Automatic Event Reference Identification

Olivia March
CSSE
University of Melbourne
Victoria 3010
Australia
oliviacm@csse.unimelb.edu.au

Timothy Baldwin
NICTA Victoria Research Laboratories
University of Melbourne
Victoria 3010
Australia
tim@csse.unimelb.edu.au

Abstract

Event reference identification is often treated as a sentence level classification task. However, several different event references can occur within a single sentence. We present a set of experiments involving real world event reference identification at the word level in newspaper and newswire documents, addressing the issue of effective text representation for classification of events using support vector machines. Our final system achieved an F-score of 0.764, significantly exceeding that of our baseline system. Additionally we achieved a marginally higher performance than a more complex comparable system.

1 Introduction

Multi-document summarization seeks to combine the salient information from multiple independent documents into a single coherent summary. The Document Understanding Conference (DUC) shared tasks are a measured attempt to apply multi-document summarization to newswire document sets, to produce a fluent 250 word summary (Dang, 2006). The resultant summaries are then scored and assessed, depending on the specifics of the task.

Newswire and newspaper document sets such as those used in DUC often describe the same series of events across multiple sources, for example, media coverage of an election in Bolivia. Here, a time-line can be a more effective way of structuring a multi-document summary than a traditional paragraph-style summary. More generally, historical and biographical information is frequently presented as a time-line of events, and the ability to produce such time-lines from independent sources is valuable for online content providers such as Wikipedia. The following is an illustration of the utility of time-lines in analysing topics (events highlighted in **bold** to illustrate their importance in time-line generation):

> The **Russian Revolution** in 1917 was **triggered** by a combination of **economic breakdown**, war weariness, and discontent with the autocratic system of government, and it first **brought** a coalition of liberals and moderate socialists to **power**, but their failed policies led to **seizure** of power by the Communist Bolsheviks on October 25.

Existing summarization systems, such as the DUC 07 main task system of Stokes et al. (2007), are effective at general summary tasks but have difficulty with event based summaries. For example, Stokes et al. (2007) performs well on general questions centred on a specified topic such as:

> What attacks have occurred against tourists in Luxor, Egypt?

but not so well on topics requiring a listing of events, such as:

> What have been the most recent significant events in the life and career of actress Angelina Jolie?

In this paper, we identify the most effective representation of text for real-world event reference identification in newspaper and newswire text using support vector machines (SVMs). The work presented here is the basis of an event-based summary system which will in turn form a component of the Stokes et al. (2007) system, where each question will be classified as an event-based question or a general question. The corresponding summary will then be generated by the appropriate sub-system, i.e. the original general-purpose system or a dedicated event-based system.

In creating an event-based time-line, three main tasks need to be considered: (1) how to identify and group references to events in text; (2) how to ground the identified events to a timestamp; and finally (3) how to identify salient events for inclusion in the final fixed-length time-line summary. In this paper we specifically focus on the identification of event references in text.

Often the task of event identification is linked to that of temporal resolution, whereby a system resolves the time or date of the occurrence of an event. Many systems treat events as any sentence containing an event reference and focus efforts on identifying temporal cues in the text. As there can be multiple event references within a sentence we identify events at the word level.

The following example is an illustration of the type of event-based identification we aim to achieve in this paper, relative to a plain-text input sentence (events highlighted in **bold**):

> Amid **reports** that thousands of Iraqi soldiers had **surrendered**, administration aides were also upbeat in private, with one even **talking** of **victory** within a week.

The focus of this paper is an exploration of the most appropriate feature representation to use in a purely statistical approach to word-level event identification in newswire text.

In the remainder of this paper, we review the notion of "event" in the literature, and present an overview of the different approaches to event identification. We then describe our experimental methodology, including the TimeBank corpus. Finally we describe a comparative evaluation.

2 Related Work

2.1 Event Definition

The definition of an event varies in granularity depending on the desired application of event extraction.

The Scenario Template task of the Message Understanding Conference (MUC) (Grishman and Sundheim, 1996) relied on specified domain-dependent templates which define events for the purpose of information extraction. The events that are extracted depend on the templates and documents supplied. MUC consists of domain-specific scenario (topic) based templates that systems are required to fill. An example domain is financial news in the Wall Street Journal, e.g. with the scenario of change in the corporate executive management personnel of an organization (Grishman and Sundheim, 1996).

In the context of Topic Detection and Tracking (TDT) (Fiscus et al., 1998), an event corresponds to an instance of a topic identified at the document level, where the aim is to cluster documents on the same topic rather than individual events within the document. For instance, given an earthquake in the Solomon Islands on August 13, in TDT *earthquakes* would be the topic and this particular earthquake instance would be the event. The task would then be to cluster all documents that refer to this specific earthquake.

The aim of the Automatic Content Extraction (ACE) (Doddington et al., 2004) task is to identify and classify events of predefined semantic types. An event reference may be a noun, verb or phrase, however events are tagged at the sentence level (LDC, 2005). In the above-mentioned documents referring to the Solomon Islands earthquake, the ACE data would have each reference marked as an event, in addition to announcements made by officials and any other words/phrases referring to events within the documents.

Setzer (2001) (and later TimeML (Pustejovsky et al., 2005)) define an event as something that happens with a defined beginning and ending within the scope of a document. By comparison, a state is something that holds between entities without a definite beginning or end. The identification of events is based on verb and noun analysis similar in granularity to that in the ACE task. ACE and TimeML differ primarily on the semantic class labels for the events, and the annotated attributes and entities associated with each event. Where TimeML has seven semantic class labels (Occurrence, State, Reporting, I-Action, I-State, Aspectual and Perception), ACE has five more specific labels (Destruction/Damage, Movement, Creation/Improvement, Transfer of Possession or

Control and Interaction of agents) (LDC, 2005).

For our purposes in this research, we define an event to be something that happens or occurs within the context of the document, similar to Setzer (2001). However, we do not limit events to being denoted by only verbs or nouns. We assume that an event reference is contained within a sentence and does not cross sentence boundaries. Section 3.1 contains a detailed description of how an event is defined in our development data.

2.2 Previous Work in Event Classification

Approaches to event extraction vary from ontology- and frame-based systems to combined rule-based and statistical methods. In this section we describe a number of representative approaches to event extraction from the literature.

The REES system (Aone and Ramos-Santacruz, 2000) uses hand-crafted event and relation ontologies to extract 100 relation and event types, 61 of which are events. It is purported to be extensible, however only so far as the ontology is extended. Events and relations relate to templates with type, person, time, place, etc, as defined in the ontology. The system consists of three parts: a tagging module (containing NameTagger, NPTagger and EventTagger), a rule-based co-reference resolution module, and a template generation module (described as a non-hard-coded approach that uses declarative rules to generate and merge templates automatically to "achieve portability"). MUC style data was used for training and testing. The system achieved a 0.70 F-score over 26 event types.

Jones (2003) applied Hidden Markov Models to the automatic generation of scripts to identify events. HMMs were used to capture correlations of significant events in text: the nodes correspond to events in the text, and the arcs indicate which clauses occur together in text. The context of neighbouring events and textual similarity are used to decide whether or not to group clauses.

As a component of TARSQI, aimed at interpreting entities and events in temporally-based questions, Verhagen and Mani (2005) developed the EVITA event recognition tool. EVITA is used for event recognition in newswire text and analysis of grammatical features, such as tense and aspect. Events are identified using lexical analysis, context analysis of verbs, lexical lookup of adjectival events, and machine learning to determine whether an ambiguous noun is used in an event sense. They combine linguistic and statistical methods to obtain a 74.0% precision and 87.3% recall (Sauir et al., 2005). Time references and events are defined by the TimeML annotation scheme.

Preprocessing of the data was carried out using the Alembic Workbench for part of speech tagging, lematizing and chunking. A shallow parser is used to retrieve event referring expressions which are conveyed by verbs, nouns and adjectives. Verb-based events are identified by lexical lookup and contextual parsing of the verbal chunks. Events denoted by nouns are identified via WordNet and disambiguation with a Bayesian classifier trained on SemCor. Adjectives are tagged as events when they come at the head of a predictive complement, such as:

> A Philippine volcano, **dormant** for six centuries, ... (TimeML, 2006)

EVITA also classifies events semantically, which we do not consider in this paper.

Bethard and Martin (2006) describe a system to identify events and their semantic class for the purposes of question answering. They define events as a description of situations that involve some internal structure, as opposed to states which describe a situation that is static or unchanging. Bethard and Martin (2006) view event identification as a classification task. Their system STEP is able to identify events with a precision of 82% and recall of 71%. They use a suite of syntactic and semantic features as input to YamCha — a general purpose chunker — and TinySVM. Each word in the document is classified as either beginning (B), inside (I) or outside (O) an event.

3 Experimental Methodology

3.1 Data: TimeML and TimeBank

Given that our notion of event closely mirrors that of TimeBank, we were able to use the TimeBank 1.1 corpus for system development and evaluation. TimeBank 1.1 consists of 186 news articles marked up with TimeML standard 1.1. The articles are sourced primarily from the Wall Street

Journal and Associated Press newswire articles. They contain EVENT tags denoting events at the phrase and word level.

TimeML is a specification language for event and temporal expressions in text (Pustejovsky et al., 2005), and builds on TIDES TIMEX2 and the temporal annotation language presented in Setzer (2001).

TimeML addresses four basic problems in event-time identification:

- Time stamping of events (anchor event in time).
- Chronological ordering of events with respect to one another.
- Reasoning with contextually under-specified temporal expressions (e.g. *last week*).
- Reasoning about the persistence of events (how long does an event or outcome of an event last).

TimeML has four major event-time structures: EVENT (see Section 3.1.1), TIMEX3 (used to denote explicit time references), SIGNAL (used primarily to annotate function words indicating how temporal objects relate to one another) and LINK (denoting the relationship between an event and other events or temporal expressions).

3.1.1 EVENT

Events are a cover term for situations that happen or occur, and can be punctual or last for a time period. Events are generally expressed by tensed or untensed verbs, nominalisations, adjectives, predicative clauses or prepositional phrases. An EVENT in TimeML contains several attributes: a unique identifier, a class, tense, aspect and optionally another event. It may also contain another event or time to which it is temporally related.

There are seven event class types (each of which is illustrated with a set of examples):

- *Occurrence:* die, crash, build, merger, sell
- *State:* on board, kidnaped, in love
- *Reporting:* say, report, announce
- *I-Action:* attempt, try, promise, offer
- *I-State:* believe, intend, want
- *Aspectual:* begin, finish, stop, continue
- *Perception:* see, hear, watch, feel

We currently disregard the event classes and use a binary classification, identifying each word as an event or non-event.

3.2 System Description

We preprocessed the TimeBank data to remove extraneous XML tags, decompose it into sentences and words, and discard punctuation. Finally we assign each word with a POS tag using the Stanford POS tagger (Toutanova et al., 2003).

In our experiments, we use three classification algorithms: decision trees, naive Bayes and support vector machines. The chosen training and evaluation platform was WEKA (Witten and Frank, 2005). Stratified 10-fold cross-validation was used throughout, where the labeled text fragments were randomly allocated to different combinations of training and testing splits. We report on the average performance over the 10 runs.

Preliminary experiments over the three classification algorithms indicated that SVMs were more effective in the classification of event references in text. BSVM (Hsu and Lin, 2002) was used for subsequent experiments, again using 10-fold cross-validation.

In Section 3.3, we present a range of different feature types, designed to explore the impact of different information sources on event reference classification.

In our classifiers, we look first at sentence-level event classification (i.e. does a given sentence contain one or more event references) and then word-level event classification (i.e. is a given word an event reference). The sentence-level classifier is used to filter out the non-event sentences, and only event-containing sentences are passed on to the word-level classifier.

3.3 Features

We use a basic word feature vector to represent the words being classified. Each distinct word in the input text corresponds to a unique feature. The text is transformed into a vector of N values where N is the number of distinct words in the TimeBank corpus. The following are common text representation techniques in text classification, the effects of which we explore in the context of event identification.

- **Context Window**
  For identification of events at the word level we explore the use of a context window of the zero to three preceding words, in addition to the word being classified.

- **Feature Representation**
  We also look at the effect of representing the current and context window words as a bag of word types (binary representation), or list of position-specified words (representing each word by its positional value [current word = 1, previous word = 2, etc; other words = 0]).

- **Stop Words**
  Stop words are short, frequently occurring words such as *and*, *or*, *in*, *of*. Stop word removal reduces the feature space with generally little impact on classification performance in document categorisation tasks, but has been shown to be detrimental in sentence-level classification tasks (Khoo et al., 2006). In our research, we use the stop word list of van Rijsbergen (1979).

- **POS**
  Parts of speech (POSs) are assigned to each word using the Stanford maxent POS tagger, and optionally used as an alternative to word features for context words (i.e. each context word is represented as its POS).

- **Feature Generalisation**
  We optionally group numbers, times and named entities into single features. Numbers and dollar amounts were identified using regular expressions, and named entities were tagged using the Alembic Workbench (Aberdeen et al., 1995).

3.4 Evaluation Method

We use F-score as our primary evaluation metric, as defined below relative to Precision and Recall:

$$\text{Precision} = \frac{\textit{\# words correctly classified as events}}{\textit{\# words classified as events}} \quad (1)$$

Model	Precision	Recall	F-Score
Baseline	0.800	0.550	0.652
STEP	0.820	0.706	0.759
EVITA	0.740	0.873	0.801

Table 1: Performance of comparable event identification systems and our baseline system, as evaluated over the TimeBank data

$$\text{Recall} = \frac{\textit{\# words correctly classified as events}}{\textit{\# words that are events}} \quad (2)$$

$$\text{F-Score} = \frac{2 \times \textit{Precision} \times \textit{Recall}}{\textit{Precision} + \textit{Recall}} \quad (3)$$

Our baseline takes the form of a unigram tagger, where each word is labeled with its most frequent label. Table 1 shows the performance of the baseline system, in addition to the the comparable STEP (Aone and Ramos-Santacruz, 2000) and EVITA (Sauir et al., 2005) results — as described in Section 2.2 — both of which use the TimeBank data in the development and testing of their systems.

4 Results

4.1 Sentence Level Classification of Events

In this section we first present the results of our sentence-level event identification system, where each sentence is classified as containing one or more event references, or alternatively containing no event references. Three learners — C4.5 decision tree, SVM and naive Bayes — were run over a random subset of the TimeBank data, the results of which are shown in Table 2.

All three methods performed exceptionally well, at an F-score of or close to 1. This level of performance is due to 82% of the sentences in the corpus containing an event reference, and most of these containing more than one event reference. Therefore only one of the many event references within a sentence needs to be identified for the sentence to be correctly classified.

The naive Bayes classifier and decision tree models proved difficult to run over the entire TimeBank corpus due to resource constraints. As a result, we chose to use BSVM (Hsu and Lin, 2002) for all subsequent sentence-level experiments.

Algorithm	Precision	Recall	F-Score
Naive Bayes	0.995	0.998	0.996
SVM	1.000	1.000	1.000
C4.5	1.000	1.000	1.000

Table 2: Performance of sentence level classification with stop word removal and feature generalisation

As there can be multiple references to different events within a single sentence, sentence level classification is not fine-grained enough for event clustering and extraction in the context of event-based summary generation. The sentence classifier is instead used as a pre-filter for word-level event classification, as discussed in the following section.

4.2 Word-level Classification of Events

As stated above, in classifying individual words for event reference, we first use the sentence-level classifier to remove sentences that do not contain events. This proved to have no effect on the accuracy of the word level system, but did offer a slight reduction in the time required by the word-level classifier.

Both the naive Bayes classifier and C4.5 decision tree resulted in majority class classifiers (see Table 1), where all words were tagged as non-events. The SVM, on the other hand, produced a variety of results, and thus forms the basis of all results presented in this section.

Below, we will discuss our findings into the effects of stop word removal, feature representation, context window size, POS tagging and feature generalisation on word-level event classification.

4.2.1 Context Window Size

Increasing the context window, using a binary feature representation, decreased the accuracy of the classifier, as shown in the leftmost segment of Figure 1. The zero word context improves only 1% with an increase from 15 to 18 thousand training instances compared with an improvement of 7% for the three word prior context model. At this point, we have been unable to improve over the baseline system.

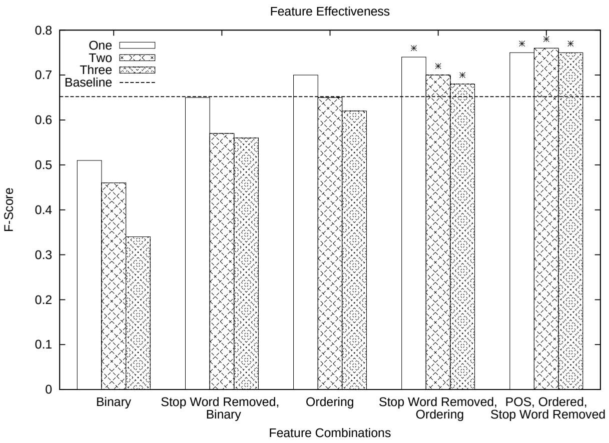

Figure 1: The F-score of models trained over different feature combinations. (* = statistically significant difference: two-tailed paired t-test, $p < 0.05$)

4.2.2 Feature Representation

Replacing the binary bag of words representation with a position-specified binary word representation has a dramatic impact on our results, as indicated in the middle set of results in Figure 1. Maintaining the ordering of 2 word prior context has an F-score of 0.68, compared to a binary representation of 2 word prior context with an F-score of 0.41.

4.2.3 Stop Words

Khoo et al. (2006) found that stop word removal had a detrimental effect on classification at the sentence level. Our results, however, demonstrate that stop word removal offers a slight increase in the accuracy of the word-level classifiers. Table 3 shows the results obtained using SVM to identify event references at the word level using different combinations of text representation techniques. As we can see from the table, stop word removal had a positive effect on the system irrespective of whether a binary or word order representation was used. With a one word POS context window and feature generalisation, stop word removal increases the F-score by 0.043. Similarly, using a binary representation and grouping, stop word removal increases the F-score by 0.069.

4.2.4 Part Of Speech

Including the POS tags of the context and word being classified increases performance of

Feature combination	Precision	Recall	F-score
Word order and grouping	0.712	0.675	0.693
Word order, grouping and stop word removal	0.761	0.712	0.736
Binary and grouping	0.623	0.462	0.531
Binary, grouping and stop word removal	0.673	0.542	0.600

Table 3: The results of word-level event identification using an SVM and different combinations of features

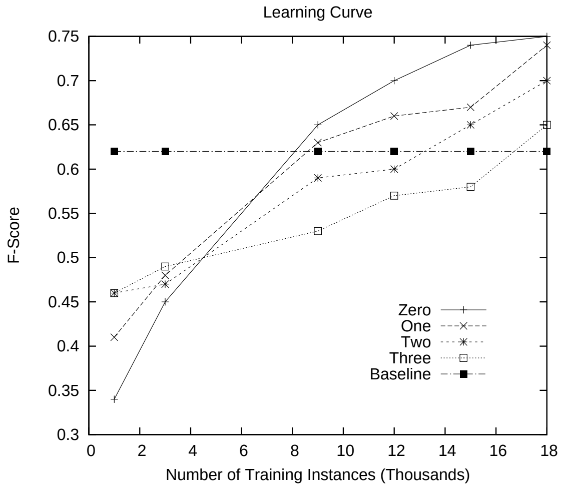

Figure 2: The learning curves for the models of different context window sizes, with feature generalisation and stop word removal

the overall system. However, the greatest increase comes from the inclusion of the preceding context represented as a part of speech, without context words. This facilitates the generalisation of the patterns leading up to the event reference. Note that the word being tagged is retained as a binary feature, as its exclusion would leave a POS pattern feature that is too general given the limited combination of less than fifty distinct POS tags.

4.2.5 Feature Generalisation

Grouping specific types of words such as numbers, named entities and time references into a single feature proved an effective way of reducing the feature space. However, closer analysis of the word-level event classification results showed that some of the terms being grouped together occurred as events whereas others did not. There are 610 NUMBER references in the TimeBank data. 247 of these are an event or part thereof and 363 are not, leading to most NUMBER references being tagged as non-events. For example, monetary amounts can be a state (type of event) if they are a milestone for a company, or alternatively they can simply be an amount lost by the company.

> The insurer's earnings from commercial property lines **fell** 59% in the latest quarter, while it **lost** $7.2 million in its personal property business, compared with earnings of **$6.1 million**[1]a year ago.

The other groupings used, TIME and NAMED ENTITY, are never event references in the TimeBank corpus, and thus their grouping reduces the feature space without loss of the ability to disambiguate. Further investigation into a more fine-grained grouping of numerical terms is required before grouping numbers is included in our final system.

4.3 Discussion

Sentence-level classification proved a simple task with an almost perfect classification accuracy. The use of a sentence-level classifier for prior filtering of the word-level event system had no effect of the overall F-score.

Our final system combined POS representation of the two word prior context with context positional information and stop word removal to achieve an F-score of 0.764, exceeding our baseline of 0.653 and marginally exceeding the performance of the comparable STEP system with an F-score of 0.759. Over 50% of incorrectly classified words resulted from event words which occur only once in the TimeBank corpus. A further 20% are words that appear as both events and non-events in the training data. 7% of the remaining errors result from incorrect tokenisation dur-

[1] States that describe circumstances in which something obtains and holds true are called predictive states. The validity of this is dependent on the document creation time, and includes some quantitative statements such as those that appear in financial journals (TimeML, 2006)[pg14].

ing preprocessing. There are 1540 words in the TimeBank data that occur only once as events.

To investigate the impact of the amount of data on our results, we generate learning curves for word order-sensitive feature representation with stop word removal and feature generalisation, over diminishing amounts of the training data used in cross-validation. The curves are presented in Figure 2 over different context word window sizes. As we can see, the models based on larger context windows are still climbing steeply over the final increment of training instances, while the zero-word context model appears to be flattening out. As such, we would expect that an increase in the number of training instances to improve the performance of the models with larger context windows, although it remains to be seen how much they would improve relative to the smaller word windows.

Maintaining word order has the biggest effect on the performance of the classification system. Representing the preceding context as POS rather than words also increases the accuracy of the overall system and, contrary to expectations, the removal of stop words marginally increased the performance. As indicated by the learning curve (Figure 2), increasing the number of training instances has the potential to improve the performance of the models which include context.

The two-word POS context model misclassifies a different subset of words to the three-word POS context model. Therefore, combining different models may increase the performance of the overall system.

The majority of errors came from previously unseen words in the training data. Future work will thus focus on how the system can better handle unseen words.

5 Future Work

We intend to look at the effect verb types obtained from WordNet (Fellbaum, 1998) or VERBOCEAN (Chklovski and Pantel, 2004) have on word-level event identification, to assist in better handling of unseen words. We will also look at how the system performance scales over a larger data set.

6 Conclusion

We presented a set of experiments involving real-world event reference identification at the word level in newspaper and newswire documents. We addressed the issue of effective text representation for classification of events using support vector machines. Our final system combined two word POS prior context, with positional information and stop word removal. It achieved an F-score of 0.764, significantly exceeding that of our baseline system with an F-score of 0.653.

Acknowledgments

NICTA is funded by the Australian government as represented by Department of Broadband, Communication and Digital Economy, and the Australian Research Council through the ICT Centre of Excellence programme.

References

John Aberdeen, John D. Burger, David S. Day, Lynette Hirschman, Robinson, Patricia, and Marc B. Vilain. 1995. MITRE: Description of the Alembic system used for MUC-6. In *Proceedings of MUC-6*.

Chinatsu Aone and Mila Ramos-Santacruz. 2000. REES: A large-scale relation and event extraction system. In *Proceedings of the 6th Applied Natural Language Processing Conference*, Seattle, USA.

Steven Bethard and James Martin. 2006. Identification of event mentions and their semantic class. In *Proceedings of the 2006 Conference on Emprical Methods in Natural Language Processing*, Sydney, Australia.

Timothy Chklovski and Patrick Pantel. 2004. VERBOCEAN: Mining the web for fine-grained semantic verb relations. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-04)*, Barcelona, Spain.

Hoa Trang Dang. 2006. Overview of DUC 2006. In *Proceedings of the Document Understanding Conference Workshop*, New York, USA.

George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. 2004. Automatic content extraction (ACE) program - task definitions and performance measures. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation*. Lisbon, Portugal.

Christiane Fellbaum. 1998. *WordNet An Electronic Lexical Database*. The MIT Press, first edition.

Jon Fiscus, George Doddington, John Garofolo, and Alvin Martin. 1998. NIST's 1998 topic detection and tracking evaluation. In *Proceedings of the DARPA Broadcast News Workshop*.

Ralph Grishman and Beth Sundheim. 1996. Message Understanding Conference 6: A brief history. In *Proceedings of the 16th Conference on Computational Linguistics*, Copenhagen, Denmark.

Chih-Wei Hsu and Chih-Jen Lin. 2002. A comparison on methods for multi-class support vector machines. *IEEE Transactions on Neural Networks*, 13:415–425.

Dominic R. Jones. 2003. Identifying events using similarity and context. In *Proceedings of the Seventh Conference on Natural Language Learning*, Edmonton, Canada.

Anthony Khoo, Yuval Marom, and David Albrecht. 2006. Experiments with sentence classification. In *Proceedings of the 2006 Australasian Language Technology Workshop*, Sydney, Australia.

LDC. 2005. ACE (automatic content extraction) English annotation guidelines for events.

James Pustejovsky, Robert Ingria, Roser Sauri, Jose Castano, Jessica Littman, Rob Gaizauskas, Andrea Setzer, Graham Katz, and Inderjeet Mani. 2005. The specification language TimeML. In *The Language of Time: A Reader*, chapter 27. Oxford University Press.

Roser Sauir, Robert Knippen, Marc Verhagen, and James Pustejovsky. 2005. EVITA: A robust event recognizer for QA systems. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 700–707, Vancouver, Canada.

Andrea Setzer. 2001. *Temporal Information In Newswire Articles: An Annotation Scheme and Corpus Study*. Ph.D. thesis, University of Sheffield.

Nicola Stokes, Jiawen Rong, and Lawrence Cavedon. 2007. NICTA's update and question-based summarisation systems at DUC 2007. In *Proceedings of the Document Understanding Conference Workshop*, Rochester, USA.

TimeML. 2006. TimeML annotation guidelines version 1.2.1.

Kristina Toutanova, Dan Klein, Christopher Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of HLT-NAACL 2003*, pages 252–259, Edmonton, Canada.

C.J. Keith van Rijsbergen. 1979. *Information Retrieval*. Butterworth-Heinemann, 2nd edition.

Marc Verhagen and Inderjeet Mani. 2005. Automating temporal annotaion with TARSQI. In *Proceedings of the ACL 2005 Interactive Poster and Demonstration Sessions*, Ann Arbor, USA.

Ian H. Witten and Eibe Frank. 2005. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, San Fransisco, second edition.