What's the Date?
High Accuracy Interpretation of Weekday Names

Paweł Mazur[1,2]

Robert Dale[2]

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

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

Abstract

In this paper we present a study on the interpretation of weekday names in texts. Our algorithm for assigning a date to a weekday name achieves 95.91% accuracy on a test data set based on the ACE 2005 Training Corpus, outperforming previously reported techniques run against this same data. We also provide the first detailed comparison of various approaches to the problem using this test data set, employing re-implementations of key techniques from the literature and a range of additional heuristic-based approaches.

1 Introduction

Many temporal expressions in text are underspecified, requiring contextually-sourced information in order to determine their correct interpretation. In some cases, it is sufficient to determine what is sometimes called the **temporal focus**, so that the precise location of a relative temporal expression on a timeline can be determined with respect to this 'time of speaking'. Consider, for example, expressions like the following:

(1) three days ago
(2) last Monday
(3) in two weeks time

Once we know the temporal focus, calculation of the temporal location referred to in each of these cases is straightforward, since the temporal expressions themselves explicitly indicate what we will call **the direction of offset** (here, respectively, past, past and future). However, in other cases there is no explicit indication of the direction of offset from the temporal focus. This is most obviously the case when bare expressions based on calendar cycles—i.e., weekday names and month names—are used, as in the following example:

(4) Jones met with Defense Minister Paulo Portas on *Tuesday* and will meet Foreign Minister Antonio Martins da Cruz before leaving Portugal *Wednesday*.

Here, the proper interpretation of the references to *Tuesday* and *Wednesday* requires at the least a correct syntactic analysis of the sentence, in order to locate the **controlling verb** for each weekday name. The tense of this verb can then be used to determine the direction—either in the past or in the future—in which we need to look to establish the fully specified date referred to. In the case of example (4), this means determining that *Tuesday* is in the scope of the verb *met*, and that *Wednesday* is in the scope of the verb group *will meet*. As we note below, it turns out that there are cases where even the controlling verb does not provide sufficient information to determine the direction of offset. But even in those cases where the tense of the verb *does* provide the relevant information, there are two problems. First, especially when the sentences considered are complex, there is a non-negligible likelihood that the analysis returned by a parser may not be correct, and this is especially the case when the sentences in question contain structures such as prepositional phrases: the attachment of these is notoriously a source of ambiguity, and they just happen to often be the hosts to temporal expressions. Second, even if a parser provides the correct analysis, parsing technology is still computationally expensive to use when processing very large bodies of text; if we are interested in time-stamping events described in significant volumes of data, we would prefer to have a faster, more heuristic-based approach.

In this paper, we explore the development of a fast and high accuracy algorithm for the interpretation of weekday names, in particular with regard to determining the direction of offset to be used in the temporal interpretation of these expressions: in essence, how can we determine whether the day referred to is in the past or in the future?

The rest of the paper is structured as follows. In Section 2 we present some general observations on the interpretation of weekday names in text. Section 3 provides a review of related work. In Section 4 we describe the corpus used for evaluation, and in Section 5 we describe in detail the various algorithms we evaluated. Section 6 compares the results of the various algorithms when applied to the corpus, and Section 7 provides an error analysis. Finally, in Section 8 we draw some conclusions and point to future work.

2 The Problem

The interpretation of relative temporal expressions[1] can be seen as a two step process: (1) first we have to determine a reference point for the interpretation of the expression;[2] (2) then we have to calculate the actual position of the referred-to time on the timeline.

Once we have the reference point determined, the interpretation of the offset from this reference point requires us to determine the magnitude and direction of offset. As noted above, in some cases the tense of the controlling verb will indicate the direction of offset; but prepositional attachment ambiguity can easily damage the reliability of such an approach, as demonstrated by the following minimal pair:

(5) We can show you some pictures on *Monday*.
(6) We can show you some pictures from *Monday*.

In example (5), the correct PP attachment is required in order to determine that *Monday* is in the scope of the verb group *can show*, allowing us to infer that the Monday in question is in the future. Example (6), on the other hand, is quite ambiguous and requires world knowledge in order to determine the correct attachment.

We are interested, therefore, in determining whether some heuristic method might provide good results. In the rest of this paper, we focus on the determination of the direction of offset. We will not explicitly address the question of determining the temporal focus: although this is clearly a key ingredient, we have found that using the document creation date performs well for the kinds of documents (typically newswire stories and similar document types) we are working with. More sophisticated strategies for temporal focus tracking would likely be required in other genres.

3 Related Work

The literature contains a number of approaches to the interpretation of weekday names, although we are not aware of any pre-existing direct comparison of these approaches.[3]

Filatova and Hovy (2001) assign time stamps to clauses in which an event is mentioned. As part of the overall process, they use a heuristic for the interpretation of weekday names: if the day name in a clause is the same as that of the temporal focus, then the temporal focus is used;[4] otherwise, they look for any 'signal words' or check the tense of the verb in the clause. An analogous approach is taken for the interpretation of month names.

Negri and Marseglia (2005), in their rule-based system for temporal expression recognition and normalisation, use what they call 'context words', such as *following* or *later*, to decide on the interpretation of a weekday name. Consider the following example:

(7) He started studying on March 30 2004, and
passed the exam the following *Friday*.

Here, having identified the date *March 30 2004* (which happens to be a Tuesday), they then recognise the structure '*following* + trigger' and reason that the *Friday* is three days later.

[1] In the literature, a variety of different terms are used: (Schilder and Habel, 2001) call these expressions **indexicals**, and (Han et al., 2006b) uses the term **relative** for what we call **anaphoric** references: in our terminology, both deictic and anaphorical expressions are relative.

[2] This reference point is often referred to as the **temporal focus** or **temporal anchor**.

[3] Although Ahn et al. (2007) compared their results with those presented by Mani and Wilson (2000), they went on to point out that, for a variety of reasons, the numbers they provided were not really comparable.

[4] Filatova and Hovy use the term **reference point** for what we call the temporal focus.

There have also been machine-learning approaches to the interpretation of temporal expressions. Ahn et al. (2005) describes a system developed and tested on the ACE 2004 TERN test corpus. Using lexical features, such as the occurrence of *last* or *earlier* in a context window of three words, their maximum entropy classifier picked the correct direction ('backward', 'same', or 'forward') with an accuracy of 59%; the addition of features encoding information about tense increased the result to 61%.

Ahn et al. (2007) go on to describe a system using a classifier based on support vector machines and an extended set of features over a larger subset of the data. This algorithm was used to determine the direction of all relative temporal expressions, not just the names of weekdays. They used three sets of features:

1. Character type patterns, lexical features such as weekday name and numeric year, a context window of two words to the left, and several parse-based features: the phrase type, the phrase head and initial word (and POS tag), and the dependency parent (and corresponding relation) of the head.
2. The tense of the closest verb (w.r.t. dependency path), the POS tag of the verb, and the POS tags of any verbal elements directly related to this verb.
3. Features comparing year, month name and day name of a temporal expression to those of the document creation date.

Their experiments demonstrated that the third set was the most useful.

Han et al. (2006a) report on the development of the Time Calculus for Natural Language (TCNL), a compact formalism designed to capture the meaning of temporal expressions in natural language, which is built on top of their constraint-based calendar model (see (Han and Lavie, 2004)). In this formalism each temporal expression is converted to a formula in TCNL, which then can be processed to calculate the value of a temporal expression. Interpretation of weekday names uses the tense of the nearest verb chunk and the presence of lexical items such as *next*. Their temporal focus tracking mechanism allows correct interpretation of cases like 'I am free *next week*. How about *Friday*?', where the TCNL formula for *Friday*, being +{fri}, reflects the occurrence of *next* in the preceding sentence.

Table 1: ACE 2005 Training Corpus

Domain	#Docs	#Words	# TIMEX2
BC	60	40415	626
BN	226	55967	1455
CTS	39	39845	409
NW	106	48399	1235
UN	49	37366	741
WL	119	37897	1003
Total	599	259889	5469

Table 2: Weekdays in ACE 2005 Training Corpus

Domain	#Docs	# TIMEX2	# per doc
BC	4	7 (1.91%)	1.75
BN	25	31 (8.47%)	1.24
CTS	2	2 (0.54%)	1.00
NW	102	292 (79.56%)	2.86
UN	3	3 (0.81%)	1.00
WL	19	32 (8.72%)	1.68
Total	155	367 (100%)	2.37

Most closely relevant to the work described in the present paper are the approaches described in (Baldwin, 2002), (Jang et al., 2004) and (Mani and Wilson, 2000). Since we have re-implemented versions of these algorithms for the present paper, we leave description of these to Section 5.

4 Corpus and Experimental Setup

For this work we used the ACE 2005 Training Corpus, which is publicly available and distributed by the Linguistic Data Consortium (LDC).[5] It has already become the gold standard in the information extraction community, especially for the temporal expression recognition and normalisation (TERN) task, and currently it provides the largest available corpus of annotated temporal expressions. Table 1 presents some relevant statistics, and Table 2 shows the distribution of bare weekday names (as TIMEX2 counts) in the corpus across the various genres represented.[6]

For the work described here, we used only those documents in the corpus that contained at least one weekday name; all subsequent analysis makes use only of the gold standard annotations of the bare weekday names in these documents, thus significantly reducing corpus processing time. This results in a total of 367 instances, once errors (of which there are quite a few) in the gold standard

[5] The corpus's catalogue number is LDC2006T06.

[6] BC = Broadcast Conversations; BN = Broadcast News; CTS = Conversational Telephone Speech; NW = Newswire; UN = Usenet Newsgroups; and WL = Weblogs.

annotations have been repaired. We made the following changes to the gold standard data:

- One day name had been missed by the annotators; we added this.
- Some 40 values were corrected from the format YYYY-Wnn-m to YYYY-MM-DD: although both are correct in some sense, the ACE guidelines indicate that the second is the preferred form.
- Eight cases where the incorrect value had been provided by the annotators were corrected.

Specific details of these corrections, and the complete data set used, are available on the web.[7]

5 Evaluated Approaches

We implemented and evaluated a number of both simple and more complex approaches to determining what date is meant in a text when a bare weekday name is used. These methods, described below, can be divided into two main classes: (a) 7-day window based, and (b) tense analysis based. Our new algorithm is a hybrid solution that incorporates ideas from both of these approaches.

5.1 Baselines

Our baselines are motivated by the observation that days referred to by bare weekday names are typically temporally close to the temporal focus.[8] **Past 7-day Window (inclusive)**: This baseline looks for the specified day in a 7-day window whose last day is the temporal focus. In other words, day names are always assumed to refer to days in the last week, including the 'day of speaking'.

Past 7-day window (exclusive): This is the same as the approach just described, except that we look for the referred-to day in the week leading up to but not including the 'day of speaking'.

Future 7-day window (inclusive): This is the future-oriented version of the first approach described above: we look for the specified day in a 7-day window whose first day is the temporal focus. This assumes that all day name references are to the present or future.[9]

Future 7-day window (exclusive): In this case the 7-day window starts on the day following the 'day of speaking'.

5.2 Algorithms

5.2.1 Baldwin's 7-Day Window

This algorithm was presented in (Baldwin, 2002; Jang et al., 2004). It is similar to our window-based baselines, but in this case the temporal focus is the middle day of the 7-day window. This approach was used in their research after observing that 96.97% of weekday name expressions in their English corpus referred to dates within such a window. Suppose we have the following sentence in a document with creation date 2003-06-16 (a Monday):

(8) Police arrested her in Abilene, Texas, *Saturday* where she had moved with a friend June 2.

The 7-day window then spans from Friday (June 13) to Thursday (June 19). The reference to *Saturday* is assigned (correctly) the value of the second day in the window, i.e. 2003-06-14. Note that this method will deliver the wrong result when the referred-to day actually falls further than three days either side of the temporal focus. Suppose, for example, we have the following sentence in a document written on 2005-01-01 (a Saturday):

(9) We got into Heathrow on *Monday* morning.

Here the 7-day window spans from Wednesday to Tuesday, and so the reference to *Monday* will be assigned the incorrect interpretation 2005-01-03.

5.2.2 Mani and Wilson's Tense Estimation

In the system presented in (Mani and Wilson, 2000), weekday name interpretation is implemented as part of a sequence of interpretation rules for temporal expression interpretation more generally. This algorithm attempts to establish the tense of what we have called the controlling verb in the following way. First, it looks backwards from the temporal expression in question to any previous temporal expression in the sentence, or if there is none, to the beginning of the sentence. If no verb is found here, then it looks between the temporal expression and the end of the sentence; and if a verb is still not found, then it looks in front of any preceding temporal expression found back to the beginning of the sentence. If the verb found is in past tense, the direction of offset is assumed to be

[7] Visit `http://TimexPortal.info`.

[8] Recall that in the present work we take the temporal focus to be the document creation date.

[9] An informal check of email data drove Han et al. (2005) to use the simple strategy of always assuming that weekday names refer to days in the future.

Table 3: Interpretation rules

Tense	Example	Direction
Present Continuous	I am flying to New York on Monday.	Future
Past Simple	I wrote a paper on Monday.	Past
Future Simple	I will write a paper on Monday.	Future
Present Perfect	I have been writing a paper since Monday.	Past
Bare Past Participle	The draft written on Monday was useless.	Past
Modal Verb	I should finish the paper on Monday.	Future
Modal Verb + *have*	I should have submitted the paper on Monday.	Past

backwards; if the tense is future, then the forward direction is used. If the verb found is in present tense, then the temporal expression is passed to a further set of interpretation rules, which check for things like the occurrence of lexical markers such as *since* or *until*.[10] For example, in example (4), repeated below, the algorithm would correctly pick *met* for *Tuesday* and *will meet* for *Wednesday*, interpreting Tuesday as a day in a past and Wednesday as a day in future.

(4) Jones met with Defense Minister Paulo Portas on *Tuesday* and will meet Foreign Minister Antonio Martins da Cruz before leaving Portugal *Wednesday*.

However, this approach will not correctly interpret example (10):

(10) Still a decision has to made on what, if any, punishment he will face in the wake of that incident *Tuesday* night.

In this case, the wrong verb will be identified, and the direction of offset will be incorrect.

5.2.3 Simple Tense Estimation

As an alternative to Mani and Wilson's approach, we also implemented a much simpler tense estimation heuristic. This checks whether the sentence contains any tokens with the VBD (i.e., past tense) part of speech tag;[11] if one is found, then the direction of offset is assumed to be backwards, and if not, then we use the forward direction. In the case of example (4), this will assign the correct value to *Tuesday*, but the wrong value to *Wednesday*.

5.2.4 Dependency-based Tense Determination

The two previous algorithms attempt to determine the controlling verb using very simple heuristics. Of course, a more reliable way of determining the controlling verb is to use a parser. We used the Stanford parser's dependency information output (see (de Marneffe et al., 2006)) to find the controlling verb of a weekday name in a sentence. This algorithm does this by traversing the resulting dependency tree from the node containing the weekday name to its root until a verb is found, and then following further dependencies to identify the whole verbal sequence.

5.2.5 A Hybrid Algorithm

Heuristic methods for determining tense are risky, especially as the distance between the controlling verb and the temporal expression increases. We therefore propose a hybrid approach that attempts to leverage both tense estimation approaches like Mani and Wilson's, and Baldwin's window-based approach. This algorithm was developed on the basis of an error analysis of the results of using Baldwin's algorithm. It embodies a two-step approach, where we first look only in the very local environment for clues as to the tense of the controlling verb, then fall back on Baldwin's algorithm if no such evidence is found close by. First, we check if the temporal preposition *since* appears immediately in front of a weekday name; if so, the direction of offset is assumed to be backwards; otherwise, the algorithm looks for any verbs in a window of three tokens before and three tokens after the temporal expression. If a verb is found, then its tense is used to determine the direction (using the same rules as in Mani and Wilson's approach). If no verb is found, then a 7-day window with the temporal focus as the middle day is used, just as in Baldwin's algorithm.

[10] We have reimplemented this algorithm based on the description given in the cited paper, but some details are unclear, so we acknowledge that the original implementation might produce slightly different results.

[11] Where POS tags are required in our algorithms, we used Mark Hepple's part of speech tagger, an implementation of which is available as a plugin for the GATE platform (`http://gate.ac.uk`).

Table 4: Results

Algorithm	Errors	Correct
Past 7-day Window (Inclusive)	51	316 (86.10%)
Past 7-day Window (Exclusive)	240	127 (34.60%)
Future 7-day Window (Inclus.)	129	238 (64.85%)
Future 7-day Window (Exclus.)	316	51 (13.90%)
Sentence Tense Estimation	38	329 (89.65%)
Dependency-Based	29	338 (92.10%)
Mani and Wilson's	27	340 (92.64%)
Baldwin's 7-day Window	21	346 (94.28%)
Voting	16	351 (95.64%)
Hybrid	15	352 (95.91%)

Table 5: Processing times

Algorithm	Time [seconds]
Past 7-day Window (inclusive)	79.9
Past 7-day Window (exclusive)	79.7
Future 7-day Window (inclus.)	79.2
Future 7-day Window (exclus.)	79.4
Sentence Tense Estimation	80.6
Dependency-Based	616.5
Mani and Wilson's	80.9
Baldwin's 7-day Window	79.4
Voting	636.1
Hybrid	80.2

5.2.6 Voting

This algorithm uses a voting mechanism over the output of Baldwin's, Mani and Wilson's, and the Dependency-based Tense Determination algorithms. If all values are different (no majority) then Baldwin's result is used.

5.3 Tense Interpretation Rules

Once the verb group is found by any particular algorithm, it needs to be analysed to determine what its tense is; this information is then used to determine the direction of offset. The interpretation rules are summarized in Table 3.

6 Results

Table 4 presents the results achieved with each of the algorithms.

The 51% difference between the inclusive and exclusive baselines is indicative of the fact that, in this data, in over 50% of cases the correct date was in fact the document creation date. This phenomenon is due to the large proportion of newswire data in the corpus; in this genre, it is common to use the weekday name even when reporting on events that happen on the same day as the reporting takes place. Also of note is that the best performing baseline, 'Past 7-day window (inclusive)', achieves 86.10% accuracy despite its being an extremely naive approach.

All the algorithms tested here performed better than the baselines. The best performing algorithm was the Hybrid method, which made 15 errors, resulting in an accuracy of 95.91%; the Voting method came second with 16 errors. Baldwin's 7-day window algorithm correctly interpreted 94.28% of weekday names. The big advantage of this algorithm, along with all the baselines, is their complete resource independence: they do not use any parsers or POS taggers.

Perhaps surprisingly, Mani and Wilson's tense estimation heuristic was more effective than tense determination based on a dependency parse tree; this reinforces our earlier point about the risks of using parsers. It is also important to note that there are huge differences in execution time for parser-based approaches. Table 5 presents times for processing the entire corpus for temporal expression recognition and interpretation; the parser-based algorithm required 616 seconds, in contrast to around 80 seconds for each of the other algorithms.[12]

There were 296 cases (80.65%) that were correctly interpreted by all of the following algorithms: Sentence Tense Estimation, Mani and Wilson's, Dependency-based Tense Determination, Baldwin's 7-day Window, and Hybrid. There are also three cases where all these algorithms provided an incorrect value:[13]

(11) reporter: *friday night* in the gaza strip and a journalist is about to lose his life.

(12) president bush head to the g-8 summit in france on *friday* with victory over saddam hussein and in his pocket and a soaring approval rating by the american public, but do europeans share the same enthusiasm for the president?

(13) I will return this piece of shit on Friday, only to rent another vehicle *Monday morning* while we take the wife's car to the shop to get her 1400 bucks worth of damage repaired.

[12] Note that the parser was only called for those sentences that contained bare weekday names, and not for other sentences in these documents.

[13] We present these examples with their original spelling and casing.

In example (11), the algorithms interpreted *Friday night* as a day in future. However, this text is a case of what is sometimes called the **historical present**, where for rhetorical effect the author speaks in present tense from a past point in time; it is not obvious how any algorithm would determine the correct answer here. Example (12) is ungrammatical as a consequence of a missing 's' in *head*; consequently, the POS tagger did not annotate this word as a verb, and the algorithms identified *do* or *bush* as a verb, leading to the decision that the referred-to *friday* is in the future; however, the gold standard interpretation puts this in the past (note that, even with the correct verb form of *heads*, all the algorithms would still get the wrong date). It so happens the correct date here is also outside the 7-day window. In example (13), because the weekday name used is the same as the day name of the document creation date, all the algorithms assigned the document creation date instead of a date seven days later.

7 Error Analysis

The Hybrid Algorithm achieved the best accuracy of 95.91%, which corresponds to 15 error cases. These were as follows:

- Eight cases where there was no verb found in the three-token neighbourhood of the temporal expression; in these cases the 7-day window method was used, but this did not find the correct value.
- Three cases where the algorithm identified a verb that was not the controlling verb; for example, it picked *will meet* instead of *met* to interpret *Tuesday* in the sentence given in example (4).
- Two cases where the document creation date was very misleading (see below).
- Two cases where past tense was used to talk about plans for the future which were subsequently cancelled, as in *discussions were scheduled to end Friday, when Kelly was to fly*. . . .

In 204 cases the algorithm interpreted the weekday name based on a verb found in the three-token neighbourhood; and in 163 cases it used the fallback 7-day window strategy. Since the Hybrid Algorithm was built as an extension of Baldwin's method, it is worth knowing whether there were any cases where the original 7-day window method got the correct value and the Hybrid Algorithm got it wrong. There were six such cases:

- Two of them occurred for documents with a misleading document creation date. In a typical example, a document with the timestamp 17-04-2004 (a Thursday) contained the sentence 'Malaysia's Appeal Court *Friday* refused to overturn the conviction . . . '. As the document timestamp was used as the temporal focus, *Friday* was interpreted as a day in the past, when in fact it was the day after the timestamp.
- The other two cases demonstrate a weakness in our approach, exemplified by the sentence given in example (4): here the algorithm incorrectly uses the verb group *will meet* when interpreting *Tuesday*.
- The remaining two cases were cases where the verb groups *were scheduled to end* and *scheduled to begin* were used to talk about future events.

In these last cases, the controlling verb is an infinitive, and there is no way, in the absence of either world knowledge or a much more sophisticated analysis of the text, of determining whether the scheduled event is in the past or the future. Sentences like these are a particular problem for Mani and Wilson's algorithm, where a significant number of misinterpretations involve sentences in which the past tense is used to talk about subsequently-changed plans for future, as in the following:

(14) A summit between Sharon and his Palestinian counterpart, Mahmoud Abbas, had been planned for *Wednesday* but was postponed . . .

Here, this utterance could be legitimately produced both before and after the Wednesday in question, so no simple algorithm will be able to determine the direction of offset.

8 Conclusions and Future Work

We have investigated the problem of the interpretation of bare weekday names in texts, and presented a new heuristic which extends Baldwin's (2002) approach. Our evaluations on a widely-available data set show that our Hybrid Algorithm was the best performing algorithm, achieving an accuracy of 95.91% with 15 errors out of 367 instances. The algorithm is implemented within our DANTE system for temporal expression interpretation (Dale and Mazur, 2006; Mazur and Dale, 2007).

It seems quite possible that our heuristics take advantage of phenomena that are specific to newswire texts and other similar types of reportage. Although these are precisely the kinds of texts where, in our own work, we need to provide fast processing of large volumes of text, it remains to be seen how these heuristics fare when faced with a broader range of text types. In particular, other text types are likely to require more sophisticated approaches to temporal focus tracking than we have used here.

Also, we have not attempted to replicate here the machine learning approaches described in (Ahn et al., 2005) and (Ahn et al., 2007), nor Han's use of constraint satisfaction problem methods (see (Han et al., 2006a)). The comparative evaluation of these is left for future work.

References

Ahn, D., S. F. Adafre, and M. de Rijke. 2005. Recognizing and interpreting temporal expressions in open domain texts. In *We Will Show Them: Essays in Honour of Dov Gabbay, Vol 1*, pages 31–50.

Ahn, D., Joris van Rantwijk, and Maarten de Rijke. 2007. A Cascaded Machine Learning Approach to Interpreting Temporal Expressions. In *Proc. of HLT: The Annual Conference of the North American Chapter of the ACL (NAACL-HLT 2007).*

Baldwin, J. 2002. Learning Temporal Annotation of French News. Master's thesis, Dept. of Linguistics, Georgetown University, April.

Dale, R. and P. Mazur. 2006. Local Semantics in the Interpretation of Temporal Expressions. In *Proceedings of the Workshop on Annotating and Reasoning about Time and Events*, pages 9–16, Sydney, Australia, July.

de Marneffe, M.-C., B. MacCartney, and Ch. D. Manning. 2006. Generating Typed Dependency Parses from Phrase Structure Parses. In *Proceedings of the IEEE / ACL 2006 Workshop on Spoken Language Technology*.

Filatova, E. and E. Hovy. 2001. Assigning Time-Stamps to Event-Clauses. In Harper, L., I. Mani, and B. Sundheim, editors, *Proceedings of the Workshop on Temporal and Spatial Information Processing*, pages 1–8, Morristown, NJ, USA.

Han, B. and A. Lavie. 2004. A Framework for Resolution of Time in Natural Language. *ACM Transactions on Asian Language Information Processing (TALIP)*, 3(1):11–32.

Han, B., D. Gates, and L. Levin. 2005. Anchoring Temporal Expressions in Scheduling-related Emails. In Katz, Graham, James Pustejovsky, and Frank Schilder, editors, *Annotating, Extracting and Reasoning about Time and Events*, Dagstuhl Seminar Proceedings.

Han, B., D. Gates, and L. Levin. 2006a. From Language to Time: A Temporal Expression Anchorer. In *Proceedings of the Thirteenth International Symposium on Temporal Representation and Reasoning*, pages 196–203.

Han, B., D. Gates, and L. Levin. 2006b. Understanding Temporal Expressions in Emails. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 136–143, Morristown, NJ, USA.

Jang, S. B., J. Baldwin, and I. Mani. 2004. Automatic TIMEX2 Tagging of Korean News. *ACM Transactions on Asian Language Information Processing (TALIP)*, 3(1):51–65.

Mani, I. and G. Wilson. 2000. Robust Temporal Processing of News. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 69–76, Morristown, NJ, USA.

Mazur, P. and R. Dale. 2007. The DANTE Temporal Expression Tagger. In Vetulani, Zygmunt, editor, *Proceedings of the 3rd Language & Technology Conference (LTC).*

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

Schilder, F. and Ch. Habel. 2001. From Temporal Expressions to Temporal Information: Semantic Tagging of News Messages. In Harper, L., I. Mani, and B. Sundheim, editors, *Proc. of the Workshop on Temporal and Spatial Information Processing*, pages 1–8, Morristown, NJ, USA.