Finding Base Time-Line of a News Article

Sandip Debnath [1,2], **Prasenjit Mitra**[1,3], **C. Lee Giles** [1,3]

Department of Computer Sciences and Engineering[1]
The Pennsylvania State University
111 Information Sciences and Technology Building
University Park, PA 16802 USA
debnath@cse.psu.edu

eBusiness Research Center[2]
The Pennsylvania State University
401 Business Administration Building
University Park, PA 16802 USA

School of Information Sciences and Technology[3]
The Pennsylvania State University
332 Information Sciences and Technology Building
University Park, PA 16802 USA
{pmitra,giles}@ist.psu.edu

Abstract

An event without a time-line does not carry much information. Description of an event is useful only when it can be augmented with the time-line of its occurrence. This is more important with the on-line publishing of news articles. News articles are nothing but a set of text-based descriptions of events. Therefore the actual time-lines of the article as well as each individual event are most important ingredients for their informativeness. We introduce a novel approach to find the actual time-lines of news articles whenever available, and tag them with this temporal information. This involves a temporal baseline, which needs to be established for the entire article. Temporal baseline is defined as the date (and possibly time) of when the article had first been published, as stated in the article itself. Without a precise and correct temporal baseline, no further processing of individual events can be possible. We approached this problem of accurately finding the temporal baseline, with a Support-Vector based classification method. We found that the proper choice of parameters to train the Support-Vector classifier can result in high accuracy. We showed the data collection phase, training phase, and the testing phase and report the accuracy of our method for news articles from 26 different Websites. From this result we can claim that our approach can be used to find the temporal baseline of a news article very accurately.

Introduction

In Web-data mining, retrieving relevant documents has always been of great importance. This is related to other important areas of research such as text summarization and question answering. All these areas of research requires situating a document in proper time-line for better precision and recall.

We have seen prior efforts by Hwang and Schubert (Hwang & Schubert 1994), Kemp and Reyle (Kemp & Reyle 1993), Lascarides and Asher (Lascarides & Asher 1993), Allen (Allen 1984; 1995), Hitzeman (Hitzeman 1993) and others. They have used knowledge sources, tense, aspect, adverbs, rhetorical relations and of course background knowledge. For example, Lascarides and Asher used "narration" relation in sentences to identify the time of events. Others have found that news stories may not be a right place to use the narrative convention. As researchers found, events in news articles are tough to order. But even before starting to order the events in a news article, the first and foremost requirement is to find out the sentences carrying any occurrences of time units such as year, month, week, day and so on.

In most cases these time units are relative, meaning they are not expressed in complete time unit formats [1] This makes temporal ordering difficult in news articles. We have to fill up the missing parts according to the current notion or context of time in the article. For example, if month and date are available and year is not mentioned, then we can try to fill up the gap using the year when the news article was published. If time and day are mentioned, we can try to fill up the month and year similarly. We consider these time lines, where any part is missing, as relative time-lines, as they are relative in time with the context of the article. Therefore, in almost all cases, the missing parts can be understood from the temporal baseline. We refer temporal baseline or base time-line [2] interchangeably.

Base time-line is defined as the date and time (if available) of the publication of the news article, *as mentioned in the article itself*. Most news articles mention the date of publication somewhere inside the body. As mentioned above, properly calculating the exact time-lines of individual events of a news article depends on finding the base time-line (henceforth called **BT**). In this paper we focus on finding the **BT** of an article using Support-vector (**SV**) learning approach.

Not a Trivial Problem

The problem is not trivial due to several reasons. Firstly, temporal expressions are not always represented in a standard way. It has many different linguistic and geographical format. So it needs a proper grammar. Secondly the position of **BT** is not unique in a news article. Due to the open

[1] Complete time units are usually expressed in **YY::MM::DD::HH::mm::SS** following the ISO8601 guidelines or at least in a similar way, which can easily be converted to ISO8601 format using simple converter algorithms.

[2] In markup language (such as NewsML (new)) specifications, base time-line is called **DateLine**.

nature of Web-publishing and less enforcement of standardized and strict markup languages [3] news articles are very different from each other. Therefore it is not easy to extract the **BT** of a news article.

We found that almost every researcher assumed that this information is available somehow or other. Unfortunately that is not an practical assumption. Due to the information explosion on the Web, it is very difficult to keep up with the publishing speed of all news articles regularly and to crawl and cache them in timely manner. The natural delay in crawling, caching and putting them in proper temporal bins brings the necessity of an algorithm which can extract the **BT** of an article from the article itself. In case of unavailability, researchers usually assume the first available date of an article as its **BT**, As we see from figure 1 and as explained in subsection it is not a practical assumption.

We approached the problem by identifying the proper set of parameters by which we will a train a learning classifier. Agreeing with our intuition, our choice of parameter-set combined with the **SV** based classifier, produced high accuracy.

The rest of the paper is organized as follows. Some of the related previous efforts in this area of research are mentioned in the next section. In section "Our Approach" we describe our approach, in section "Time Format and Grammar" we mention the time format and standards, used in the experiment and in section "Data Preparation and Training Phase" we describe the method to prepare the data. We evaluated our classification accuracy in section "Testing and Evaluation Phase" and conclude thereafter.

Related Work

Tagging news article with temporal information has received much attention these days. Most of the prior work is based on Natural Language Processing (*NLP*). Unfortunately, we have not seen much prior work to find the **BT** of an article. The reason, as explained above, is due to the popular assumption that news data can be available from archive sorted according to the date and time of publishing or in case of un-availability researchers usually take the first available date and time as the **BT** of a news article. None of these assumptions are practical.

Therefore according to our search through the literature we could not find any previous attempt to find the **BT** of an article. We cited a few cases of general temporal information extractors here, some of them are outstanding work.

Starting with Allen's general theory of action and time (Allen 1984) we have seen very effective efforts towards structuring textual documents into temporally well-defined blocks. Some early approaches are very formal with finding time or time related expressions in documents but they were instrumental in setting up the ground-breaking steps. Based on that others tried to use rule-based or sometimes knowledge-based techniques. But most of the researchers related to text summarization, question answering or temporal ontology building, used or tried to use techniques and

[3]For example see NewsML (new), NITF (nit), XMLNews (xml) and RSS (rss) among others

```
DD   -> [01,02,03,04, ... 31] (Date)
MM   -> [00,01,02,03, ... 12] (Month){Jan=00/01}
YYYY -> [DDDD] (Year){D = [0,1,2, ..., 9]}
SM   -> [Jan,Feb,Mar, ..., Dec] (Short Month Id)
LM   -> [January,February,March, ..., December] (Long Month Id)
SD   -> [Sun,Mon,Tue, ..., Sat] (Short Day Id)
LD   -> [Sunday,Monday,Tuesday, ..., Saturday] (Long Day Id)
NU   -> [one,two,three,four, ... ] (Numbers)
NT   -> [first,second,third,fourth, ... ] (Numbered)
SP   -> [new year,valnetine,christmas,Independence, ... ] (Special Days)
HH   -> [01,02,03,04, ... 24] (Hour)
NN   -> [00,01,02, ... 59] (Minute)
SS   -> [00,01,02, ... 59] (Second)
AP   -> [AM,PM,HRS,HR,'o clock etc.] (Time Designator)
etc.
-------------------------------------------------------------------
Date -> [MMsDDsYYYY,YYYYsMMsDD,SMsDDsYYYY,LMsDDsYYYY,NTsSMsYYYY, ... ]
Refd -> [day before,day before yesterday,yesterday,today,tomorrow, ... ]
Refm -> [last month,present month,this month,next month, ... ]
Refy -> [last year,this year,present year,next year, ... ]
Time -> [HH AP,HHsNN AP,HHsNNsSS AP, ... ]
etc.
```

Figure 2: A portion of our temporal grammar used in the **TimeFinder** algorithm.

advances in *NLP*. *NLP* has its roots long back in time with Reichenbach (Reichenbach 1947) who pointed out the difference between the point of speech (time of utterance), the point (time) of the event and point of reference or the reference time.

In Time Frames (Koen & Bender 2000), Koen and Bender stated the benefits of the time augmentation of news. Their time extractor extracts time with moderate precision and recall.

MIT's Questioning News System (Sack 1997) used individual documents of a set, but did not create a temporal structure as such.

Other researchers such as Allen (Allen 1995), Dorr (Dorr & Olsen 1997), Mani (Mani & Wilson 2000), Lascarides (Lascarides & Asher 1993), Passonneau (Passanneau 1988), Ferro (Ferro *et al.* 2001), tried to approach it from *NLP* perspective using discourse structures, tense of the verb or the aspect. But as we have seen and explained before there is not enough evidence of classifying the temporal expressions using machine language techniques to find out the **BT**. It sounds obvious that without the proper **BT**, no technique could give the proper time-line of any events inside the article.

Our Approach

We used a **SV** classifier (Hastie, Tibshirani, & Friedman 2003; Scholkopf *et al.* 2000; 2001) to find the accurate **BT** of a news article. We first compiled our own temporal grammar (figure 2). We then devised **TimeFinder** algorithm (based on this grammar) to find all possible temporal expressions inside an article. Once it finds and builds the temporal expression set, we compute the values of several parameters (these parameters are described in section) for

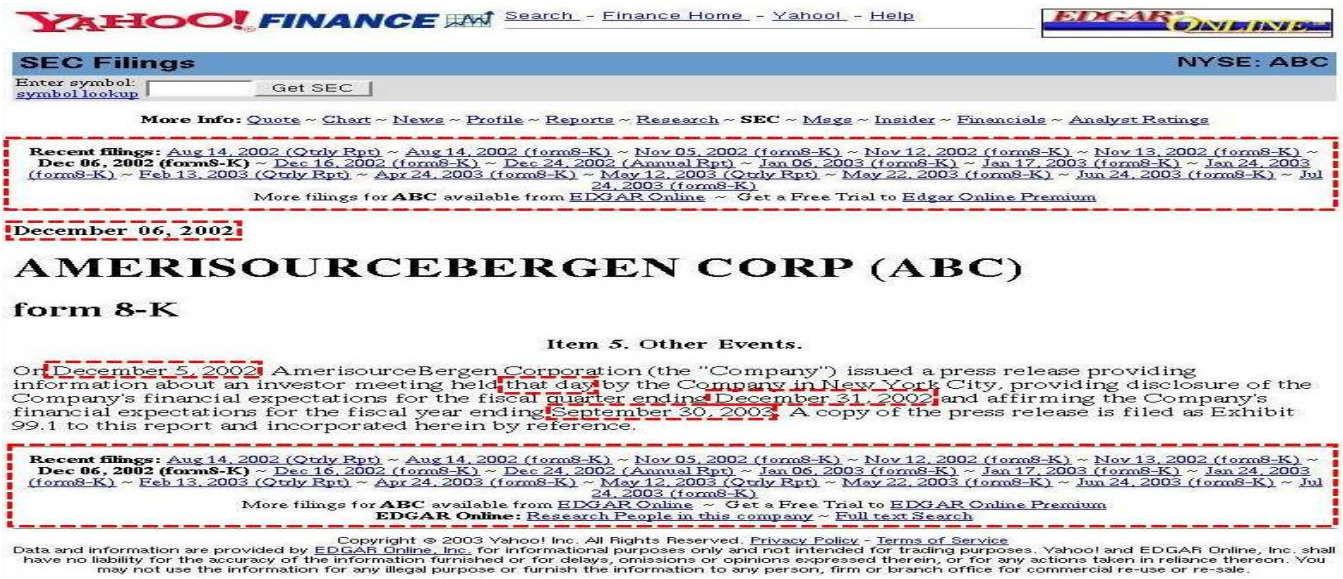

Figure 1: A sample Yahoo finance page with temporal expressions highlighted.

each of these time expressions. We train the **SV** classifier with this data. For a new set of articles, we process them first through **TimeFinder** to generate the set of temporal expressions available. We measure the parameter values for all these expressions and use our trained classifier to find the **BT**s of these new articles

Example

Figure 1 shows a sample HTML page from Yahoo finance [4] Web-site. Dotted rectangular boxes indicate the temporal expressions. We see several temporal expressions in the beginning of this article. However, none of them are the **BT**. The **BT** of this article is "December 06, 2002", (just above the heading **"AMERISOURCEBERGEN CORP (ABC)"**), contrary to the popular assumption of taking the first temporal expression as the **BT**. Clearly we can not rely on that assumption.

Time format and Grammar

We need to use a standard date and time format to identify every temporal expression in an article. According to the ISO 8601 guidelines, the standard way of expressing the date is **YYYY-MM-DD** and that of time is **hh:mm:ss**. There is also specifications and specialized off-the-shelf algorithms available for time zone and day-light saving time data [5]

We devised our own grammar in **TimeFinder** algorithm to extract temporal expressions. This include most of the time and date formats like "Jan 20, 2004", "01/20/2004", "2004-01-20", "Jan 20th, 2004", "20th Jan, 2004" etc. All of our news articles are from US news Web-sites and so there was no need of month-date-position disambiguation ("DD/MM/YY" or "MM/DD/YY") Moreover we also look for phrases like"2 months ago", "3 weeks after", "in 5 minutes", etc. Though we can not always fill every entries of "YYYY-MM-DD hh::mm::ss", at least to find the **BT** we do not have to worry about that too much. We followed the initial work by Koen and Bender (Koen & Bender 2000) and classify these expressions into the following classes.

- **Interval** Intervals are expressions like "twenty to twenty five minutes" (exactly same examples from Koen and Bender (Koen & Bender 2000)) or "twenty-to-thirty years" etc.
- **Age** Age defines expressions like "2 years old","A week after", "2 months before" etc. In relative sense we can get the time by using the precise base time-line (e.g. "Jan 01, 2004 20:34:00"+A week) equivalent to "Jan 08, 2004 20:34:00" for a workable algorithm.
- **Date** Dates are precise dates such "Jan 02, 2003", or "03/04/2004" or "03.04.2004" etc. There are various ways of expressing the date as explained above.
- **Precise Time** Precise times are time expressions such "2:00pm", or "Morning 7'O clock", or "18:15:01 hours". All these expressions precisely tell the time of the day. As always we fill-up the missing values with the base value, i.e. a missing second will be replaced by "00".
- **Time Duration** "Evening", "Morning", "Dawn" etc. are obviously not very precise time expressions, but we can get a clear idea from these expressions in the same way as described in the "Age" part. We can use the base time-line to find out the date and approximate time duration of the event.

[4] http://finance.yahoo.com

[5] Arthur David Olson and others maintain a database of all current and many historic time zone changes and daylight saving time algorithms: http://www.twinsun.com/tz/tz-link.htm

- **Special Day** "Christmas", "New Year's Eve", "Thanksgiving", "Rosh Hashanah" etc. come under this category which can precisely tell the date of the year without any "Precise Time". Here you need the base year to properly identify the date.

We show a snippet of our **TimeFinder** grammar in figure 2. Due to the space constraint we could not show the whole grammar but interested reader can contact us to obtain a copy of it. This figure shows a few different ways we can expect the date, time and other temporal expressions' formats. Creating an extensive grammar is very important to increase the recall value of **TimeFinder**.

Data Preparation and Training Phase

We measure the values of the parameters (as described in subsection) of all the temporal expressions by using our algorithm **TemporalDataPreparer**. Our algorithm has several passes and it is described here in algorithm 1. **ContentExtractor** is an intelligent HTML to text converter devised by Debnath et. al. (Debnath, Mitra, & Giles 2005), which breaks the whole page into logical blocks, identifies the redundant blocks comparing with other HTML pages from the same source and keep the informative blocks of text. During the first pass, the **TimeFinder** function finds all probable temporal expressions in the article (we call this article as "training article", as it is used to train the classifier), and converts them into ISO 8601 format (as much as possible with unknown fields empty and keeping the other duration/age related expressions as they are). During the second pass, **TemporalDataPreparer** asks the user to identify the **BT** of this training article. It produces all the temporal expressions, generated from the first pass, to the user. The user identifies the correct temporal expression which can be attributed as the **BT** of the article. (We used a regular linux command line interface for the user input submission). During the third pass, **TemporalDataPrepaper** pulls every temporal expression hash key and measures the values of different parameters (described next) by using **MeasureParameterValue** function. This function performs all possible counting of paragraphs, sentences, and word occurrences before and after every temporal expression. This needs a a complete splitting of the article into paragraphs, sentences and words.

Data Parameters

The following set of parameters are used to create data to train the **SV** classifier. We included 15 different parameters to properly characterize the temporal expressions. Most of them fall under distance measures, but some of them can be considered as frequency measures.

- **Paragraph Distance (PD)**: The paragraph distance consists of two parameters – how many paragraphs are there before a time expression or **PDB** and how many paragraphs are there after the time expression or **PDA**.
- **Sentence Distance (SD)**: The sentence distance consists of two parameters – how many sentences are there before

Algorithm 1: TemporalDataPreparer (for Training phase): This algorithm prepares data to train **SV** classifier.

Input : HTML Page H, Parameter Set $\mathcal{P}$
Output : Training Set to train the Support-Vector Classifier
Standard: ISO 8601 standard for date and time
begin
Extract the textual content from the HTML page using our intelligent algorithm, which eliminates the redundant blocks such as navigational links, headers or footers
$X \longleftarrow ContentExtractor(H)$
Pass 1:
$\mathcal{T} \longleftarrow$ TimeFinder(H)
Extract all time expressions using our grammar
Let us assume that the set of all time expressions in this page is $\mathcal{T}$
Pass 2: (User interface)
Ask the user to specify which time-line $t \in \mathcal{T}$ is the **BT**.
Pass 3:
Measuring the $\mathcal{P}$ parameter values for the time expressions which are selected in the first pass.
for *each* $t_i \in \mathcal{T}$ **do**
$P_{t_i} \longleftarrow MeasureParameterValue$(t,H,X);
(P_{t_i} is a data row in the $|\mathcal{T}|X|\mathcal{P}|$ matrix.)
Prepare all parameter values in tabular format and stores them in training datafile.
end

Algorithm 2: MeasureParameterValue (used in both training and testing phase): This function calculates all the parameter values for a temporal expression in an HTML page H.

Input : Time expression t, HTML Page H, Text Page X converted from H
Output : Values of Temporal Data Parameters
begin
Function MeasureParameterValue(t, H, X**)**
We need both X, and H as some of the parameter calculations depend on HTML characters and some will be calculated from the text version of the article.
begin
Takes the content X and breaks it into Paragraphs, Sentences, Words . . . etc.
$\mathcal{P}$ is the Data Parameters as described in Section
for *each parameter* $p \in \mathcal{P}$ **do**
$p_{t_i} \longleftarrow$ value of p in X for t_i;
Push p_{t_i} in P_{t_i}.
return P_{t_i} (A row vector)
end
end

a time expression or **SDB** and how many sentences are there after the time expression or **SDA**.

- **Word Distance (WD)**: The word distance also consists of two parameters – how many words are there before a time expression or **WDB** and how many words are there after the time expression or **WDA**.
- **Reporter Names (RN)**: Reporter names or the names of reporting agencies (**RN**) are also a prime factor in identifying the beginning of a news story and the time. This idea is very similar to the way we understand the beginning of a story. The distance between each time expression and the names of the reporter or reporting agencies in words or characters are stored. In our algorithm, we used a knowledge base of all available reporting agency's names and our algorithm checks the occurrence using some simple regular expression rules and then stores the distance in character from all the temporal expressions to the Reporter names. Sometimes reporter names also come in the middle or end of the article and this may reduce the accuracy, but we believe that with all the other parameters together our approach can find the proper **BT** with high accuracy.
- **Specific Words (SW)**: Specific words (**SW**) are also very important to properly identify the **BT**. Words like "By", "On", etc. has more often been seen near the base timeline's temporal expression compared to other temporal expressions.
- **Specific Symbols (SS)**: In the same way we also consider the occurrences and distances between specific symbols (**SS**) and the time expressions. These symbols include special character-set like "–" or ":" which are also common near the base time expression.
- **Font Face Variation (FFV)**: Font face variation (**FFV**) is also another important factor which can be used to identify the location of **BT**. We see that usually the news publication date is placed close to the headline of the news article and usually the headline is written in different character size or in bold face. The regular text in normal font face follows it. Though things are not always written in the same way (that is why it is a challenging problem), yet there is a correlation between their locations and **BT**. We wanted to exploit this correlation and so we marked the places in the document where a change of font face occurs. Then we calculated distance $D^i \forall i \in |\mathcal{T}|$ where D^i is the shortest distance between t_i (the i^{th} temporal expression) and the marks. So if there are M places where font face changes, $D^i = \min\ (Distance(i,j))$, where $Distance(i,j)$ is character difference between t_i and j^{th} mark.
- **Similarity Measures (SM)**: Similarity measures involve word level similarity between sentences before and after a time expression. The reason behind choosing this parameter is the observation that usually the headline of a news article and the first paragraph just after the **BT** describe the same event, sometimes even using identical words or phrases.

Site	Avg TE	Accuracy
Associated Press	9.57	98.55
Briefing.com	18.54	92.67
BusinessWeek Online	18.91	93.25
Business Wire	13.33	96.78
CBS MarketWatch	21.14	94.01
CCBN	3.48	94.43
Dow Jones Business News	7.62	98.48
EDGAR Online	54.92	92.91
EDGAR Online Financials	2.97	92.00
FT.com	8.00	90.22
First Call Events	4.00	97.01
Forbes Magazine	15.33	98.02
Forbes.com	16.96	98.82
Investor's Business Daily	14.50	96.01
Market Wire	10.47	96.68
Morningstar .com	11.00	92.07
Motley Fool	16.39	93.80
NewsFactor	15.75	91.72
PR Newswire	12.68	95.26
PrimeZone Media Network	8.28	96.22
Reuters	8.89	98.60
SmartMoney.com	21.50	92.11
StarMine	3.06	89.03
TheStreet.com	7.39	70.50
Wall Street Transcript	24.33	95.54
Yahoo	12.30	95.54

Table 1: The columns represent the **Source Web-site**, the **Average number of Temporal Expressions per article**, and the **Accuracy**.

We have chosen the above parameters to mimic how a human being would find out the **BT** of a news article. Some of the parameters alone may not be sufficient in distinguishing the **BT** from other temporal expressions but taking everything into account helped in getting high accuracy as described next.

Testing and Evaluation Phase

We crawled financial news articles from various (here 26) financial Web-sites (shown in the table 1) to build an archive of news articles. These are linked to the corresponding stock symbols in finance pages of Yahoo (fin 2003). From that list, we took 114 stock symbols and their news articles (over 1000 in number). We used half of this set for training and the rest half for testing the classifier. Algorithm 3 shows the testing process.

The table 1 shows the accuracy of this data. From this table we claim that our approach of data preparation and the use of **SV** classifier can give excellent accuracy in finding the **BT** of news articles.

Conclusion

We devised a grammar for temporal expressions, and presented a **SV** learning based approach to find the **BT** of news articles. We claimed our contribution in finding the right set of parameters which can efficiently classify the **BT**. We

Algorithm 3: FindAccuracy (testing phase): This algorithm uses classifier $\mathcal{C}$ to classify the temporal expressions.

Input : Classifier $\mathcal{C}$, HTML Page H

Output : Accuracy

begin

$X \longleftarrow ContentExtractor(H)$

Pass 1:

$\mathcal{T} \longleftarrow$ TimeFinder(H)

Pass 2:

Measure the $\mathcal{P}$ parameter values for all the time expressions extracted in the first pass.

for *each* $t_i \in \mathcal{T}$ **do**

$p_{t_i} \longleftarrow MeasureParameterValue$(t,H,X);

p_{t_i} is a data row in the $|\mathcal{T}|X|\mathcal{P}|$ matrix.

Prepare all parameter values in tabular format and stores them in testing datafile.

Pass 3:

Feed the testing datafile to the classifier $\mathcal{C}$

Find the **BT** and match with the labelled dataset and find the accuracy

end

tested our performance by using news articles from 26 different Web-sites and it proved to be good in finding the **BT** with high accuracy. In future we would like to report on associating the **BT** with referenced temporal expressions and on building a chronology of news events.

References

Allen, J. F. 1984. Towards a general theory of action and time. In *Artificial Intelligence*, 123–154.

Allen, J. F. 1995. Natural language understanding: Discourse structure, tense and aspect. In *Addison-Wesley Chapter 16:5*, 517–533.

Debnath, S.; Mitra, P.; and Giles, C. L. 2005. Automatic extraction of informative blocks from webpages. In *the upcoming proceedings of the Special Track on Web Technologies and Applications in the ACM Symposium of Applied Computing*.

Dorr, B., and Olsen, M. B. 1997. Driving verbal and coompositional lexical aspect for nlp applications. In *In the proceedings of ACL*, 151–158.

Ferro, L.; Mani, I.; Sundheim, B.; and Wilson, G. 2001. Tides temporal annotation guidelines draft - version 1.02. In *MITRE Technical report*.

2003. Yahoo finance page - http://finance.yahoo.com.

Hastie, T.; Tibshirani, R.; and Friedman, J. 2003. *The Elements of Statistical Learning: Data mining, Inference, and Prediction*. Springer Verlog.

Hitzeman, J. 1993. *Temporal Adverbials and the Syntax-Semantics Interface*. University of Rochester, Rochester, New York.

Hwang, C., and Schubert, L. K. 1994. Interpreting tense, aspect, and time adverbials: a compositional, unified approach. In *Proceedings of the 1st International Conference on Temporal Logic*, 238–264.

Kemp, H., and Reyle, U. 1993. *From Discourse to Logic*. Kluwer Academic Publishers.

Koen, D., and Bender, W. 2000. Time frames: Temporal augmentation of the news. In *IBM Systems Journal*, volume 39, 597–616.

Lascarides, A., and Asher, N. 1993. Temporal relations, discourse structure, and commonsense entailment. In *Lingusitics and Philosophy*, 437–494.

Mani, I., and Wilson, G. 2000. Robust temporal processing of news. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, 69–76.

News ml - http://www.newsml.org/iptc/newsml/1.2/.

News industry text format or nitf - http://www.nitf.org.

Passanneau, R. J. 1988. A computational model of the semantics of tense and aspect. In *Computational Linguistics*, 44–60.

Reichenbach, H. 1947. The tenses of verb. In *Elements of Symbolic Logic*, 287–298.

Rdf site summary or rss - http://www.purl.org/rss/1.0/spec.

Sack, W. 1997. The questioning news system. In *Technical Report presented at the MIT media Library*.

Scholkopf, B.; Smola, A.; Williamson, R.; and Bartlett, P. L. 2000. New support vector algorithms. neural computation. In *Neural Computation 12*, 1207–1245.

Scholkopf, B.; Platt, J.; Shawe-Taylor, J.; Smola, A. J.; and Williamson, R. C. 2001. Estimating the support of a high-dimensional distribution. In *Neural Computation 13*, 1443–1471.

Xmlnews - http://www.xmlnews.org/.