Information Retrieval, 7, 347–368, 2004

Simple Semantics in Topic Detection and Tracking

JUHA MAKKONEN juha.makkonen@cs.helsinki.fi
HELENA AHONEN-MYKA helena.ahonen-myka@cs.helsinki.fi
MARKO SALMENKIVI marko.salmenkivi@cs.helsinki.fi
Department of Computer Science, P.O. Box 26 (Teollisuuskatu 23), FIN-00014, University of Helsinki, Finland

Received May 15, 2003; Revised July 24, 2003; Accepted September 11, 2003

Abstract. Topic Detection and Tracking (TDT) is a research initiative that aims at techniques to organize news documents in terms of news events. We propose a method that incorporates simple semantics into TDT by splitting the term space into groups of terms that have the meaning of the same type. Such a group can be associated with an external ontology. This ontology is used to determine the similarity of two terms in the given group. We extract proper names, locations, temporal expressions and normal terms into distinct sub-vectors of the document representation. Measuring the similarity of two documents is conducted by comparing a pair of their corresponding sub-vectors at a time. We use a simple perceptron to optimize the relative emphasis of each semantic class in the tracking and detection decisions. The results suggest that the spatial and the temporal similarity measures need to be improved. Especially the vagueness of spatial and temporal terms needs to be addressed.

Keywords: topic detection and tracking, retrieval model, information extraction, temporal expression, geographical ontology

1. Introduction

Topic detection and tracking (TDT) is a research initiative concerned with techniques to organize news documents. In contrast to the more traditional information retrieval problems, the focus in TDT is on news events: in breaking the text into cohesive stories, spotting something previously unreported, tracing the development of the event, and grouping together news that discuss the same event. The problem area has also been called *event-based information organization* (Allan 2002a). The user of this kind of system could be, for example, an information worker, a specialist or a reporter who needs to keep up with several sources of news: radio and television broadcasts and on-line Internet news. The user might wish to follow the course of events regarding forest fires in Portugal, the development of the presidential elections in Lithuania, or just be informed if anything new takes place in Africa or in the metal industry, for example.

Focusing on news events influences the nature of the retrieval problems. A TDT system runs on-line and has absolutely no knowledge of the coming events in advance. This makes the application of machine learning methods difficult. Often, an event involves only a few documents that appear within a short period of time, and because the events evolve, the essential vocabulary describing an event may change considerably in short time. Detecting when a story discusses a news event for the first time has been a particularly difficult task. Spotting something 'new' requires a highly effective representation of 'old', i.e., what has already been seen. The results by Allan et al. (2000) suggest that it is highly unlikely that a technology based on a full-text similarity will yield such a representation.

We present an approach for TDT that employs *semantic classes*, i.e., groups consisting terms that have similar kind of meaning: locations, proper names, temporal expressions and general terms. This split in term-space enables us to use class-wise similarity measures. By mapping the terms of a given class onto an ontology, we are able to establish a semantical similarity between the terms instead of using just a binary string matching. Especially with intensional data, such as temporal expressions, this kind of mapping is needed in order to have any means of comparison.

The comparison of two documents is carried out class-wise: the names in the one document are compared to the names in other, the locations in one against the locations in the other, and so on. As a result, we have vector of similarity values that we turn into a single yes/no decision by a weighted sum. The weights are optimized with a linear perceptron.

This paper is organized as follows. Section 2 gives a concise view of topic detection and tracking by presenting event and topic definitions, problems inherent to TDT and the previous work. In Section 3 we bring forward a document representation using semantic classes, and present methods for comparing documents class-wise, and particularly comparing the spatial and the temporal information. Section 4 describes the TDT algorithms that utilize perceptrons. Section 5 illustrates our experiments with the TDT2 corpus and Section 6 is a conclusion.

2. Topic detection and tracking

The news-stream in topic detection and tracking is a compilation of on-line news and transcribed radio and TV broadcasts from one or more sources and possibly in one or more languages. The topic detection and tracking is considered to comprise five tasks (Allan 2002a): (1) *topic tracking* monitors news stream for stories discussing given target topic, (2) *first story detection* (FSD, also *new event detection*) makes binary judgment on each document whether it discusses a new, previously unreported topic or not, (3) *topic detection* (also *cluster detection* or just *detection*) forms topic-based clusters of documents, (4) *link detection* determines whether two given documents are about the same topic, and (5) *story segmentation* finds the boundaries for cohesive text fragments.

Some of these tasks have been approached with traditional information retrieval techniques. For example, the topic tracking can be understood as an information filtering task in which the system is given a small number of sample documents and is expected to spot all further documents discussing the topic of the samples. The topic detection involves text clustering. The story segmentation shares its rationale and motivation with text and discourse segmentation. Nevertheless, the setting of TDT presents unusual problems that complicate the use of traditional techniques. The topics are unknown in beforehand. The systems run on-line and can make very few assumptions on the incoming data. The topics often involve only a small number of documents that are encountered in a burst, and the essential vocabulary describing a topic may change drastically in a short time.

In Section 2.1 we first present the definition of an event and a topic. Then, in Section 2.2 we make an effort in formalizing some of the problems in TDT. The previous approaches that have tried to overcome these problems are presented in Section 2.3.

2.1. Topic definition

There are events taking place in the world, and some of them are acknowledged in the news. Naturally, a TDT system does not perceive the events themselves, but rather makes an effort in deducing them from the news-stream. News are written by humans and published and distributed by agencies and companies. Although the concept of *event* may seem intuitively clear and self-explanatory, formulating a sound definition is difficult. Predating TDT research, numerous historians and researchers of political science have wrestled with the definitions (see e.g., Falk 1989, Gerner et al. 1994). What seems to be somewhat agreed upon is that an event is some form of activity carried out by some agent or agents somewhere at some time. In topic detection and tracking, an event is defined as follows (Allan et al. 1998a):

Definition 1. An event is a unique thing that happens at some specific time and place.

This definition is intuitively quite sound. There are, however, events of different scales, and this definition seems to neglect events which either have a long-lasting nature (Intifada, Kosovo-Macedonia, struggle in Columbia), escalate to several directions (September 11, war in Iraq), or are not tightly spatio-temporally constrained (global warming, SARS- and BSE-epidemics). Some of these problematic events would classify as *activities* (Papka 1999), but when encountering a piece of news, we do not know *a priori* whether it is a short term event or long term activity, just a simple incident or the start for a complex chain of actions.

A *topic* is considered a set of documents that relate strongly to each other via a *seminal event*, an event that triggers the topic. For example, the first story reporting a victim of a deadly virus produces a new topic, and any further stories on the development and spreading of the virus, the issued quarantines, the quest for a remedy, the economical effects, for example, are part of this topic. The *de facto* definition of topic along which the ‘official’ TDT2 and TDT3 corpora are produced is as follows (Cieri et al. 2002):

Definition 2. A topic is an event or an activity, along with all related events and activities.

The terms ‘event’ and ‘topic’ have been used interchangeably in TDT for historical reasons, which makes the distinction slightly difficult. Initially, the focus was confined on identifying news events, and later the scope was broadened to involve topics as defined above. However, the topics are defined in terms of events.

2.2. Problems in event-based information retrieval

The different TDT tasks can all be considered to be some sort of detection: given an input and a hypothesis about the data, a TDT system makes a decision, whether that hypothesis holds (Fiscus and Doddington 2002). In information retrieval there are plenty of detection tasks, but in TDT the tasks are intimately related to time. The nature of TDT detection tasks is probably best portrayed by comparing them to another tasks of information retrieval, text categorization, for example.

Automatic text categorization is usually carried out using some machine learning system (see e.g., Sebastiani 2002, Yang and Liu 1999). Such a system is taught to recognize the difference between two or more predefined classes or categories by providing a number of pre-labeled samples to learn from. As to classes and word frequencies, this training material is assumed to lend itself to the same underlying distribution as the material that is to be categorized. More formally, the documents $D = \{d_1, d_2, \ldots, d_{|D|}\}$ and their labels $C = \{c_1, c_2, \ldots, c_{|C|}\}$ are seen to be governed by an unknown distribution. This distribution is expressed as a function $\check{h}$ that assigns to each document-label pair $\{\langle d_i, c_j \rangle \in D \times C \mid 1 \leq i \leq |D|, 1 \leq j \leq |C|\}$ a boolean value indicating their relevance, i.e.,

$$\check{h} : D \times C \rightarrow \{-1, 1\}.$$

The task of classification is to come up with a hypothesis

$$h : D \times C \rightarrow \{-1, 1\}$$

that represents $\check{h}$, practically, with the 'highest' accuracy. This accuracy is evaluated with a pre-labeled testing material that contains the same classes C as the training material.

Now, with TDT the problem is different. Let us assume that the documents and events yield to an unknown distribution represented by the function

$$\check{g} : D \times E \rightarrow \{-1, 1\}$$

that assigns each document $d_i \in D$ a boolean value indicating whether it discusses event $e_j \in E$ or not. The problem is that domain of $E = \{e_1, e_2, \ldots, e_{|E|}\}$ is time-dependent. This means that the hypothesis

$$g : D \times E \rightarrow \{-1, 1\}$$

cannot be evaluated similarly to text categorization, because the test set does not contain the same events as the testing set. In fact, we have no *a priori* knowledge of the domain E. What we are left with is a pair-wise similarity of documents. By examining the pair-wise comparisons in the training set, we can formulate a hypothesis

$$k : D \times D \rightarrow \{-1, 1\}$$

that assigns the pair $\langle d_i, d_j \rangle \in D \times D$ a boolean value 1, if the documents discuss the same event, -1 otherwise. Any two documents about the same event are (ideally) *similar in a similar way*. This somewhat trivial observation has some implications worth mentioning.

Firstly, the detection and tracking are based on pair-wise comparisons of documents, which requires exhaustive computation. An event can be represented by a centroid vector or by some compilation of on-topic documents, but the number of one-document events is very high, as unlabeled documents are considered singleton events. The detection and tracking algorithms have to take these into account and make sure, they do not miss anything relevant.

Secondly, the system's knowledge of events E relies on the first-story detection. FSD has been characterized as '*queryless retrieval*' as we do not know what to look for exactly. 'New' is something unexpected that is sufficiently different from the old. Hence, one has to run tracking with just one document in order to determine its novelty, and if none of the old documents match to the new one, the document is considered a first-story. Allan et al. (2000) showed that the performance of this kind of tracking-based FSD is highly unlikely to attain a reasonably effective level, if tracking relies on full-text similarity.

In addition, as the events evolve the vocabulary of the relevant documents can change quite considerably over time. For instance, there is no mentioning of Timothy McVeigh until the 61st document reporting the Oklahoma City bombing, but then he was arrested, convicted and later executed (Allan et al. 1998c). This change in the vocabulary complicates the construction of event representation as it would have to address this degradation. The *event evolution* is one of the major challenges of TDT.

2.3. *Related work*

There have been many attempts to overcome the problems discussed in the previous section. The methods applied in TDT cover a good portion of the prevailing techniques information extraction, retrieval and filtering, text clustering and text categorization and natural language processing. Fundamentally, a TDT system runs on-line and does not have any knowledge of the unseen documents, which makes it a case for clustering. There has been, however, research on retrospective topic detection and tracking, where the system is shown all of the data at once (Allan et al. 1998a, Yang et al. 1999), but the focus is mainly on the on-line setting.

Allan et al. (1998c) considered each incoming document a query that was made on the previous documents. If the returned answer was not similar enough, the story was considered a first-story. The terms were weighted with a modification of *TFIDF* and by its *surprisingness*. A term was seen as surprising if it had not occurred recently. The need for exhaustive pair-wise comparisons of documents was avoided by the use of an inverted index. Motivated by this computational efficiency, we employ the inverted index as well.

Yang et al. (1999) employed a group-average and a single-pass clustering in topic detection. Typically, the group-average clustering (GAC) iterates over the data, starting from one-document singletons and at each step merging the closest clusters until given number of clusters is found or the clusters are too distant to be merged. The computational complexity of GAC is typically quadratic to the number of documents. The efficiency was increased by dividing the clusters into evenly-sized buckets (Cutting et al. 1992) and applying GAC locally to a bucket before removing bucket boundaries. The single-pass approach constructs clusters in one go. A document is associated with an event, if the similarity exceeds a pre-set threshold. This approach was accompanied by a time penalty: events tend to be temporally proximate and thus older documents are less likely to discuss the same event.

In topic tracking Yang et al. (1999, 2000) have favored k-nearest neighbor, kNN, with some modifications. The advantage of kNN over many other classifiers is that being instance-based it makes fewer assumptions on the data. The centroid (or centroids) representing a topic can easily be changed, unlike in Rocchio, for instance. However, Rocchio has been combined with topic-type categories that on one hand reduced the number of computation as the search-space was confined to the topic-type and on the other hand enabled a topic-type-based term-weighting Yang et al. (2002b). Yang et al. (2002a) have also experimented a multi-strategy approach that combines Rocchio, kNN and language modeling into a TDT system with some success.

Since the analysis of the performance of first-story detection by Allan et al. (2000), there has been more work on utilizing information external to the system or produced by natural language processing techniques. Carthy (2002), for instance, used WordNet (Miller 1995) in building lexical chains, i.e., sequences of related words in the text, which were then used in combination with keywords in topic tracking. Intuitively, it would seem that proper names and other named entities (NE) would benefit the differentiating of events. Both Allan et al. (1999) and Yang et al. (2002b) extracted seven types of NE's for the purposes of FSD: locations, names of individuals and organizations, time and date references, and sums of money and percentages. Pons et al. (2002) used temporal references in building a hierarchy of topics and events. Makkonen et al. (2002) split the term space into four classes of semantically similar words, names, locations, temporal expression and general terms, and conducted the comparisons of documents class-wise. Later, this comparison was augmented with two class-based ontologies: time-axis and geographical taxonomy (Makkonen et al. 2003). We develop this approach further by applying the techniques on English and introducing a proper optimization for the weights of the semantic classes.

3. Enhanced document representation

It has been difficult to detect two distinct train accidents and bombings, for instance, as different events (Allan et al. 1998a). The terms occurring in the two documents are so similar that a term space and a weighting-scheme in use fail to represent the required distinction. Allan et al. (1998b) suspect that only a small number of terms is adequate to make the distinction between different news events. The problem is, of course, to know which ones. Intuitively speaking, when news report two different train accidents, intuitively it would seem that the location and the time, possibly some names of people, are the terms that make up the difference. Papka observes that when increasing the weights of noun phrases and dates, the classification accuracy improves and when decreasing them, the accuracy declines (Papka 1999).

Our approach is based on this observation. We extract four types of terms: locations, temporal expressions, names and general terms. The exploitation of named entities is by no means a novel technique (see e.g., Allan et al. 1999, Yang et al. 2002b), but since we store the terms in distinct vectors and conduct the comparison of two documents vector-wise, we are able to assign each term-type a similarity measure. Because all the temporal expressions, for example, have a relation to all other temporal expressions via this dedicated similarity measure, we are in fact introducing simple semantics.

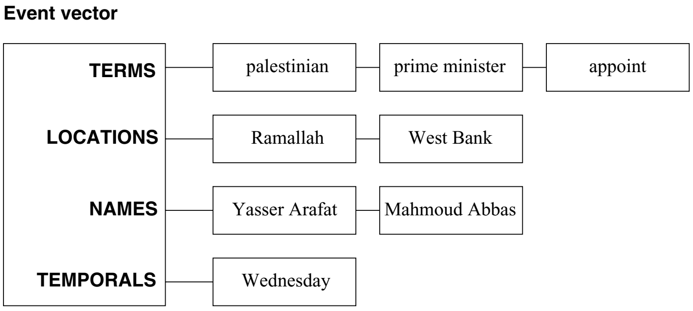

Figure 1. An example of an event vector. "*RAMALLAH, West Bank—Palestinian leader Yasser Arafat appointed his longtime deputy Mahmoud Abbas as prime minister Wednesday*, ..." (AP: Wednesday, March 19, 2003).

3.1. Event vector

A news document reporting an event states at the very barest *what* happened, *where* it happened, *when* it happened, and *who* was involved. The automatic extraction of these facts for natural language understanding can be troublesome and time-consuming. Many of the previous detection and tracking approaches have to encapsulated all the content in a single vector. To be able to differentiate topics of the same type while maintaining robustness, we assign each of the questions a *semantic class*, i.e., groups of semantically related words, as we have previously reported (Makkonen et al. 2002). The semantic class of LOCATIONS contains all the places mentioned in the document, and thus gives an idea, where the event took place. Similarly, TEMPORALS, i.e., the temporal expressions denote a point or an interval of time, and bind the text onto the time-axis. NAMES are proper noun phrases that represent the people or organizations involved in the news story. What happened is represented by 'normal' words which we call TERMS.

The representation of the document using semantic classes is illustrated in figure 1. This *event vector* comprises four sub-vectors that reside in distinct spaces due to the semantical dissimilarity. If two documents coincide as to temporal expressions and locations, for example, it would serve as an evidence for them to discuss the same event. Of course, the on-line news stream reports events as they are fresh, and thus the temporal similarity would be quite high for the news published on the same day.

3.2. Comparing event vectors

By dividing the term space into semantic classes we are able to compare the documents *class-wise*. This means that we examine the corresponding sub-vectors of two event representations at a time: LOCATIONS in the one document against the LOCATIONS in the other, NAMES in the first document against NAMES in the other and so on. Within each class we can choose the measure of similarity independent of another class. For example, the similarity of two LOCATION terms can be based on a geographical proximity and thus the terms *London* and *Thames* would be highly relevant. Similarly, the utterances "*next week*" and "*the last* *week of March 2003*" differ on the surface, but when evaluated with respect to the utterance time they could denote the same temporal interval.

Thus, instead of having just a binary similarity based on a string matching, we can map the terms onto an ontology, where their relation can be more fine-grained than match–mismatch. The result of class-wise comparisons is a vector $\mathbf{v} = (v_1, v_2, v_3, v_4) \in \mathbb{R}^4$ comprising the class-based similarities.

In the following, we present the class-wise similarity functions: the common TERMS and NAMES in Section 3.2.1, TEMPORALS and LOCATIONS in Sections 3.2.2 and 3.2.3, respectively.

3.2.1. General similarity. Naturally, all terms are not equally informative. In determining the of TERMS and NAMES we use the *term-frequency inverted document frequency*, TFIDF (Salton and Buckley 1988). Let $T = \{t_1, t_2, \ldots, t_n\}$ denote the terms and $D = \{d_1, d_2, \ldots, d_m\}$ documents. Thus, the weight is determined by a function $w : T \times D \to \mathbb{R}$ such that

$$w(t, d) = f(t, d) \cdot \log\left(\frac{|D|}{g(t)}\right), \tag{1}$$

where function $f : T \times D \to \mathbb{N}$ represents the number of occurrences of term t in document d, $|D|$ is the total number of documents, and function $g : T \to \mathbb{N}$ is the number of documents in which term t occurs, i.e., the document frequency of term t.

The similarity σ of two sub-vectors X_k and Y_k of semantic class k is based on the cosine of the two,

$$\sigma(X_k, Y_k) = \frac{\sum_{i=1}^{|k|} w(t_i, X_k) \cdot w(t_i, Y_k)}{\sqrt{\sum_{i=1}^{|k|} w(t_i, X_k)^2} \cdot \sqrt{\sum_{i=1}^{|k|} w(t_i, Y_k)^2}}, \tag{2}$$

where $|k|$ is the number of terms in the semantic class of k.

3.2.2. Temporal similarity. There are three problems one has to deal with before temporal expressions can be used automatically in the TDT tasks: *recognition*, *formalization* and *comparison*. First, the expressions have to be extracted from the text. Second, the expressions need a formal interpretation, an interval on a linearly ordered global time-line, for example. In addition to the expression, one often needs some context information, such as the utterance time and the tense of the relevant verb, to map the natural language expression on to the time-line. And finally, there has to be a suitable method of employing the formalized expressions.

In recognizing temporal expressions, we employ functional dependency parsing (see, e.g., Järvinen and Tapanainen 1997) and finite-state automata. Once an expression is recognized, the terms it contains are converted to operations that move the utterance time back or forth on the time with respect to the utterance time. We define these operations on top of a *calendar* (Goralwalla et al. 2001) that defines a global time-line, granularities of time units and conversion functions between the granularities. The granularity Year would consist of 365 units of days, with the exception of leap years. The granularity March would equal to every twelfth element of granularity Month. A Month contains from 28 to 31 Days, and so on.

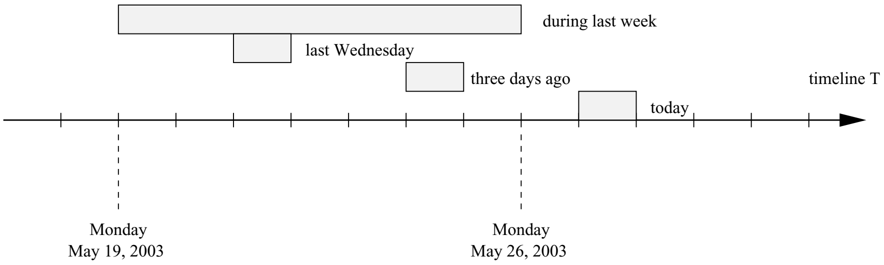

Figure 2. An example of mapping four temporal expressions uttered on May 27, 2003.

For example, in figure 2 there are four temporal utterances mapped onto a timeline. The expression "*during last week*" first shifts the utterance time (Tuesday May 27, 2003) to the beginning of the previous element of granularity Week and then spans the duration from the Monday to the next Sunday. Similarly, the expression "*last Wednesday*" finds the previous element of granularity Wednesday. However, if the expression was "*on Wednesday*", the mapping would require the verb tense. Otherwise, it is not clear, which Wednesday is the expression refers to. In sentences like "*The meeting was scheduled for Wednesday*" the ambiguity cannot be resolved with just the verb tense.

Many of the expressions can be formalized in a straight-forward manner. We are, however, able to deal with more complex expressions, such as 'The strike started *on the 15th of May 1919*. It lasted *until the end of June*, although there was still turmoil *in late January next year*". Further details can be found in Makkonen and Ahonen-Myka (2003).

The temporal similarity of two documents is a result of a pair-wise comparison of the expressions: each start-end pair of one document is compared to each of the start-end pairs of the other. Krippendorff (1995) has carried out various investigations with intervals and motivated by his work we propose a cross-tabulation illustrated in figure 3. The diagonal represents the synchronous points between the time-axis of document A and time-axis of document B. The shaded areas correspond to the intersections between the intervals $A = \{A_1, A_2, A_3\}$ and $B = \{B_1, B_2, B_3, B_4\}$.

If the two sets contain the same intervals, they cover each other completely. In such case, all of the intervals would be shaded completely along the diagonal in figure 3. In case there are the intervals are not equal, the larger intervals provide weaker coverage than shorter ones. For example, consider comparing a day and a year versus a day and a weekend. We base the temporal similarity on how well the two sets of intervals cover each other. The more the intervals overlap with respect to their lengths, the higher the similarity.

Let $\mathcal{T}$ stand for a global time-line. An interval $x \subseteq \mathcal{T}$ on this time-line is defined simply as a pair of a start and an end points, $x_s, x_e \in \mathcal{T}$ such that $x = [x_s, x_e]$. To determine the similarity of two intervals, we measure the portion of overlap with respect to the lengths of the intervals. The similarity of two temporal intervals $x \subseteq \mathcal{T}$ and $y \subseteq \mathcal{T}$ is defined by a function

$$\mu_t(x, y) = \frac{2\,\Delta([x_s, x_e] \cap [y_s, y_e])}{\Delta(x_s, x_e) + \Delta(y_s, y_e)}, \qquad (3)$$

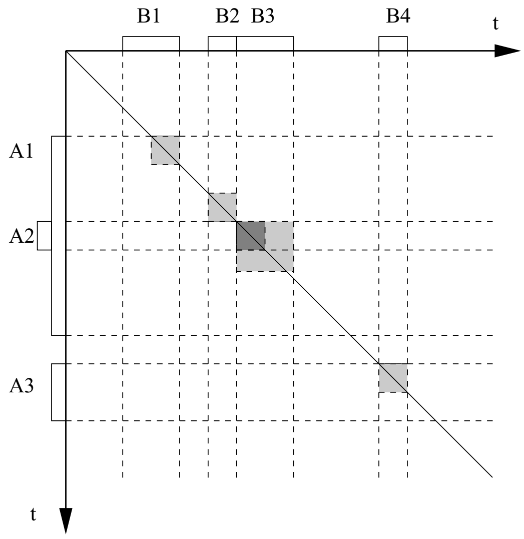

Figure 3. A cross-tabulation of two sets of intervals A and B.

where $\Delta : \mathcal{T} \times \mathcal{T} \rightarrow \mathbb{R},\ \Delta(x_s, x_s) = 1$ is the duration (in days) of the given interval. If the interval x is completely covered by the interval y, then $\mu_t(x, y) = 1$. If they are distinct, then $\mu_t(x, y) = 0$. In the example of figure 3, the intersections $A_3 \cap B_4$ and $A_2 \cap B_3$ would result in a higher μ_t-value than any of the intersections $A_1 \cap B_1$, $A_1 \cap B_3$, and $A_1 \cap B_2$, because the sizes of the intersections $A_3 \cap B_4$ and $A_2 \cap B_3$ are closer to the sums $|A_3| + |B_4|$ and $|A_2| + |B_3|$.

In measuring the temporal similarity of two documents, we calculate $\mu_t(x_i, y_j)$ for each pair of intervals of TEMPORAL sub-vectors $X = \{x_1, x_2, \ldots, x_n\}$ and $Y = \{y_1, y_2, \ldots, y_m\}$. The results are stored in a *cover matrix* illustrated in figure 4. From these pair-wise similarities we select the maxima for each interval x_i denoted by $\max(x_i, Y)$. The similarity of two TEMPORAL sub-vectors X and Y is the average of these maxima. Hence, the temporal similarity $\sigma_t(X, Y)$ of the two sets of intervals determined by a function

$$\sigma_t(X, Y) = \frac{\sum_{i=1}^{n} \max(\mu_t(x_i, Y)) + \sum_{j=1}^{m} \max(\mu_t(X, Y_j))}{m + n} \tag{4}$$

	y_1	$\ldots$	y_m	max
x_1	$\mu_t(x_1, y_1)$	$\ldots$	$\mu_t(x_1, y_m)$	$\max(\mu(x_1, Y))$
$\vdots$	$\vdots$		$\vdots$	
x_n	$\mu_t(x_n, y_1)$	$\ldots$	$\mu_t(x_n, y_m)$	$\max(\mu(x_n, Y))$
max	$\max(\mu_t(X, y_1))$		$\max(\mu_t(X, y_m))$	

Figure 4. A cover matrix of TEMPORAL sub-vectors X and Y.

Table 1. An example of a 5-level ontology.

Location	Type	Level 1	Level 2	Level 3	Level 4	Level 5
Delft	City	Europe	W. Europe	Netherlands	Zuid-Holland	Delft
Europe	Continent	Europe	–	–	–	–
Haag	City	Europe	W. Europe	Netherlands	Zuid-Holland	Haag
Main	River	Europe	W. Europe	Germany	Rhine	Main
Netherlands	Country	Europe	W. Europe	Netherlands	–	–
North Sea	Sea	Atlantic	North Sea	–	–	–
Rhine	River	Europe	W. Europe	Switzerland, Germany, France, Netherlands	North Sea	Rhine

The values of $\mu_t(x, y)$ vary between 0 and 1, and naturally the average of their maxima vary in the same range.

3.2.3. Spatial similarity. When reporting floods in Siberia, the news might use geographical terms such as Russia, Lena, Vilyuy, Lensk and Yakutsk. Clearly, these terms have nothing in common on the surface; their relevance cannot be understood without a geographical ontology. Thus, each term would have to be mapped onto a structure, where the meaning of a term is its relation to other terms.

We employ a 5-level hierarchy in our knowledge of the world as portrayed in Table 1. The levels involved depend on the type of the location. As to land, the levels are continent, region, country, administrative region (e.g., province, state, commune, municipality, municipio, gemeente, kommun), and city. In addition to administrative region, level 4 can also be mountains, seas, lakes and (larger) rivers that include or connect to mountain peaks and (smaller) rivers.

Figure 5 shows a simplified hierarchy containing a number of places. Each node in the tree stands for a location. In case we want to measure the similarity of two such locations

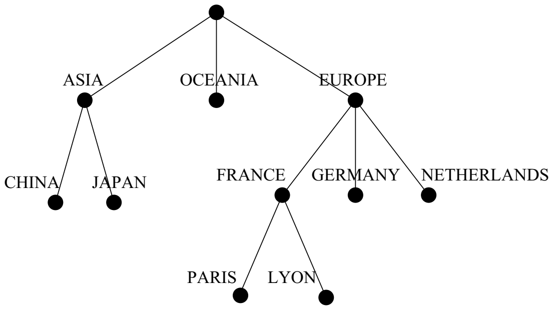

Figure 5. A simplified ontology tree.

x and y, we divide the length of the common path $x \cap y$ with the sum of the paths to the elements

$$\mu_s(x, y) = \frac{\lambda(x \cap y)}{\lambda(x) + \lambda(y)}, \tag{5}$$

where $\lambda(x)$ is the length of the path from the root of the ontology to the element x. We assign $\mu_s(x, x) = 1$. Comparing *France* and *Germany* in the simplified ontology of figure 5 yields $1/(2 + 2) = 1/4$. Similarly, *China* and *Paris* yield $0/(2 + 3) = 0$. *Paris* and *France* have similarity of $2/(2 + 3) = 2/5$.

All the spatial references of one document are to be compared with all the spatial references of another. For this, we employ the cover matrix presented in Section 3.2.2. For each term in one sub-vector we calculate the maximum similarity among the terms of the other. The spatial similarity $\sigma_s(X, Y)$ of two LOCATIONS vectors $X = \{x_1, x_2, \ldots, x_n\}$ and $Y = \{y_1, y_2, \ldots, y_m\}$ is the average of these maxima, i.e.,

$$\sigma_s(X, Y) = \frac{\sum_{i=1}^{n} \max(x_i, Y) + \sum_{j=1}^{m} \max(X, y_j)}{m + n}. \tag{6}$$

4. Topic detection and tracking algorithms

The class-wise comparison of two event vectors results in a vector $\mathbf{v} = (v_1, v_2, v_3, v_4) \in \mathbb{R}^4$. The question remains, how to go about turning this vector into a decision, whether two documents discuss the same topic or not. In our previous work, in addition to the traditional similarity coefficients, we presented a heuristic weighted sum that also turned out to give the best results (Makkonen et al. 2003).

4.1. Weighted sum

The similarity of two event vectors $\mathbf{X}$ and $\mathbf{Y}$ is based on a weighted sum $\delta(\mathbf{X}, \mathbf{Y})$ of the class-wise similarities

$$\delta(\mathbf{X}, \mathbf{Y}) = \sum_{i=1}^{4} w_i \cdot \sigma_i(X_i, Y_i) = \langle \mathbf{w} \cdot \mathbf{v} \rangle, \tag{7}$$

in which $w_i \in \mathbb{R}$ is the weight for the semantic class i, σ_i is the similarity measure of semantic class i, X_i and Y_i are the sub-vectors of the semantic class i, and $\langle \mathbf{w} \cdot \mathbf{v} \rangle$ is the inner-product of the weight vector and the class-wise similarity vector. The of approach of Makkonen et al. (2003) was heuristic for two reasons: Firstly, the weights representing the relative emphasis of the semantic classes were a result of trial-and-error, instead of proper optimization. Secondly, we rewarded for the co-occurrence of non-zero values in similarities of TERMS, LOCATIONS and NAMES and punished for the lack of the similarity of TERMS. In the following we address both of these inadequacies.

One way to transform the vector **v** into a decision is to divide the space of class-wise similarities into two sub-spaces by a hyperplane. A positive distance indicates that the documents are about the same topic, and negative distance that they are not. The hyperplane is defined by a linear equation

$$\psi(\mathbf{v}) = \langle \mathbf{w} \cdot \mathbf{v} \rangle + b = 0, \tag{8}$$

where $\mathbf{w} \in \mathbb{R}^4$ is orthogonal to the hyperplane and $b \in \mathbb{R}$ is the bias. One very simple solution for finding **w** and b would be a Rosenblatt's *perceptron* (see e.g. Mitchell 1997). Provided with both positive and negative examples the perceptron iterates several times over the data, and adjusts the weight vector each time it makes a misclassification until no mistakes are made. If the data is linearly separable, the perceptron is guaranteed to converge to the optimal set of weights.

One of the problems we have had was that the data is not linearly separable and it was very difficult to build a reliable classifier with four dimensional vectors. To circumvent this and to address the second heuristic aspect we had earlier, we define a function $\phi : \mathbb{R}^4 \to \mathbb{R}^{15}$ that expands the vector **v** with the powerset of its dimensions. In other words, the initial vector $\mathbf{v} = (v_1, v_2, v_3, v_4)$ is expanded into

$$\mathbf{v}' = (v_1, v_2, v_3, v_4, v_1v_2, v_1v_3, \ldots, v_1v_2v_3, v_1v_2v_4, \ldots, v_1v_2v_3v_4).$$

Now, the linear equation of the hyperplane is expressed as

$$\psi(\mathbf{v}) = \langle \mathbf{w}' \cdot \phi(\mathbf{v}) \rangle + b = 0, \tag{9}$$

where $\mathbf{w}' \in \mathbb{R}^{15}$ is a 15-dimensional weight vector. The ϕ-mapping increased the learnability of the data considerably, although it is still not linear. We used the delta rule in adjusting the weights which ensures the asymptotical converge towards the optimal weight vector. The powerset expansion can lead to redundant dimensions, but the perceptron would attribute them with weights close to 0.

4.2. *Topic tracking*

According to the topic tracking task definition, each topic is trained and tracked independently of the others, and at encountering a new document the system has to make just one binary decision: is this document discussing the topic or not (Allan 2002b). The given topic is defined in terms of sample documents, usually from one to four. We do not update the document representation.

A straight-forward topic tracking algorithm is shown in Table 2. In the beginning, we construct the event vector for the given topic from the sample documents. Then, we go through the incoming documents one-by-one, build the event vector and compare it class-wise against the topic vector. The judgment on rows 8–10 is based on the inner product between the class-wise similarities and the pre-calculated weight vector **w**.

Table 2. A simple topic tracking algorithm with weighted sum.

```
topic ← buildVector();
for each new document $d$
    doc ← buildVector($d$);
    $\mathbf{v}$ ← (); answer ← ();
    for each semantic class $c$
        $v[c]$ ← $sim_c(doc_c, topic_c)$;
    end;
    if ($\langle \mathbf{w} \cdot \phi(\mathbf{v}) \rangle + b \geq 0$)
      then decision ← 'YES';
      else decision ← 'NO';
    fi;
end;
```

4.3. First story detection

The first-story detection makes a decision whether the incoming document discusses a new topic or not, i.e., when to start a new topic cluster. Unlike topic tracking, first-story detection considers all the previous data before making its judgment.

The first-story detection algorithm is described in Table 3. Initially, the set of found topics is empty. Naturally, the first document in the system is also a first-story. For each incoming document, we try to find the closest match from the previous documents. If the closest match is similar enough, i.e., if the decision function yields a value large enough, the documents are considered to discuss the same topic. If the similarity is below a threshold, the document is considered to be a first-story.

5. Experiments

5.1. Corpus and ontology

One of the participants in the TDT programme has been the Linguistic Data Consortium. It has been responsible for the construction and maintenance of several corpora designed for the purposes of evaluating TDT systems. We run our experiments on TDT2 corpus that spans from January 4 to June 30, 1998 consisting of over 60,000 documents. The news are from six sources of three different types: two on-line newspapers (the New York Times News Service and the Associated Press Worldstream News Service), two TV broadcasts (CNN "Headline News" and ABC "World News Tonight") and two radio broadcasts (Public Radio International's "The World" and the Voice of America). Table 4 presents the number of documents and the average size of a document per source. We do not include the three Asian sources in Mandarin Chinese in the experiments.

The corpus is text, and roughly two thirds of it is transcribed from speech to text by *automatic speech recognition* (ASR), and that the broadcasts are considerably shorter than

Table 3. A first-story detection algorithm.

```
topics ← (); decision ← ()
for each new document d
    doc ← buildVector(d);
    max ← 0; max_topic ← ();
    for each topic
        for each semantic class c
            v[c] ← sim_c(doc_c, topic_c);
        end;
        if (⟨w · φ(v)⟩ + b ≥ max)
            max ← ⟨w · φ(v)⟩ + b;
            max_topic ← topic;
         end;
     end;
     if (max < θ)
         then decision[d] ← 'first-story';
         else decision[d] ← max_topic;
     fi;
     add(topics, doc);
 end;
```

Table 4. Information on TDT2 corpus. The total size is 64,527 documents.

Source	Type	# documents	Avg # words/doc
New York Times (NYT)	Newswire	11,795	850.363
Associated Press (APW)	Newswire	12,760	346.623
CNN	Television	21,588	70.001
ABC	Television	3,179	108.732
Public Radio Intl (PRI)	Radio	4,390	224.320
Voice of America (VOA)	Radio	10,815	101.876

the newswire documents. The ASR is prone to spelling mistakes, especially on non-English names.

There are about 10,000 documents with topic labels of 100 events distributed over the length of the corpus, 35 of which are in the training set comprised by the first two months. The validation set contains 25 and the testing set 34 events. A more detailed anatomy of TDT2 can be found in Cieri et al. (2002).[1]

Our ontology is a combination of the data from Statistics Finland (Tilastokeskus), World Factbook (CIA 2003), and a list of geographic feature names (NIMA). The amount of different kinds of geographic names can be seen in Table 5.

Table 5. Ontology statistics.

Type		Type	
Continents	6	Mtn. peaks	269
Regions	32	Mountains	145
Countries	285	Rivers	390
Adm. districts	3350	Lakes	282
Cities	49826	Oceans/seas	77
Deserts	31	Islands	305

5.2. Term extraction in TDT2

Most of our term extraction relies on the Connexor Functional Dependency Grammar parser for English (EN-FDG)[2] that is capable of syntactical, morphological and dependency functional parsing. As TERMS we pass subjects, objects, attributive nominals, prepositional complements and main verbs. If a term is recognized as NAME or a LOCATION, it is not passed as a TERM. Although the prepositions, particles and other functional words were filtered by the selection criteria, we applied a stoplist to filter the very common and content-poor verbs (e.g., see, believe, think, and so on).

The recognition of NAMES and LOCATIONS was based on Connexor's Term Extractor (EN-BRACKETS). However, the majority of the data is automatically recognized speech and some portion of it lacks capital letters that are crucial for the efficient term extraction. We were able to fill in some of the upper-case letters with a set of simple syntax-based automata and a *gazetteer*, a list of names of people, organizations and geographical locations. This operation increased the average of LOCATIONS from 0.000 to 2.485 and the average of NAMES from 2.470 to 3.253 in the CNN material.

Table 6 presents the expectation of the number of elements of semantic classes in a document originating from different sources. Clearly, the on-line news of NYT and APW contain most terms in all semantic classes. The ratio of LOCATIONS and NAMES is very high in the the radio broadcasts compared to the rest. This may be due to the habit of reporters introducing themselves and the name of the station and its whereabouts at every turn. The frequencies of terms in CNN material are decreased by numerous empty or very short documents that contain only brief references of what are the topics in during the next half an hour.

Table 6. The average number of occurrences of a semantic class per source.

Semantic class	NYT	APW	CNN	ABC	PRI	VOA	Avg
LOCATIONS	13.726	11.016	2.485	3.780	7.345	6.368	7.910
NAMES	35.802	14.654	3.253	5.656	6.332	7.850	14.109
TERMS	191.077	87.932	31.139	51.011	74.505	54.178	86.581
TEMPORALS	9.018	5.495	1.558	2.616	2.463	2.301	4.348

Table 7. Contingency table for TDT system responses (Fiscus and Doddington 2002).

		Corpus annotation	
		Target	Non-target
System	Yes (*target*)	correct$_{(+)}$	*false alarm*
Response	No (*non-target*)	*miss*	correct$_{(-)}$

5.3. *Evaluation*

Traditionally in IR, the effectiveness of a system is assessed in terms of correct judgments with respect to number of judgments made (precision) and the number of all possible correct judgments (recall). The tasks of TDT are fundamentally detection tasks and therefore, instead of using precision and recall, the evaluation is based on error-rates, the number of false-alarms and misses (Fiscus and Doddington 2002). The matrix for system responses for TDT is shown in Table 7.

We also report the precision p, recall r and F_1 measures:

$$p = \frac{\#correct_{(+)}}{\#correct_{(+)} + \#false\ alarm}$$

$$r = \frac{\#correct_{(+)}}{\#correct_{(+)} + \#miss}$$

$$F_1 = \frac{2 \cdot p \cdot r}{p + r}$$

In order to enable cross-system comparisons, the error-rates are combined into one a single detection cost by the following formula:

$$C_{\text{det}} = (C_{\text{miss}} \cdot P_{\text{miss}} \cdot P_{\text{target}} + C_{\text{fa}} \cdot P_{\text{fa}} \cdot (1 - P_{\text{target}})) \tag{10}$$

where C_{miss} and C_{fa} are the costs of a miss and a false-alarm, P_{miss} and P_{fa} are the probabilities of a miss and false-alarm which are determined in the evaluation, and P_{target} is the prior target probability that represents the portion of labeled documents with respect to all the documents in the corpus. Following a convention in the TDT evaluations (see e.g. Yang et al. 2002b), we assign $C_{\text{miss}} = 1.0$, $C_{\text{fa}} = 0.1$ and $P_{\text{target}} = 0.2$.

The scores of C_{det} are often normalized onto a more meaningful range by dividing the score by the cost of answering YES or NO consistently. Thus, we get the normalized detection cost

$$(C_{\text{det}})_{\text{norm}} = \frac{(C_{\text{miss}} \cdot P_{\text{miss}} \cdot P_{\text{target}} + C_{\text{fa}} \cdot P_{\text{fa}} \cdot (1 - P_{\text{target}}))}{\min(C_{\text{miss}} \cdot P_{\text{target}}, C_{\text{fa}} \cdot P_{\text{target}}))}. \tag{11}$$

A cost $(C_{\text{det}})_{\text{norm}}$ of zero would mean that the system is infallible, but the score of one would mean that it is doing no better than saying YES or NO to all documents.

5.4. Results

We trained the perceptron with 2000 positive and 2000 negative samples and evaluated it with a set of the same size. The data was not linearly separable, and in the training set there were a little over a hundred documents that could not be classified correctly despite increasing the epochs and adjusting the learning rate. We also tested nonlinear models with Radial Basis Function of SVM^{light} (Joachims 2002) with a number of cost models (stressing the high recall/low miss-rate) and data samples, but none of the classifiers was substantially better than the linear perceptron.

Our baseline is a simple cosine classifier with TFIDF term weights (see Eqs. (1) and (2)). The features were selected on the basis of the part-of-speech; we chose verbs, nouns, and adjectives that were not in a stop-list. The IDF-weights were calculated from the training and validation sets of the TDT2 corpus. Neither tracking nor first story detection deferred their decision and in all the tracking experiments the number of samples defining the topic $N_t = 1$.

The topic tracking micro-average (story-weighted) results are presented in Table 8. In the 'best' column $(C_{\text{det}})_{\text{norm}}$ means that the row was obtained by minimizing the normalized detection cost. Similarly, the row with F_1 was the result of the best F_1-measure.

The baseline results are highly similar to those of Schultz and Liberman (2002), who had a similar kind of monolingual tracking approach based on TFIDF-weighted cosine. They found that the simple tracking method performed equally well or better than the more sophisticated methods. Here, the semantical augmentation degraded the results.

An obvious reason is that the similarity measures of TEMPORALS and LOCATIONS increase the similarity too much with distantly relevant terms, as neither of them has a vagueness function that would account for the inherent indefiniteness of the term. The matching terms '*Asia*' yield the same similarity value as the terms '*Washington*', although the former covers approximately one third of the land mass of Earth and contains about half of the population. On the other hand, as the name of the capital '*Washington*' occurs often in the context of international or national politics in the form "*Washington says so-and-so*", for example.

Similarly, two temporal expressions denoting the year 1997, for example, overlap each other completely and would thus give a similarity value of one, just as two matching definate dates would. Then again, in the corpus year 1979 refers typically to either Pol Pot or to Iranian revolution. Things in distant past are not associated with exact dates very often. It is not straight-forward to establish a vagueness function that would take both the duration

Table 8. The micro-average results of topic tracking.

Method	Best	C_{det}	$(C_{\text{det}})_{\text{norm}}$	P_{miss}	P_{fa}	p	r	F_1
Cosine	$(C_{\text{det}})_{\text{norm}}$	0.0058	0.0720	0.0100	0.0470	0.2361	0.7900	0.2927
Cosine	F_1	0.0524	0.6553	0.2582	0.0097	0.5297	0.7418	0.5481
Weighted sum	$(C_{\text{det}})_{\text{norm}}$	0.0471	0.5214	0.1818	0.0668	0.1646	0.8181	0.2741
Weighted sum	F_1	0.0849	1.0621	0.4242	0.0015	0.8636	0.5758	0.6910

Table 9. The micro-average results of first story detection.

Method	Best	C_{det}	$(C_{\text{det}})_{\text{norm}}$	P_{miss}	P_{fa}	p	r	F_1
Cosine	$(C_{\text{det}})_{\text{norm}}$	0.0033	0.0414	0.0000	0.0414	0.4583	1.0000	0.6386
Cosine	F_1	0.0381	0.4768	0.1818	0.0223	0.5625	0.8181	0.6667
Weighted sum	$(C_{\text{det}})_{\text{norm}}$	0.0036	0.0446	0.0000	0.0446	0.4400	1.0000	0.6111
Weighted sum	F_1	0.0558	0.6977	0.2727	0.0159	0.6154	0.7272	0.6667

and the distance into account. In addition, it is difficult to determine, what interval "*three years ago*" really refers to.

The micro-averaged results of first story detection are shown in Table 9. Unlike in tracking, the detection methods go through all the previous documents before making a decision. This has benefitted the weighted sum more that the baseline. It seems that by choosing the most similar document from the previous ones and making a decision based on that, both methods were able to decrease the misses down to zero. Of course, the tasks are not the same, but in the tracking results, it is particularly the number of misses that degrade the weighted sum performance.

Allan et al. (2000) see the "reasonable" performance as missing less than 10% of the first stories while generating only 0.05% false alarms (Allan et al. 2002). Although both of the first-story detection approaches meet the first criterion, the number of false alarms is still too high.

Although the performance of the semantic class approach in tracking is low and considerably worse than the baseline, there is a slight consolation in that the F_1-measures of the semantic class approach are actually better than what we had earlier (Makkonen et al. 2003) in spite of that the TDT2 corpus is ten times larger and far more heterogeneous than the Finnish single-source collection. Clearly, the proper optimization and the expansion of the class-wise similarity vector were beneficial.

What is there to be done to improve the performance? Firstly, the spatial and temporal similarities need to be re-thought. Better similarity functions would have an impact on the learnability of the similarity vectors and thus on the perceptron weights and thus on the results. As discussed above, the vagueness should be taken into account, but it is not at all clear, in what way the geographical or the semantical features and corpus-based statistical measures should be combined. We are thinking of representing temporal intervals as probability distributions, which would account for vagueness in expressions such as "*three years ago*". Also, the recognition of TEMPORALS, LOCATIONS and NAMES should be improved.

Secondly, there is work to be done on term selection techniques and term weighting. Currently, the feature selection uses only syntactical characteristics and a stop-list. By building *a posteriori* approaches that are shown all the data and the labels, we plan to examine what kind of terms undermine or reinforce the tasks and can they be detected automatically. In the same way, we plan to investigate the temporal and spatial cohesion of an event: how many and what kind of terms appear as the event evolves? Are the new terms semantically relevant to the old.

6. Conclusions

We presented an approach on a topic detection and tracking approach that employs semantic classes in event representation. We split the term space into four semantic classes: places, names, temporal expressions and general terms. This makes it possible to compare two documents class-wise, and assigning each class a dedicated similarity measure that can utilize an external ontology. We have built a geographical and temporal ontologies the use of which relies on extensive use of natural language processing techniques.

We built an optimizer for the weights of the semantic classes. A simple perceptron was trained with samples of pair-wise comparisons of documents to distinguish, when two documents discuss the same event and when they do not.

The results showed the efficiency of cosine based similarity with TFIDF term weights, which has been a difficult baseline to surpass. The semantical augmentation seemed to degrade the performance, especially in topic tracking. We suspect that this is at least partially due to the inadequate spatial and temporal similarity functions. In the future, we will work on similarity functions for spatial and temporal terms that would take vagueness into account and would reduce the noise. We will also work on term selection and term weighting.

Notes

1. There is plenty of information available at the LDC's WWW site http://www.ldc.upenn.edu/Projects/TDT2/.
2. http://www.connexor.com.

References

Allan J (2002a) Introduction to topic detection and tracking. In: Allan (2002b), pp. 1–16.

Allan J (2002b), Ed. Topic Detection and Tracking: Event-based Information Organization. Kluwer Academic Publishers, Norvell, MA, USA.

Allan J, Carbonell J, Doddington G, Yamron J and Yang Y (1998a) Topic detection and tracking pilot study: Final report. In: Proceedings of DARPA Broadcast News Transcription and Understanding Workshop. Lansdowne, VA, pp. 194–218.

Allan J, Jin H, Rajman M, Wayne C, Gildea D, Lavrenko V, Hoberman R and Caputo D (1999) Topic-based novelty detection. Technical Report, Center for Language and Speech Processing, Johns Hopkins University, Baltimore, MD, Summer Workshop Final Report. http://www.clsp.jhu.edu/ws99/ (visited September 19th, 2003).

Allan J, Lavrenko V and Jin H (2000) First story detection in TDT is hard. In: Proceedings of the 9th International Conference on Information and Knowledge Management (CIKM). ACM Press, pp. 374–381.

Allan J, Lavrenko V and Papka R (1998b) Event tracking. Technical Report IR-128, Department of Computer Science, University of Massachusetts.

Allan J, Lavrenko V and Swan R (2002) Explorations within topic tracking and detection. In: Allan (2002b), pp. 197–224.

Allan J, Papka R and Lavrenko V (1998c) On-line new event detection and tracking. In: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM Press, pp. 37–45.

Carthy J (2002) Lexical chains for topic tracking. PhD thesis, Department of Computer Science, National University of Dublin.

Central Intelligence Agency, CIA (2003) The World Factbook. http://www.cia.gov/cia/publications/factbook/ (visited September 19th, 2003).

Cieri C, Strassel S, Graff D, Martey N, Rennert K and Liberman M (2002) Corpora for topic detection and tracking. In: Allan (2002b), pp. 33–66.

Cutting DR, Karger DR, Pedersen JO and Tukey JW (1992) Scatter/Gather: A cluster-based approach to browsing large document collections. In: Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM Press, pp. 318–329.

Falk P (1989) The past to come. Economy and Society, 17(3):374–394.

Fiscus J and Doddington G (2002) Topic detection and tracking evaluation overview. In: Allan (2002b), pp. 17–31.

Gerner DJ, Schrodt PA, Francisco R and Weddle JL (1994) The analysis of political events using machine coded data. International Studies Quarterly, 38:91–119.

Goralwalla IA, Leontiev Y, Özsu MT, Szafron D and Combi C (2001) Temporal Granularity: Completing the Puzzle. Journal of Intelligent Information Systems, 16(1):41–63.

Järvinen T and Tapanainen P (1997) A dependency parser for english. Technical Report TR-1, Department of General Linguistics, University of Helsinki.

Joachims T (2002) Learning to Classify Text Using Support Vector Machines. Kluwer Academic Publishers, Boston.

Krippendorff K (1995) On the reliability of unitizing continuous data. In: Marsden PV, Ed., Sociological Methodology. Blackwell, Cambridge, MA, pp. 47–76.

Lavrenko V, Allan J, DeGuzman E, LaFlamme D, Pollard V and Thomas S (2002) Relevance models for topic detection and tracking. In: Proceedings of Human Language Technology Conference. San Diego, CA, pp. 104–110.

Leek T, Schwartz R and Sista S (2002) Probabilistic approaches to topic detection and tracking. In: Allan (2002b), pp. 67–84.

Makkonen J and Ahonen-Myka H (2003) Utilizing temporal information in topic detection and tracking. In: Koch T and Solveig IT, Eds., Proceedings of the 7th European Conference on Digital Libraries (ECDL). Springer-Verlag, pp. 393–404.

Makkonen J, Ahonen-Myka H and Salmenkivi M (2002) Applying semantic classes in event detection and tracking. In: Sangal R and Bendre SM, Eds., Proceedings of International Conference on Natural Language Processing (ICON). Mumbai, India, pp. 175–183.

Makkonen J, Ahonen-Myka H and Salmenkivi M (2003) Topic detection and tracking with spatio-temporal evidence. In: Sebastiani F, Ed., Proceedings of the 25th European Conference on Information Retrieval Research (ECIR). Springer-Verlag, Heidelberg, pp. 251–265.

Miller GA (1995) WordNet: A lexical database for English. Communications of ACM, 38(11):39–41.

Mitchell TM (1997) Machine Learning. McGraw-Hill.

NIMA, National Imagery and Mapping Agency, Geographic Feature names. http://www.nima.mil/gns/html/index.html (visited September 19th, 2003).

Papka R (1999) On-line new event detection, clustering and tracking. PhD Thesis, Department of Computer Science, University of Massachusetts.

Pons A, Berlanga R and Rumz-Shulcloper J (2002) Temporal-semantic clustering of newspaper articles for event detection. In: Proceedings of Pattern Recognition in Information Systems (PRIS2002). Ciudad Real, Spain, pp. 104–113.

Salton G and Buckley C (1988) Term-weighting approaches in automatic text retrieval. Information Processing and Management, 24(5):513–523.

Schultz JM and Liberman MY (2002) Towards a "Universal Dictionary" for multi-language information retrieval applications. In: Allan (2002b), pp. 225–242.

Sebastiani F (2002) Machine learning in automated text categorization. ACM Computing Surveys, 34(1):1–47.

Swan R and Allan J (1999) Extracting significant time varying features from text. In: Proceedings of the Eighth International Conference on Information and Knowledge Management (CIKM-99). ACM Press, pp. 38–45.

Tilastokeskus (Statistics Finland) http://www.stat.fi (visited September 19th, 2003).

Yamron JP, Gillick L, van Mulbregt P and Knecht S (2002) Statistical models of topical content. In: Allan (2002b), pp. 115–134.

Yang Y, Ault T, Pierce T and Lattimer C (2000) Improving text categorization methods for event detection. In: Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM Press, pp. 65–72.

Yang Y, Carbonell J, Brown R, Lafferty J, Pierce T and Ault T (2002a) Multi-strategy learning for TDT. In: Allan (2002b), pp. 85–114.

Yang Y, Carbonell J, Brown R, Pierce T, Archibald BT and Liu X (1999) Learning approaches for detecting and tracking news events. IEEE Intelligent Systems Special Issue on Applications of Intelligent Information Retrieval, 14(4):32–43.

Yang Y and Liu X (1999) A re-examination of text categorization methods. In: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM Press, pp. 42–49.

Yang Y, Zhang J, Carbonell J and Jin C (2002b) Topic-conditioned novelty detection. In: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM Press, pp. 688–693.