

From Temporal Expressions to Temporal Information: Semantic Tagging of News Messages

Frank Schilder and Christopher Habel

Department for Informatics

University of Hamburg

Vogt-Kölln-Str. 30

22527 Hamburg

Germany

{schilder|habel}@informatik.uni-hamburg.de

Abstract

We present a semantic tagging system for temporal expressions and discuss how the temporal information conveyed by these expressions can be extracted. The performance of the system was evaluated wrt. a small hand-annotated corpus of news messages.

1 Introduction

This paper describes a semantic tagging system that extracts temporal information from news messages. Temporal expressions are defined for this system as chunks of text that express some sort of direct or inferred temporal information. The set of these expressions investigated in the present paper includes dates (e.g. 08.04.2001), prepositional phrases (PPs) containing some time expression (e.g. *on Friday*), and verbs referring to a situation (e.g. *opened*). Related work by Mani and Wilson (2000) focuses only on the core temporal expressions neglecting the temporal information conveyed by prepositions (e.g. *Friday* vs. *by Friday*).

The main part of the system is a temporal expression tagger that employs finite state transducers based on hand-written rules. The tagger was trained on economic news articles obtained from two German news papers and an online news agency (*Financial Times Deutschland*, *die tageszeitung* and www.comdirect.de).

Based on the syntactic classification of temporal expressions a semantic representation of the

extracted chunks is proposed. A clear-cut distinction between the syntactic tagging process and the semantic interpretation is maintained. The advantage of this approach is that a second level is created that represents the *meaning* of the extracted chunks. Having defined the semantic representation of the temporal expressions, further inferences, in particular on temporal relations, can be drawn. Establishing the temporal relations between all events mentioned by a news article is the ultimate goal of this enterprise. However, at the current stage of this work the semantic analysis is still in progress. For the time being, we focus on the anchoring of the temporal expressions in the absolute time line and present an already substantial subset of a full semantics that will eventually cover the entire set of temporal expressions extracted.

Finally, the evaluation of the temporal expression tagger provides precision and recall rates for tagging temporal expressions and drawing temporal inferences.

2 Representing time in news articles

Since we focus on a particular text domain (i.e. news articles), the classification of temporal expressions can be kept to a manageable set of classes.

2.1 Classification of temporal expressions

The main distinction we make is between time-denoting and event-denoting expressions. The first group comprises chunks expressing temporal information that can be stated with reference to a calendar or clock system. Syntactically speaking,

these expressions are mainly expressed by prepositional, adverbial or noun phrases (e.g. *on Friday* or *today* or *the fourth quarter*).

The second group, event-denoting expressions, refers to events. These expressions have an implicit temporal dimension, since all situations possess a temporal component. For these expressions, however, there is no direct or indirect link to the calendar or clock system. These expressions are verb or noun phrases (e.g. *increased* or *the election*).

2.1.1 Time-denoting expressions

Temporal reference can be expressed in three different ways:

Explicit reference. Date expressions such as *08.04.2001* refer explicitly to entries of a calendar system. Also time expressions such as *3 p.m.* or *Midnight* denote a precise moment in our temporal representation system.

Indexical reference. All temporal expressions that can only be evaluated via a given index time are called indexical. Expressions such as *today*, *by last week* or *next Saturday* need to be evaluated wrt. the article's time stamp.

Vague reference. Some temporal expressions express only vague temporal information and it is rather difficult to precisely place the information expressed on a time line. Expressions such as *in several weeks*, *in the evening* or *by Saturday the latest* cannot be represented by points or exact intervals in time.

For the given domain of news article, the extraction of a time stamp for the given article is very important. This time stamp represents the production time of the news information and is used by the other temporal expressions as an index time to compute the correct temporal meaning of the expression. Note that an explicit date expression such as *24.12.* can only be evaluated wrt. the year that the article was written. This means that even an explicit temporal expression can contain some degree of indexicality.

2.1.2 Event-denoting expressions

Two types of event-denoting expressions have to be distinguished, on the one hand, sentences, and, on the other, specific noun phrases. In the

former case, the verb is the lexical bearer of information about the event in question, in the latter case, specific nouns, especially those created by nominalisation, refer to an event.

Since temporal information is the topic of the system described in this paper, only a subset of event-denoting nouns have to be considered. These expressions — as *election* in the phrase *after the election* — which serve as temporal reference pointers in building the temporal structure of a news, can be marked by a specific attribute in their lexical entry. Furthermore, in the text classes we have investigated, there is a small number of *event nouns*, which are used as domain dependent pointers to elements of temporal structures. For the domain of business and stock market news, phrases such as *opening of the stock exchange*, *opening bell*, or *the close* are examples of domain specific event expressions.

2.2 Representation of temporal information: the time domain

The primary purpose of the present paper is to anchor the temporal information obtained from natural language expressions in news messages in *absolute time*, i.e. in a linearly ordered set of abstract time-entities, which we call *time-set* in the following. One of the major tasks in this anchoring process is to augment the temporal information in case of indexical and vague temporal descriptions (see section 4.3 for more details). Since these expressions do not specify an individual time-entity of the time-set, it is necessary to add temporal information until the temporal entity build up from natural language is fully specified, i.e. can be anchored in the time-set.

2.2.1 The granular system of temporal entities

The temporal information obtained from news messages is organised in a granular system of temporal entities including such granularity levels as *GL-day*, *GL-week*, *GL-month* and *GL-year*.¹ Individual days are anchored by a

¹In the present paper we focus on the conception of *granularity level* in semantic and pragmatic inferences. Therefore, we do not discuss the formal notions of granular systems for temporal entities here. Compare, e.g. Bettini et al. (2000), for a framework of temporal granularity, which could be used for the purposes we discuss here.

date, e.g. `date(2001, 3, 23)`, on the *time line*, i.e. the time-set. Further information, for example, the *day of the week*, can also be included by an additional slot of the time entity: `time = ['Fri', date(2001, 3, 23)]`. Time entities of coarser granularity levels, e.g. weeks, are represented on the basis of intervals, which can be determined by a start, that is an entity of GL-day, and a specific duration: `time = ['Mon', date(2001, 4, 2), '7 days']`.²

The concept of temporal granularity is reflected linguistically, for example, in the use of demonstratives as determiners of time expressions in German: *dieser Freitag* ('this Friday') refers to that Friday which is located in the current week (i.e. the time entity of the next coarser level of temporal granularity). The same phenomenon holds with *dieser Monatserste* ('this first day of the month')

In the following we will apply the granularity structure of temporal expressions only with respect to the *finer than - coarser than* relation between levels of granularity, which is different from the *is part of* relation between temporal entities. For example, whereas between days and weeks there is a unique functional relationship, namely that there is exactly one week (as standard calendar unit) that an individual day is a part of, a week can temporally *overlap* with one or two months (Technically, *overlap* can be realized by temporal relations of Allen-style; see Allen (1983)). Nevertheless, GL-week *finer than* GL-month holds in the granularity system.³

²Whether the GL-week information remains implicit, i.e. is inferable from duration, or is made explicit, i.e. coded by a GL-week-stamp, depends on some design decisions dependent on the conceptual richness of domain modelling. For example, in a standardised world of ISO-weeks, which start on Monday, only, it is not necessary to use GL-week-stamps. On the other hand, if ISO-weeks, and business weeks—of five-day length—are conceptual alternatives, then it is appropriate to use explicit granularity-level stamps.

³The phenomena of overlapping temporal entities of different granularity systems, for example the *system of calendar time-entities* vs. the *system of business time-entities*, or the *astronomical system of seasons of the year* vs. the *meteorological seasons of the year* are especially relevant for processing vague and ambiguous temporal expressions. Due to the temporal and spatial limitations of this paper, we can not go into the details here.

2.2.2 Definition of temporal relations

Temporal relations are explicitly marked by temporal prepositions (e.g. *before*, *on* or *by*). We use the following seven temporal relation: *before*, *after*, *incl*, *at*, *starts*, *finishes*, *excl*. The preposition *on* as in *on Friday*, for instance, denotes the inclusion relation *incl*, whereas the preposition *by* as in *by Friday* is represented as *finishes*.

Note that the seven temporal relations employed by the current version are equivalent to sets of Allen's interval relations (Allen, 1983).⁴

before	{ <i>b, m</i> }
after	{ <i>bi, mi</i> }
incl	{ <i>d, s, f, eq</i> }
at	{ <i>di, si, fi, eq</i> }
starts	{ <i>s</i> }
finishes	{ <i>f</i> }
excl	{ <i>b, bi, m, mi</i> }

Table 1: the temporal relations used

3 Extraction of temporal information

Similar to other approaches to information extraction or tagging, a cascade of Finite State Transducers (FST) was employed. The following sections provides a brief introduction to this technique before the overall system architecture is described in more detail.⁵

3.1 Preliminaries

The temporal expression chunks are extracted via an FST. FSTs are basically automata that have transitions labelled with a translation instruction. A label of the form *a:b* indicates such a translation from *a* to *b*. Take as an example the simple FST in figure 1. If the input contains the sequence of the three subsequent characters *F*, *S*, and *T*, the same output is produced with the sequence of these three characters put into brackets. The input stream "*FSTs are basically automata*" is, for instance, translated, into "*[FST]s are basically automata*".

⁴Allen (1983) proposes a temporal reasoning system that contains all 13 conceivable relations between intervals: *b(efore)*, *m(eets)*, *o(verlaps)*, *s(tarts)*, *d(uring)*, *f(inishes)*, the 6 reverse relations *bi*, *mi*, *oi*, *si*, *di* and *fi* and *eq(ual)*.

⁵The semantic tagging system is written in SWI-PROLOG 4.0.2

on Monday (time-denoting expression)	<CHUNK id = t43 type = time sem = [incl,[E,t42]] time = ['Mon',date(2001,4,2), time(_,-,-), gl(_,-,day,-)] > by Friday </CHUNK>
ftd.de, Fr, 16.3.2001, 11:00 (document time stamp)	<CHUNK id = t1 type = time ag = 'FTD' sem = now time = ['Fri',date(2001,3,16), time(11,00,-), gl(-,second,now)] > ftd.de, Fr, 16.3.2001, 11:00 </CHUNK>
closed (event-denoting expression)	<CHUNK id = e23 type = event sem = close(e23) temp = [-,[t(e23),-]] said </CHUNK>

Table 2: Examples of tagged temporal expressions

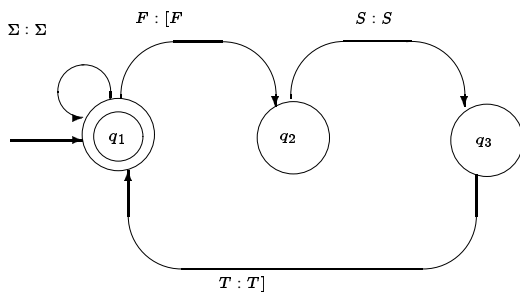


Figure 1: A simple FST

3.2 Classes of temporal information

The FSTs defined are fed by the output of a Part of Speech (POS) tagger.⁶ The POS tagger specifies the syntactic categories and a lemma for every word of the input text. The syntactic information is then stored in an XML file.⁷ Given the derived syntactic categories and the lemma information for every word of the text, several FSTs specialised into different classes of temporal expressions are run.

Temporal Expressions. One FST consisting of 15 states and 61 arcs tags all occurrences of time-

⁶A decision-tree-based POS tagger developed by (Schmid, 1994) was integrated into the system.

⁷Some of the XML and HTML handling predicates the system uses stem from the PiLLOW package developed by Manuel Hermenegildo and Daniel Cabeza (URL www.clip.dia.fi.upm.es/miscdocs/pillow/pillow.html).

denoting temporal expressions. The POS information stored in an XML file as well as a predefined class of temporal lemmas are used by this FST. The class of temporal lemmas used include days of the week (e.g. *Friday*), months (e.g. *April*) as well as general temporal descriptions such as *midday*, *week* or *year*. Since German is a very productive language regarding compound nouns, a simple morphological analysing tool was integrated into this FST as well. This tool captures expressions such as *Rekordjahr* ('record year') or *Osterferien* ('Easter holiday').

The extracted temporal expression chunks are marked by the CHUNK tag and an attribute `type = time`. See the first row of table 2 for an example. Note that the attributes `sem` and `time` carry semantic information. The meaning of these values are explained in section 4. detail.

Document time stamp. The document time stamp for a given article is crucial for the computation of almost all temporal expressions (e.g. *now*). In particular, this index time is indispensable for the computation of all temporal expressions that express an indexical reference (see the second row of table 2).⁸

⁸This FST consists of 7 states and 15 arcs. It also extracts the name of the newspaper or agency as indicated by the attribute `ag`. So far only the newspaper names and agencies

Verbal descriptions. Another FST that contains 4 states and 27 arcs marks all verbs as previously tagged by the POS tagger. As already pointed out these temporal expressions denote an event. The tag for such expressions is `<CHUNK type = event> </CHUNK>` (see table2; third row).

Nominal descriptions. So far there is only an experimental FST that extracts also nominal descriptions of events such as *the election*. More tests have to be carried out to determine a subset of nouns for the given domain. These nouns should then also be used to denote events mentioned in the text which can be combined with time-denoting expressions, as in *after the election in May*.

3.3 System output

After all expressions have been tagged, an HTML file is produced highlighting the respective expressions. See the snapshot in figure 2.⁹ While reading the output stream from the FSTs temporal inferences are drawn by the system. In particular, expressions bearing indexical references are resolved and the event descriptions are matched with the time denoting temporal expressions.

Note that the values for CHUNK attributes `sem`, `time`, and `temp` as indicated by the three examples in table 2 are PROLOG expressions. While translating the tagged text a PROLOG predicate triggers other predicates that compute the correct temporal information. An additional HTML file is also generated that contains the derived temporal information in standard ISO format, provided an explicit reference was given or was resolved. In the case of vague reference (e.g. *afternoon*) the semantic description is kept (e.g. 20:01:04:03:afternoon).¹⁰ In addition, the temporal relations holding between the events and times expressed

mentioned by the article of the training set can be extracted. A future version of the temporal expressions tagger should also be capable of tagging previously unknown names. However, note that this is rather a named entity recognition task and therefore goes beyond the scope of this paper.

⁹Time-denoting expressions are indicated by a dark (or magenta) background, while event-denoting expressions are indicated by a lighter (or yellow) background. The document time stamp is tagged by a very dark (or green) background.

¹⁰Future research will focus on the temporal inferences that can be drawn with these vague descriptions taking into account the different granularity levels.

by the text are stored as well.

4 Semantic descriptions and temporal inferences

4.1 Semantics for temporal expressions

With respect to processing temporal information, the crucial distinction between time-denoting and event-denoting expressions is that event-denoting expressions lack the direct link to temporal entities. An event-denoting expression (e.g. a verb) refers to an event of a certain type. The verb *to meet*, for instance, can be formalised as $meet(e_1)$. In order to add the temporal information to the event, a function `temp` is defined that gives back the time when the event occurred (i.e. run-time of the event). A time-denoting expression such as *on Monday* that is combined with the event description carries some temporal information that can further specify the run time `temp(e1)` of the event `e1`.

4.2 Semantics for temporal prepositions

PPs are the carrier of temporal relations. The semantics for a preposition is, therefore, as follows: $rel(t, e)$. For each preposition a temporal relation rel was defined. The preposition *by* expresses, for instance, the `finishes` relation, as in *by Friday*. Temporal expressions that do not contain a preposition are assumed to express an inclusion relation, as in *Die Pflegeversicherung war 1995 [...] in Kraft getreten* ('the statutory health insurance coverage of nursing care for the infirm took effect in 1995').

4.3 Derivation of meaning

The temporal information expressed by a sentence as in example sequence (1) is derived via unification of the semantic attributes derived for the temporal expression chunks.

- (1) Die US-Technologiebörse Nasdaq
 The US-technology stock market Nasdaq
 hatte {am Montag} mit einem Minus
 had on Monday with a minus
 von 3,11 Prozent bei 1782 Punkten
 of 3.11 percent at 1782 points
 [geschlossen].
 closed.

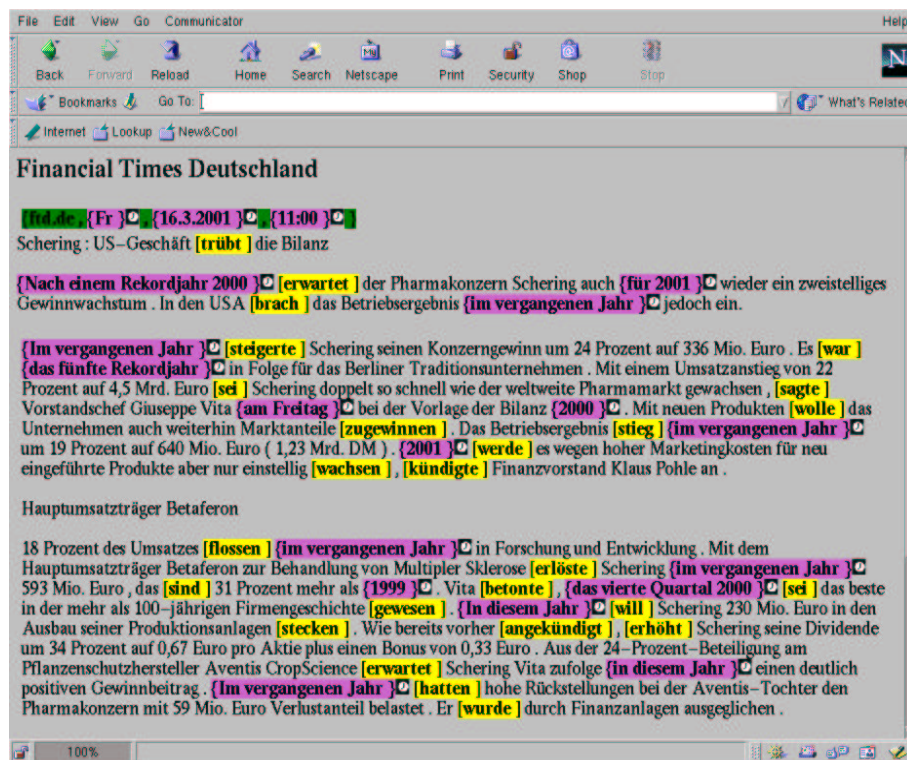


Figure 2: A snapshot of the temporal expressions tagger

‘The Nasdaq closed with a minus of 3.11 percent at 1782 points on Monday.’

Two temporal expressions are marked by the tagger: *am Montag* (‘on Monday’) and *geschlossen* (‘closed’). The former expression is a time-denoting expression that consists of a preposition and a time-denoting expression that is stored by the FST. The derivation of the semantics for this expression is done during the tagging process for the temporal expressions.

First, the preposition *am* (‘on’) denoting an inclusion relation between an event and a time is processed. The expressed temporal relation is represented by a PROLOG list (i.e. $[incl, [E, T]]$). After having processed the following noun referring to a time (i.e. *Monday*), the following semantic representation is obtained via unification: $sem = [incl, [E, t1]]$, where $t1$ refers to the following time stamp $time = [‘Mon’, date(, -, -), time(, -, -), gl([-, ‘1 day’, -)]$.¹¹

¹¹Note that the underscore “_” refers to an anonymous variable in PROLOG.

In the next step, the verbal expression tagger combines the temporal information derived for *am Montag* with the event representation for *geschlossen*. The following semantic representation is assigned to the verb *geschlossen* during the tagging of the verbal expressions: $sem = close(e23)$ $temp = [_, [t(e23), _]]$. This means that event $e23$ is of type closing and the run-time $t(e23)$ of this event stands in some to-be-specified relation with another expression. Next, the temporal information extracted by the FST specialised in time-denoting expression is unified with the value of the $temp$ -attribute. The result is $[incl, [t(e23), t1]]$.

So far, only the temporal relation that the event of closing happened within a time frame of one day has been determined. Since *Montag* contains an indexical reference, this reference has to be resolved. The document time stamp is needed here. All references regarding this index time are resolved during the generation of the HTML output file. Accordingly, the following time stamp is generated for *am Montag*: $time = [‘Mon’, date(2001, 4, 2),$

`time(-,-,-), gl([-,'1 day',-])`. The time information is left open because the current granularity level is `GL-day`.

However, this information could be further specified by modifiers such as in *nächstes Jahr* ('next year'). The third slot in `gl` is reserved for these modifiers. The first slot can be filled by temporal modifier that refer to a subpart of the expressed temporal entity, as in *Beginn des Jahres* ('beginning of the year'). The resulting representation of an expression such as *Beginn letzten Jahres* ('beginning of last year') is `gl([begin, year, last])`.

4.4 Pragmatic inferences for anchoring indexicals: The case of 'last'

Temporal expressions of the type *last Friday* are similar to the phenomena discussed in the section above. German has three lexemes, namely *letzt*, *vergangen* and *vorig* that express this idea. The differences in meaning are—in referring to a specific day—more of the type of individual preferences than of real alternatives in meaning. Which day is referred to by using *vorigen Montag*? This depends on the time of utterance. In general, there seems to be a tendency to interpret this expression as synonymous to *Monday of the previous week*, i.e. to make use of the *previous*-operation on the coarser level `GL-week`, instead of using this operation on the level `GL-day`. But, if uttered on Friday, our informants would give the Monday of the same week a preference in their interpretation.

Thus the *granularity-level up strategy* is not always successful. As an alternative strategy we propose the *strategy of the gliding time window*. Similar to the first proposal a granularity of week-size is relevant, but the relevant time entity in question is centered around the focused day of the week. In other words, looking forward and backward in time from the perspective of a Friday, the next Monday is nearer—or more activated—than the last Monday, although it is in the same calendar week. Thus, this Monday, i.e. the last Monday, has to be marked explicitly by *vorige*, and therefore, the Monday before this, has to be specified as *Montag der vorigen Woche* ('Monday of last week').

5 Evaluation

We evaluated the temporal expression tagger wrt. a small corpus consisting of 10 news articles taken from Financial Times Deutschland. We can report precision and recall rates regarding the recognition of simple temporal expressions and complex temporal expression phrases. Based on the extracted temporal expression chunks the temporal information was derived and evaluated.

5.1 Tagging results

First, the class of simple temporal expressions was tagged and analysed. Mani and Wilson (2000) call this class TIMEX expression (of type TIME or DATE). We computed the precision and recall values for our data regarding this type of expressions in order to obtain a better comparability with the results obtained by this earlier study. However, as pointed out earlier, we consider PPs carrying information regarding temporal relations as quite crucial for the derivation of temporal information. This class of complex temporal expressions provides more detailed information about the temporal information expressed by a text.

Table 3 contains the results of the evaluation wrt. the two classes of temporal expressions. There was a total of 186 simple and 182 complex temporal expressions previously annotated.

	Simple temp. Expr.	Complex temp. Expr.
Precision	92.11	87.30
Recall	94.09	90.66

Table 3: Performance of the temporal expressions tagger

An error analysis showed that the main source of missed temporal expressions was the occurrence of a combined temporal expression, as in *2000/01*. There were 6 cases when the tagger did not correctly analyse this type of expression.

5.2 Temporal information

The analysis of the temporal expressions included an evaluation of the temporal relations derived. Since all temporal prepositions and the class of temporal expressions that can be recognised by

the FSTs come with a predefined semantics, precision and recall rates are the same. The overall performance showed a precision and recall rate of 84.49. As indicated by table 4, errors were only made for expressions that express an indexical reference. These errors were in most cases due to a missing semantics assigned to the respective expression. Since this part of the system is still work in progress, we have not yet defined a complete semantics for all temporal expression. Hence the performance of the system regarding temporal inference is likely to improve in the future.

	Reference expressed		
	explicit	implicit	vague
Total	49	109	7
Wrong	0	25	0
Precision	84.49		

Table 4: performance of the temporal inference derivation

6 Conclusion and outlook

We presented a semantic tagging system that automatically tags the occurrence of temporal expressions such as *3. June, on Monday and last month* for German news messages. In addition, a semantics for most of the temporal expressions was defined so that temporal inferences were drawn regarding dates and events described. A more complex set of temporal expressions as extracted by recent systems (e.g. (Mani and Wilson, 2000)) was tagged. Our definition of temporal expressions also includes PPs capturing temporal relations. The system achieved an overall precision rate of 84.49 which is likely to go up as soon as the semantic definition of all temporal expressions will be completed.

Our system also covers indexical and vague temporal expressions. Temporal reasoning and pragmatic inferences drawn on the basis of these expression is the focus of on-going and future work.

The system we described in the present paper is intended to become a part of an experimental multi-document summarisation system currently under development. Our studies focus on financial news messages obtained from on-line infor-

mation services in Germany. The task the system has to solve is the production of summaries of the most recent — and especially, most referred to — topics. Our experience in this domain shows that there is one topic which leads to five to twenty news messages almost every day. These news messages are mostly unrelated, and they often only focus on the last one or two hours. Thus a bare collection of such messages is nearly useless for a reader who wants to be informed at the end of the day. For a user of an on-line information service summarisations of several articles on the same *hot topics* would have an enormous advantage compared to unsummarised collections of news messages.

The processing of temporal expressions plays a major role in building up these summaries, because temporal information is ubiquitous in this class of news. In addition, developing stories are reported via a stream of in-coming news messages. Producing coherent news depends heavily on the correct extraction of temporal information expressed by these messages.¹²

References

- James F. Allen. 1983. Maintaining Knowledge about Temporal Intervals. *Communications of the ACM*, 26(1):832–843.
- Regina Barzilay, Kathleen McKeown, and Michael Elhadad. 1999. Information fusion in the context of multi-document summarization. In *Proceedings of the 37th Annual Meeting of the ACL*, pages 550–557, Maryland, MD.
- Claudio Bettini, Sushil Jajodia, and Sean X. Wang. 2000. *Time granularities in databases, data mining, and temporal reasoning*. Springer-Verlag, Berlin.
- Inderjet Mani and George Wilson. 2000. Robust temporal processing of news. In *Proceedings of the 38th Annual Meeting of the ACL*, Hong Kong.
- Dragomir R. Radev and Kathleen R. McKeown. 1998. Generating natural language summaries from multiple on-line sources. *Computational Linguistics*, 24(3):469–500.
- Helmut Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of International Conference on New Methods in Language Processing*.

¹²Cf. (Radev and McKeown, 1998; Barzilay et al., 1999)