 93% and Precision of 96%; and
- at MUC-7, Recall of 89% and Precision of 99% for dates, and Recall of 81% and Precision of 97% for times (Krupka and Hausman, 1998).

TempEx (see (Mani and Wilson, 2000)) was the first TIMEX2 tagger developed. It is a relatively simple Perl tool that implements a few heuristics based on part-of-speech tags using finite state automata. It also performs limited normalisation of the expressions. The most recent version, from December 2001, implements the 2001 version of the TIMEX2 standard. There are certain classes of phrases that are not recognized by this tool: for example, *the last Monday of January*, *the end of 1999*, and *late yesterday morning*. This tool was provided to all participants of ACE TERN 2004 for use as an external source of text features; as such, it provides a reasonable baseline for performance on new data.

GUTime (Verhagen et al., 2005) was developed as an extension of TempEx for the purpose of constructing an automatic temporal annotation tool for TimeML (see (Pustejovsky et al., 2004)). TimeML is a sophisticated schema for the annotation of events; its complexity means that automatic tagging of events is best achieved via a cascade of modules that successively add more and more TimeML annotations to the document being processed. In this context, GUTime is the module responsible for the detection of temporal expressions and the introduction of the `TIMEX3` tag into the annotations. GUTime's coverage of temporal expressions is greater than that of TempEx. In addition, it also handles TIMEX3's functional approach to expressing values: that is, for relative expressions it first identifies what function is realised by an expression (for example, for *tomorrow* it would be PLUS ONE DAY), and the actual value of that function (for example, *25th January 1996*) can be calculated at a later stage.

Chronos (Negri and Marseglia, 2005) is a more complex system designed to perform both recognition and normalisation of temporal expressions. Text processing in Chronos involves tokenization, statistical part-of-speech tagging and multiwords recognition based on a list of 5000 entries retrieved from WordNet. Then, the text is processed by a set of approximately 1000 basic rules that recognize temporal constructions and gather information about them that is expected to be useful in the process of normalization. This is followed by the application of composition rules, which resolve ambiguities when multiple tag placements are possible. The results in terms of F-measure on the TERN 2004 data are 92.6%, 83.9%, 87.2% for detection, recognition and `VAL` attribute value, respectively.

	Detection	Extent Recognition	`VAL` Attribute
GUTime	85	78	82
ATEL	90.4	81.5	–
LingPipe	89.1	75.8	–

Table 1: The F-measure results for GUTime, ATEL and LingPipe on ACE TERN 2004 data.

The increasing availability of corpora annotated with temporal expressions makes it possible to apply supervised machine learning techniques to the time expression recognition problem. Examples of such systems are ATEL (Hacioglu et al., 2005) and Alias-i's LingPipe.[3] The former is based on Support Vector Machine (SVM) classifiers, and the latter is constructed using a Hidden Markov Model (HMM). Table 1 presents their performance.

3. System Architecture and Development

We take the view that an important step towards a truly broad coverage yet semantically well-founded approach is to recognize that there is a principled distinction to be made between the interpretation of the semantics of a temporal expression devoid of its context of use, and the fuller interpretation of that expression when the context is taken into account. The first of these, which we refer to here as the **local semantics** of a temporal expression, should be derivable in a compositional manner from the components of the expression; determining the value of the second, which we refer to as the **global semantics** of the expression, may require arbitrary inference and reasoning. Such a distinction is implicit in other accounts: Schilder's (Schilder, 2004) use of lambda expressions allows representation of partially specified temporal entities, and the temporary variables that Negri and Marseglia (Negri and Marseglia, 2005) construct during the interpretation of a given temporal expression capture something of the same notion.

The above assumptions are reflected in our design, which comprises separate and independent modules for the recognition and normalisation subtasks. These components communicate via an intermediate format for expressing the local semantics of temporal expressions, as described in (Mazur and Dale, 2006) and (Dale and Mazur, 2006).

The stages of text processing are organized as a pipeline of processing resources run, using the architec-

[2] See also (Bikel et al., 1999) for an extended description.

[3] See `http://www.alias-i.com/lingpipe`.

tural constructs provided in GATE (Cunningham et al., 2002). The elements in our pipeline are a tokenizer, gazetteers, a sentence splitter, a POS tagger, named entity recognition, temporal expression recognition, and temporal expression interpretation.[4]

3.1. Temporal Expression Recognition

The temporal expression recognizer is implemented using a JAPE grammar. The grammar consists of five phases which are run over a document in sequence. Each phase contains rules which match annotations introduced by earlier processing components (for example, the tokenizer or POS tagger) and JAPE grammar phases. There is also one initial additional phase which consists only of macros used in the grammar rules. Altogether there are 80 macros and 250 rules. Macro expansions are textually copied into the bodies of rules, and then the rules are compiled into Java code.

JAPE rules are traditional pattern–action rules, where the left-hand side contains the pattern to be matched, and the right-hand side specifies the action to be taken when the pattern is matched. The pattern on the left-hand side is written using JAPE syntax, but the right-hand side can be implemented either in JAPE or directly in Java code. Our recognition rules use 31 gazetteers with a total of 1418 entries: these are strings used in the expression of dates and times, such as numbers written in words; the names of days, months and time zones; and the most common fractions.

The development of our temporal expression recognition module took two and a half person months. The module was developed on the basis of the TIMEX2 guidelines and the examples contained therein; then we tested DANTE on the ACE 2005 development data and identified frequently-occurring cases which were problematic for the system. Addressing these problems constituted a second stage of system development.

3.2. Temporal Expression Interpretation

The interpreter module is a process that steps through a document sentence by sentence. Each temporal expression identified in the recognition stage is passed through the interpretation module, which transforms the local semantic representation into a document-internal semantic representation. The interpreter is fully implemented in Java and includes a library of functions for various calculations on dates and times. This module took approximately one and a half person months to develop.

In our current model, we assume that a document has a simple linear structure, and that any hierarchical structure in the document has no bearing on the interpretation of temporal expressions; for present purposes we also make the simplifying assumption that the **temporal focus** used to compute document-level values for temporal expressions does not advance during the processing of the document. Both assumptions may not always hold true, but are likely to work for the majority of cases we are dealing with.

Depending on the type of the temporal expression being interpreted (fully specified point in time, underspecified point in time, relative expression, duration, frequency and so on), different actions are taken. The two basic operations used in the interpretation are unification with some reference date and the addition or subtraction of a specified number of units to or from a reference date. The type of the temporal expression is also important for determining which TIMEX2 attributes other than VAL should be generated.

4. Evaluation

The most significant evaluations of DANTE to date are our participation in the ACE 2007 TERN task, and our subsequent re-evaluation of the system on the same data after further development on the ACE 2005 development data set.

The execution time for our text processing modules is presented in Table 4 as measured on a laptop with a 2GHz Intel Core Duo processor and 2GB of available RAM memory; only one core of the processor was used for processing documents. In characterising the processing cost, we do not take into account initialization of the system, the exporting of results into XML files, and the postprocessing required to meet the ACE formatting requirements, including the conversion of results from our inline XML annotation into the APF XML format.

Memory consumption during system execution is to some extent dependent on the size of the processed document, but on the ACE 2007 evaluation the variation was not great (from 116MB to 126MB). The system also required approximately 15MB of disk space to store the input corpus. The ACE 2007 evaluation data consisted of 254 documents from six different domains (see Table 4). As one might expect, documents were not equally distributed across the domains, both in terms of the number of documents and the total size of documents in a domain. We ran the system for each document source type separately in order to identify variations in performance across the different domains.

In the ACE evaluations a correctly recognized time expression is one which has a strictly accurate extent and correct values for all the TIMEX2 attributes. An annotation generated by the system is classified as matched with an annotation from the gold standard if there is minimum 30% text span overlap between them

The ACE 2007 evaluation data included 2028 time expressions to be recognized and interpreted. Across all domains we currently achieve 54.7, 57.6 and 56.1 for precision, recall and F-measure, respectively, for correct recognition of temporal expressions. After applying weights[5] to particular elements which are subject to evaluation these

[4] We refer to this here as an "interpreter" since what is really happening in the "normalisation" process is in fact the interpretation of a temporal expression in the context of the rest of the document.

[5] In the ACE 2007 TERN evaluations the weights were as follows: 1.0 for type VAL, 0.5 for ANCHOR_VAL, 0.25 for ANCHOR_DIR, 0.1 for MOD, 0.1 for SET, and 0.1 for extent (at least 0.3 overlap between matched elements, otherwise elements are not considered matched at all). TIMEX2 mention value (cost) for spurious TIMEX2 mentions was -0.75.

scores are 69.7, 69.2 and 69.4. The overall ACE TERN value for DANTE is 57.2. These results indicate that DANTE's performance is already very close to state-of-the-art systems. For 13 documents in the corpus we scored 100%, meaning that all time expressions in these documents were recognised and interpreted correctly with no false positives (spurious matches) being generated.

The precision, recall and F-measure metrics for attributes are presented in Table 2, calculated for those expressions which matched with the gold standard. Table 3 presents detailed performance statistics for DANTE across all domains.

TIMEX2 Attribute	Precision	Recall	F-Measure
VAL	99.8%	98.0%	98.9%
MOD	76.0%	75.0%	75.5%
SET	100.0%	100.0%	100.0%
ANCHOR_VAL	88.4%	83.5%	85.9%
ANCHOR_DI R	88.1%	87.4%	87.8%

Table 2: Attribute value recognition evaluation for DANTE on ACE 2007 evaluation data.

5. Error Analysis

In order to determine which aspects of DANTE most need attention, we analysed the errors made by the system on the **ACE 2005 development data set**; this is larger than the evaluation data set, containing 5428 temporal expressions (as annotated in the gold standard).

5.1. Errors in Recognition

Using the evaluation tool provided by NIST for use in the ACE program we have found that the errors in recognition of temporal expressions can be broken down as follows:

- 1056 spurious matches (51.09% of our errors),
- 586 missing temporal expressions (28.35%), and
- 425 extent errors (20.56%).

Based on an analysis of what falls into the set of spurious matches, we observe that about 50% of these are in fact due to legitimate temporal expressions that are missing from the gold standard. For the remaining 50%, DANTE's errors are generally due to ambiguity in the meaning of some expressions. Most of these are expressions based on the following trigger words: *now*, *fall*, *a second*, *night*, *May*, *March* and expressions which contain numbers wrongly recognised as years, dates or hours. In most of these cases the fix should be quite straightforward.

Among those strings which are not recognised as temporal expressions, most errors are due to either the ambiguous trigger word *time* or expressions whose extent can only be determined by syntactic means, as in *four days after Americans first penetrated the Baghdad outskirts.* In the latter case, DANTE only recognises the string *four days* as a temporal expression, and since this corresponds to less than 30% of the total length of the gold-standard expression, this is not treated by the scoring tool as a matched expression. There is also large group of expressions which appear with very low frequency, and which were not therefore considered a priority when developing DANTE.

An analysis of those cases where DANTE identified a temporal expression with an incorrect extent shows that the problems are due to failure to recognise the following: some variations of time zones; modified expressions (for example in *just recently* we recognised only *recently*); and expressions built from smaller constituent expressions (for example, in *around 11:30 Saturday night* we incorrectly recognise the time and date as separate expressions).

5.2. Errors in Interpretation

For the interpretation task, i.e., the determination of values for the TIMEX2 attributes, the error statistics are as follows (for each attribute, these show the numbers of expressions with an incorrect value):

- 1460 for the VAL attribute (69.00%),
- 1067 for the ANCHOR_VAL attribute (50.43%),
- 897 for the ANCHOR_DIR attribute (42.39%),
- 192 for the MOD attribute (9.07%), and
- 53 for the SET attribute (2.50%).

The total number of expressions with at least one incorrect attribute–value was 2116.

In determining the correct value of the VAL attribute, the biggest problem is interpreting names of weekdays such as *Tuesday*, where we often get a date one week earlier or later than the correct date. However, in the development data we observed that in about 15% of cases where there is a difference in the value generated by DANTE and that provided in the gold standard, the gold standard appears to be incorrect. Other errors in the gold standard further weaken the reliability of these numbers. Our worst results are obtained for the MOD attribute: many expressions are not given a value at all or they are given the wrong value. There are also cases when DANTE interprets expressions as modified, but they are not according to the gold standard annotators. Less problematic is the SET attribute, as it is quite obvious which expressions refer to more than one point in time, and the attribute is of a binary type.

6. Conclusions and Future Work

We have presented the DANTE system for recognition and interpretation of temporal referring expressions in English natural language texts. The system has been evaluated on the ACE 2007 evaluation corpus, which is a data set widely accepted by the community as a gold standard for the TERN task. The achieved results are good enough to use DANTE in many applications that require the interpretation of temporal expressions in text processing, such as information extraction and question answering.

The evaluation has brought to light several areas where DANTE can be improved. Our error analysis indicates that the following steps will be the most important in producing a more robust solution:

- First, we need to further develop the recognition grammar. This will require both the addition of vocabulary to our existing rules, and also the development of new

Domain	Entities in gold standard	Spurious	Missing	Error	Precision	Recall	F-measure	ACE Value
Broadcast Conversations	142	33	29	43	47.9	49.3	48.6	46.5
Broadcast News	322	103	38	69	55.6	66.8	60.6	55.2
Newswire	894	128	110	273	56.0	57.2	56.6	58.8
Telephone Conversations	70	23	11	25	41.5	48.6	44.7	51.4
Usenet Newsgroups	167	20	22	43	61.8	61.1	61.4	65.3
Weblogs	433	68	58	139	53.3	54.5	53.9	57.3
Total	2028	375	268	592	54.7	57.6	56.1	57.2

Table 3: The results of evaluation of the DANTE system on the ACE 2007 evaluation data set.

Domain	No of docs	Time [s]	Av. time per one doc [s]	Approx. size [B]	Av. time per 10kB [s]
Broadcast Conversations	9	10.902	1.211	48,722	2.29
Broadcast News	74	15.983	0.216	75,731	2.16
Newswire	106	43.632	0.412	209,973	2.13
Telephone Conversations	6	12.221	2.037	54,522	2.30
Usenet Newsgroups	13	11.398	0.877	48,377	2.41
Weblogs	46	29.355	0.638	137,549	2.19
Total	254	123.491	0.486	574,874	2.20

Table 4: Execution times on the ACE 2007 eval data set.

rules covering previously unseen structures. As the system is rule-based, this also requires careful testing to ensure that the addition or modification of rules does not introduce any incompatibilities or inconsistencies in the grammar.

- Second, we need to improve our mechanism for focus tracking in documents in order to more accurately resolve ambiguities. Although using the document creation date as the temporal focus often works fairly well, it is not reliable enough alone for a state of the art temporal expressions tagger.
- Although execution time performance is not critical in an evaluation such as this, we are keen to develop a robust and scalable solution. Our third task will therefore include identifying scalability improvements to DANTE.

7. Acknowledgements

We acknowledge the support of the Defence Science and Technology Organisation in carrying out the work described here.

8. References

Bikel, D.M., R. Schwartz, and R.M. Weischedel, 1999. An Algorithm that Learns What's in a Name. *Machine Learning*, 34:211–231.

Cunningham, H., D. Maynard, K. Bontcheva, and V. Tablan, 2002. GATE: A framework and graphical development environment for robust NLP tools and applications. In *Proceedings of the 40th Anniversary Meeting of the ACL*.

Dale, Robert and Paweł Mazur, 2006. Local Semantics in the Interpretation of Temporal Expressions. In *Proceedings of the Workshop on Annotating and Reasoning about Time and Events (ARTE)*. Sydney, Australia: Association for Computational Linguistics.

Ferro, L., L. Gerber, I. Mani, B. Sundheim, and G. Wilson, 2005. TIDES 2005 Standard for the Annotation of Temporal Expressions. Technical report, MITRE.

Hacioglu, K., Y. Chen, and B. Douglas, 2005. Automatic Time Expression Labeling for English and Chinese Text. In Alexander F. Gelbukh (ed.), *Computational Linguistics and Intelligent Text Processing, 6th International Conference, CICLing'05*, Lecture Notes in Computer Science. Springer.

Krupka, G. and K. Hausman, 1998. IsoQuest Inc.: Description of the NetOwl(TM) Extractor System as Used for MUC-7. In *Proceedings of the 7th Message Understanding Conference (MUC-7)*. Fairfax, Virginia.

Mani, I. and George Wilson, 2000. Robust Temporal Processing of News. In *ACL '00: Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*. Morristown, NJ, USA: Association for Computational Linguistics.

Mazur, Paweł and Robert Dale, 2006. An Intermediate Representation for the Interpretation of Temporal Expressions. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*. Sydney, Australia: Association for Computational Linguistics.

Mikheev, A., C. Grover, and M. Moens, 1998. Description of the LTG System Used for MUC-7. In *Proc. of MUC-7 Conf.*.

Miller, S., M. Crystal, H. Fox, L. Ramshaw, R. Schwartz, R. Stone, R. Weischedel, and the Annotation Group, 1998. BBN: Description of the SIFT System as Used for MUC-7. In *Proceedings of the 7th Message Understanding Conference (MUC-7)*. Fairfax, Virginia.

Negri, M. and L. Marseglia, 2005. Recognition and Normalization of Time Expressions: ITC-irst at TERN 2004. Technical Report WP3.7, Information Society Technologies.

Pustejovsky, J., B. Ingria, R. Sauri, J. Castano, J. Littman, R. Gaizauskas, A. Setzer, G. Katz, and I. Mani, 2004. The Specification Language TimeML. In I. Mani, J. Pustejovsky, and R. Gaizauskas (eds.), *The Language of Time: A Reader*. Oxford University Press.

Schilder, F., 2004. Extracting Meaning from Temporal Nouns and Temporal Prepositions. *ACM Transactions on Asian Lang. Information Processing*, 3(1):33–50.

Verhagen, M., Inderjeet Mani, Roser Sauri, Jessica Littman, Robert Knippen, Seok B. Jang, Anna Rumshisky, John Phillips, and James Pustejovsky, 2005. Automating Temporal Annotation with TARSQI. In *Proc. of the ACL Interactive Poster and Demonstration Sessions*. Ann Arbor, Michigan.