Department of Computer Science
Series of Publications A
Report A-2009-8

Semantic Classes in Topic Detection and Tracking

Juha Makkonen

To be presented, with the permission of the Faculty of Science of the University of Helsinki, for public criticism in Auditorium XII, University Main Building, on November 13th, 2009, at 12 o'clock noon.

University of Helsinki
Finland

Contact information

Postal address:
Department of Computer Science
P.O. Box 68 (Gustaf Hällströmin katu 2b)
FI-00014 University of Helsinki
Finland

Email address: postmaster@cs.Helsinki.FI (Internet)

URL: http://www.cs.Helsinki.FI/

Telephone: +358 9 1911

Telefax: +358 9 191 51120

ISSN 1238-8645
ISBN 978-952-10-5860-8 (paperback)
ISBN 978-952-10-5861-5 (PDF)
Computing Reviews (1998) Classification: H.3.1, H.3.3, I.5.3, 5.4
Helsinki 2009
Helsinki University Print

Semantic Classes in Topic Detection and Tracking

Juha Makkonen

Department of Computer Science
P.O. Box 68, FI-00014 University of Helsinki, Finland
juha.makkonen@cs.helsinki.fi
http://www.cs.helsinki.fi/u/jamakkon

PhD Thesis, Series of Publications A, Report A-2009-8
Helsinki, November 2009, 165 pages
ISSN 1238-8645
ISBN 978-952-10-5860-8 (paperback)
ISBN 978-952-10-5861-5 (PDF)

Abstract

Topic detection and tracking (TDT) is an area of information retrieval research the focus of which revolves around news events. The problems TDT deals with relate to segmenting news text into cohesive stories, detecting something new, previously unreported, tracking the development of a previously reported event, and grouping together news that discuss the same event. The performance of the traditional information retrieval techniques based on full-text similarity has remained inadequate for online production systems. It has been difficult to make the distinction between same and similar events.

In this work, we explore ways of representing and comparing news documents in order to detect new events and track their development. First, however, we put forward a conceptual analysis of the notions of topic and event. The purpose is to clarify the terminology and align it with the process of news-making and the tradition of story-telling.

Second, we present a framework for document similarity that is based on semantic classes, i.e., groups of words with similar meaning. We adopt people, organizations, and locations as semantic classes in addition to general terms. As each semantic class can be assigned its own similarity measure, document similarity can make use of ontologies, e.g., geographical taxonomies. The documents are compared class-wise, and the outcome is a weighted combination of class-wise similarities.

Third, we incorporate temporal information into document similarity. We formalize the natural language temporal expressions occurring in the text, and use them to anchor the rest of the terms onto the time-line. Upon comparing documents for event-based similarity, we look not only at matching terms, but also how near their anchors are on the time-line.

Fourth, we experiment with an adaptive variant of the semantic class similarity system. The news reflect changes in the real world, and in order to keep up, the system has to change its behavior based on the contents of the news stream. We put forward two strategies for rebuilding the topic representations and report experiment results.

We run experiments with three annotated TDT corpora. The use of semantic classes increased the effectiveness of topic tracking by 10-30% depending on the experimental setup. The gain in spotting new events remained lower, around 3-4%. The anchoring the text to a time-line based on the temporal expressions gave a further 10% increase the effectiveness of topic tracking. The gains in detecting new events, again, remained smaller. The adaptive systems did not improve the tracking results.

Computing Reviews (1998) Categories and Subject Descriptors:
H.3.1 Information Storage and Retrieval: Content Analysis and Indexing - indexing methods
H.3.3 Information Storage and Retrieval: Information Search and Retrieval - information filtering
I.5.3 Pattern Recognition: Clustering - similarity measures
I.5.4 Pattern Recognition: Applications - text processing

General Terms:
algorithms, experimentation, measurement,performance

Additional Key Words and Phrases:
topic detection and tracking, first story detection, topic tracking, semantic class, ontology-based document similarity, temporal expression, geographical taxonomy

Acknowledgements

My journey from a young postgraduate to a respondent has not been a straight path from A to B, nor has it been a solitary travel. Over the years, I have been aided by numerous people without whom this work would not have been possible. I am most indebted to my supervisors Helena Ahonen-Myka and Esko Ukkonen for their invaluable guidance, encouragement and patience. I also wish to thank the pre-examiners Eero Hyvönen and Kalervo Järvelin for their insightful comments and suggestions.

I owe a debt of gratitude towards the members of the DoReMi research group, especially Greger Lindén, Marko Salmenkivi, Antoine Doucet, Miro Lehtonen, Lili Aunimo, Oskari Heinonen, and Jussi Piitulainen, whose collaboration, suggestions and comments have been a great help. In addition, the informal coffee break discussions were, of course, entertaining and even enlightening on occasion.

I have received funding from Helsinki Graduate School in Computer Science and Engineering, The Algorithmic Data Analysis (Algodan) Centre of Excellence, and the From Data to Knowledge (FDK) Centre of Excellence, which I gratefully acknowledge. I wish to thank the Department of Computer Science of the University of Helsinki for providing me with extremely good working conditions and superb computing facilities.

I would like to thank my parents and friends for the encouragement and the attempts to counter-balance the periods of seclusion. Finally, I thank my wife for her love, tireless support, and patience with the near-manic work hours and generally absent-minded husband. Thank you.

Contents

Chapter 1

Introduction

Topic detection and tracking (TDT) is an area of information retrieval research that focuses on news events. It comprises tasks of segmenting the transcribed broadcast news into cohesive stories, spotting something new, previously unreported, tracking the developments of the news events, and grouping together stories that discuss the same event [11]. These techniques benefit an information worker, an analyst, a specialist, or a reporter in keeping abreast with multiple sources of news ranging from radio and television broadcasts to on-line newswire, and in monitoring the daily flood of information.

TDT has been characterized as *event-based information organization* [9]. The target of interest revolves around events, that are often defined as "something happening somewhere at some time". In contrast to the more traditional information retrieval that deals with stable concepts, categories or static content-based topics, TDT is surrounded (or plagued) by change and uncertainty. The event-based topics are dynamic, have only few relevant documents, are mostly unpredictable, and have usually a short lifespan. A TDT system needs to distinguish between the same and similar events, i.e., different instances elections, riots, and train wrecks.

The initial efforts in TDT experimented with the traditional information retrieval techniques. After some initial success, the techniques were found inadequate in coping with events and new event detection especially. A work by Allan, Lavrenko, and Jin [16] suggested an upper bound for the performance of methods based on traditional full-text similarity. As a result, the researchers have sought new ways to exploit semantic, contextual, and temporal information.

1.1 Our contributions

This work is composed of four objectives that are motivated, experimented and evaluated in relation to the problems of topic tracking and first-story detection. In the former, the system is presented with one or more sample stories discussing a news topic, and the system then monitors the incoming news stream for further occurrences of the same topic. In the latter problem, the system tries to spot new, previously unreported events in the news stream.

First, we present a conceptual analysis of events and topics in TDT. There has been rigorous work on technical foundations and probabilistic modeling of TDT (see, e.g., [8, 10, 16, 29, 38, 50, 76, 79, 80, 83, 87, 105, 110, 137, 138, 144, 155, 157, 156, 159, 160]), but not of what the data represents. Our purpose is to weed out the inconsistencies in the terminology and to align the concepts with the process by which they are instantiated, that is, the making of news. We hope our work will shed some light on broader scope of news, text and information retrieval.

Second, we define a framework of document similarity that is based on semantic classes, i.e., groups of words with similar meaning. In some ways, the *similarity of hands* framework is a formalization of our previous work [91, 93, 94, 95]. It also generalizes numerous similar or analogous approaches reported by others [36, 40, 50, 56, 66, 76, 105, 138, 163], but it takes one step further. We need not deem words simply as strings of characters. The words in a semantics class can be compared with a class-specific similarity measure, like a distance in a phylogenetic tree, a taxonomy of concepts, music genres or word senses, a geographical distance, a distributional similarity, etc. So, this framework aims at ontology-based document similarity. Although the work is far from conclusive, we support our case with experiments. According to the results, the use of similarity of hands approach increases the accuracy of topic tracking 10-30% depending on the experimental setup. On the other hand, in first-story detection the results remained more modest, that is, only up to 3-4% with the largest evaluation data. The adoption of semantic classes requires tools with which to extract, normalize and disambiguate proper names and locations. The accuracy of our named entity normalization hovered close to 90% and disambiguation of locations around 80%.

Third, we incorporate time into the presented framework of document similarity. Although time can be seen as just another ontology, it is a defining feature in the concept of an event. The previous work in TDT has, for instance, used time to discount the similarity of two documents if their publication dates are far apart [21, 29, 83, 111, 158, 157]. Some have modified

the document similarity based on the similarity of temporal expressions in the documents [74, 85, 113]. Based on our earlier work [92], we outline a technique for formalizing recognized natural language temporal expressions that anchors the temporal expressions onto the time-line correctly 94% of the cases. Moreover, we introduce a novel document similarity that anchors each occurrence of each term onto the time-line. Upon comparing documents for event-based similarity, we look not only at matching terms, but we inspect the extent to which their temporal contexts agree. The terms in a document discussing the Iranian revolution are likely to be anchored into the late 1970s. Another document on Iranian politics or nuclear programs might have some overlapping terms, but unless the temporal context in which they occur refers to the late 1970s, the document is very likely to discuss events other than the Iranian revolution. This approach increases the accuracy of the topic tracking by another 10% over the semantic class approach with the largest test data. Again, the gain in first-story detection remains small compared to the semantic class approach, only about 1%.

Finally, we experiment with an adaptive variant of the similarity of hands system. The news reflect changes in the real world, and in order to keep up, the system has to change its behavior based on the contents of the news stream. We put forward two strategies for rebuilding the topic representations and report experiment results. The adaptation shows small but not significant improvement over the semantic class approach.

1.2 Organization of the thesis

This thesis is organized as follows. In Chapter 2, we shall start with broad strokes outlining information retrieval in general and techniques that are relevant to topic detection and tracking. TDT makes use of various techniques of text retrieval, document clustering, and information extraction. More importantly, we describe the experimental model for system evaluation along the guidelines of which the system performance is determined.

Then, we move on to topic detection and tracking providing a brief history and task descriptions. In this thesis we investigate the tasks of topic tracking and first-story detection. Given the special nature of TDT tasks, there has been a notable amount of work on system evaluation that we shall cover with some detail.

Chapter 3 goes through the previous work on topic tracking and first-story detection in detail. We start from the vector-space model that is a robust and widespread algebraic model for representing text for the purposes of efficient information storage and retrieval. We then describe a

more recent alternative called language model that uses statistics in assigning probabilities for strings of words. Finally, we present previous work that uses contextual, semantical or temporal information in document similarity.

We present our conceptual analysis of events and topics in Chapter 4. Then, before inquiring into the baseline systems, we portray the text corpora used in the experiments. We are employing three text collections annotated for the purposes of TDT. In addition, we have material for background statistics of newspaper English. The details of topic tracking and first-story detection are explained through baseline experiments. We conduct failure analysis to chart the sources of detection errors.

Chapter 5 deals with the similarity of hands framework. The chapter falls into roughly three parts. First, we build the framework with a series of definitions that ultimately form the document similarity approach. Then, as the experiments rely heavily on named entities, we describe the preprocessing of documents. The process involves several steps of recognition, normalization, and disambiguation. Finally, we report the experiments with semantic classes.

The work on temporal information is presented in Chapter 6. In order to augment document similarity with time, the temporal expressions occurring in the corpus have to be recognized and formalized. Our recognition runs on finite-state automata, and the formalization employs a global timeline and a calendar algebra. We investigate three different approaches to incorporating time into the similarity of hands framework. The simplest is time-decay that decreases the document similarity in proportion to the span between the publication dates. Another approach identifies temporal expressions as terms that compose a semantic class of its own. The third approach that we call *Davidsonian indexing* is motivated by Davidson's logical analysis of action sentences that we discuss in Chapter 4. We run experiments both in temporal expression preprocessing as well as in topic tracking and first-story detection.

Chapter 7 presents adaptive topic tracking. We outline two strategies for rebuilding the topic representation, and experiment them with semantic classes and temporal information.

Finally, Chapter 8 is the conclusion.

Chapter 2

Background

In this chapter, we shall first look at some information retrieval techniques that are relevant to TDT in Section 2.1. Then, in Section 2.2, we present the overall ideas of TDT, the research problems or tasks involved, and the evaluation process by which the system performance is measured.

2.1 Information retrieval

Information retrieval is concerned with representing, storing and organizing information for easy access [24]. It can be understood as a process that tries to find "material of an unstructured nature that satisfies an information need from within large collections" [99]. Typically, this unstructured material is text and the large collections are stored and organized on a computer. The *information need* is understood simply as the topic one is interested about, and is often expressed as a *query*, e.g., a set of keywords. An information retrieval system then attempts to meet the information need by matching the given query to a document collection. As a result, a list of "hits" or "matches" is returned, that is, documents that are assumed to best satisfy the information need.

In this section, we will go through some of the essential information retrieval techniques relevant to TDT. These are text retrieval in Section 2.1.1, information filtering in Section 2.1.2, document clustering in Section 2.1.3, and information extraction in Section 2.1.4, respectively. Scientific research relies on systematic experiments, and in Section 2.1.5 we will present how information retrieval systems are evaluated.

2.1.1 Text retrieval

Text retrieval (also called ad hoc retrieval, full-text search) boils down to two fundamental problems of computer science: *searching* and *sorting*. Given a specific information request, a list of keywords, for instance, text retrieval system first searches for candidate answers – documents in which keywords occur – and then sorts the list of candidates using some relevance estimate, e.g., rewarding documents for multiple occurrences of otherwise rare keywords. The focus of interest is typically a *document*, i.e., a cohesive piece of text like a news story, article, book chapter, or web page [99].

Text retrieval usually employs the *bag-of-words* model. The documents are considered sets of words, and structures like word order, sentences and grammar are neglected. The term frequency in the document is, however, recorded [99]. Although it is a dramatic simplification of the text, it enables an efficient representation of documents as vectors in a vector-space, where each distinct word in the document collection is associated with a dimension. For all words in the document, the associated dimension in the document vector is assigned a non-zero value and documents can be compared with using standard linear algebra.

By itself, text retrieval is used to address the full-text search queries like web search, which is a substantially different problem from TDT. However, many TDT systems store the documents in full-text retrieval systems, because they provide an efficient way to search and sort documents in a large collection. In this case, similarly to an index in a relational database, a basic text retrieval system works as a short-cut to data items with certain attribute values.

2.1.2 Information filtering

TDT systems are mostly about monitoring: news documents come in, and the system needs to determine which of them discuss some new topic and which relate to some previously recognized topic. The news monitoring can be understood as a form of *information filtering*, where just as in text retrieval the information need is expressed as a query, but the given query is executed repeatedly on the incoming document stream. This makes information filtering basically a classification task that assigns items to predefined classes based on the queries. Although filtering assumes that the information need is persistent and does not change between documents, the query itself may evolve or adapt over time making the system dynamic [27].

Information filtering techniques have been used in removing redundant or unwanted information from a data stream. In spam-filtering the infor-

mation need is simply to spot junk mail from the incoming email. In text routing, documents are delivered out on basis of profiles. For instance, a company might want to route customer correspondence automatically to billing, technical support or marketing departments depending on the contents of the emails. In a similar vein, a personalized news service could deliver news based on the similarity between user profiles and the news content.

Yet, whereas information filtering typically assumes the information need remains stable, the need is never explicitly expressed in spotting new topics. The classification judgment of a new document relies on its similarity or dissimilarity to the previous documents. Spotting new topics means spotting something sufficiently different from all the previous topics. Once a new topic is spotted, it is added to the system as a new filter or a class.

2.1.3 Document clustering

Cluster hypothesis states that closely associated documents tend to be relevant to the same queries and ultimately address the same information need [148]. In simple terms, closely associated documents have substantial amount of same words, and thus they are likely to discuss the same thing. This gives rise to *document clustering*, cluster-based information retrieval that organizes the document collection into groups of closely associated documents. The clusters are not known a priori, and so the technique not only assigns documents to clusters but also creates the clusters themselves. The outcome depends on the distribution and makeup of the data. Because there is no human involvement, the technique is a form of unsupervised learning.

There are two kinds of clustering approaches. *Hierarchical clustering* creates relationships between clusters, while *flat clustering* does not [99]. To form a cluster hierarchy the methods need to process the collection as a whole, which is not feasible for online experiments. Instead, online methods resort to *single-pass clustering* [148] that creates a new cluster, if and only if the new item is not sufficiently similar to any of the previously created clusters. The judgment is done upon the arrival of a new document, and the result is an ever growing and flat set of topic clusters.

The clusters can be represented by *centroid vectors* that are averages of the documents in the cluster. In *nearest neighbor* comparison, there are no topic models per se; the incoming documents are compared directly to the previous documents and if they are found similar enough, i.e., their similarity comparison yields a high enough score, the document is considered to belong to the cluster. Suitable thresholds for similarity scores are obtained

from test runs with training data.

2.1.4 Information extraction

Information extraction comprises methods that refine machine-readable unstructured text automatically. An information extraction system skims through the input, and when it encounters an instance of pre-specified information, e.g., a word or a phrase matching a pattern, it tags the instance as an occurrence of a feature. The features are typically proper nouns, locations, currencies or parts-of-speech, but they can be virtually any phrasal units [42]. For instance, *named entity recognition* is a form of information extraction that attempts to spot all the named entities in the text, i.e., to spot the names of persons, organizations, locations, expressions of times, quantities, monetary values, and percentages.

The extracted features can then be used to summarize the content, to automatically place the events described in the text on a map or a timeline, or to construct a representation of the text suitable for a structured database. The results of information extraction have been also used widely used in finding good indexing terms for information retrieval [100]. By adopting multi-word features, such as locations or names of companies, or more complex features, such as temporal expressions or relationships between people and organizations, one effectively extends the simple bag-of-words model of information retrieval.

Information extraction relies heavily on the techniques of *natural language processing* (NLP). A natural language is simply a language humans speak (as opposed to formal languages like first-order predicate calculus or computer programming languages), and NLP is an area of research dealing with the language-based interaction of computers and humans [72]. Typical techniques comprise *tokenization*, with which the stream of text is segmented into sentences and words, and *tagging* that associates each word with an appropriate part-of-speech tag. On the sentence-level, NLP techniques are used to find the syntactical relationships between words. This information is then used in building information extraction patterns and rules.

Many TDT systems make a wide use of information extraction techniques. The extracted features, like company names, are used to boost some aspect of the document clustering or filtering.

2.1.5 System evaluation

Evaluation is simply a process used to assess, how well a system meets the information need of a user. A *user-based evaluation* involving a group of real users using the system is often beyond the practical means or the budget of the research project, and researchers resort to *system evaluation* that is a laboratory abstraction of the retrieval process without the involvement of the actual user. This experimental model in information retrieval, sometimes called the *Cranfield paradigm* [149], contains three components: a collection of text documents, a set of queries, and a pre-defined list of corresponding relevance judgments. The text collection or *text corpus* is read into the system, the set of queries are executed and the system output is compared to the relevance judgments, i.e., a list of which documents should be retrieved for which query.

The performance of an information system is measured in terms of *efficiency* or *effectiveness* [148]. A high efficiency means the execution uses only little CPU and memory. The effectiveness, on the other hand, portrays the similarity between the system's output and the relevance judgments, and a highly effective system would produce judgments similar to the pre-defined list. There are many ways to quantify the similarity of the lists. Some ways are coined into functions, *evaluation measures*, to enable cross-system comparisons.

The most common effectiveness measures are based on the outcomes portrayed in Table 2.1 (see e.g., [141]). For example, A stands for the number of instances that were both relevant and retrieved, the sum $A + C$ the number of retrieved instances and $A + B$ the relevant instances, respectively. Two measures that have become somewhat standard are *precision* and *recall* [73]. They are defined as

$$\begin{aligned} \text{precision} &= \tfrac{A}{A+C}, \\ \text{recall} &= \tfrac{A}{A+B}. \end{aligned}$$

Precision is the proportion of retrieved instances that are relevant, and recall is the proportion of relevant instances that are retrieved. We denote the set of relevant documents by ω and the set of retrieved documents by r. Their complements are denoted by overlines, i.e., $\overline{\omega}$ stands for the set of non-relevant documents and $\overline{r}$ for the set of documents that were not retrieved. Now, precision and recall may be taken as estimates of conditional probabilities, i.e., the probability of correct judgments on relevant retrieved documents ($p(\omega|r)$) and the probability that relevant documents are retrieved ($p(r|\omega)$), respectively.

There is a dependency between these two measures: one can obtain a

Table 2.1: The 2×2 contingency table of possible retrieval outcomes.

		System response	
		retrieved (r)	*not retrieved* ($\overline{r}$)
Corpus annotation	*relevant* (ω)	A	B
	non-relevant ($\overline{\omega}$)	C	D

perfect recall by retrieving all the documents and a perfect precision by retrieving just one document, assuming it is relevant. In the former case, the effectiveness is obtained at the cost of extremely low precision, and in the latter at the cost of minimal recall. This trade-off complicates cross-system comparisons. In order to rank systems' performances, precision and recall need to be combined into, for instance, *F_β-measure* [148],

$$F_\beta = \frac{(\beta^2 + 1)\, precision \cdot recall}{\beta^2\, (precision + recall)}, \tag{2.1}$$

where the variable $\beta \in [0, \infty]$ represents the relative importance of precision and recall. The measure is probably best known as F_1-measure that sets β to 1 yielding a harmonic mean of recall and precision. If either precision or recall is close to zero, F_1 will be very small as well.

The effectiveness of a system rests upon the elusive notion of *relevance*: a system is effective if it provides the user with relevant information. Despite the inherent subjectivity of relevance, Saracevic, for instance, maintains that there is a lot a group of judges would agree on [125]. The relative instability of relevance judgments is small and does not invalidate test experiments, if they are produced carefully by a group of human experts as the Cranfield paradigm suggests.

Creating text collections with relevance judgments is arduous and often unfeasible for small research groups. The Text REtrieval Conference [1] (TREC) was established to support evaluation of information retrieval systems with real-world large-scale data sets and consistent, uniform evaluation techniques following the Cranfield paradigm. Although TDT is not a research track sponsored by TREC, the text collections are produced by the Linguistic Data Consortium [2] (LDC), the producer of many TREC corpora. We shall describe the TDT corpora in detail in Section 4.2.

[1] http://trec.nist.gov/

[2] http://www.ldc.upenn.edu/

2.2 Topic detection and tracking

Topic detection and tracking (TDT) consists of several information retrieval tasks. The unstructured material it deals with consists of newswire and possibly transcribed audio. The information need addressed revolves around the notion of *event*. The goal of TDT is then to "break the text down into individual news stories, to monitor the stories for events that have not been seen before, and to gather the stories into groups that each discuss a single news topic" [9]. All of these goals are pursued with basic information retrieval techniques like information filtering, information extraction and document clustering. What makes TDT special is its event-based focus: a large collection contains only very few documents that meet the information need, that is, discuss the given event. Making the fine distinction between documents discussing same and similar events is characteristic to TDT.

2.2.1 Motivation and brief history of TDT

The initial motivation for the TDT research initiative was to provide core technology for a news monitoring tool. Such a tool would help an information worker, an analyst or a specialist keeping abreast with the enormous volume of news data by monitoring newswire and news broadcasts, alerting to new, interesting things taking place, and organizing the news stories into events. The information worker might wish to follow the course of events regarding bush fires in Australia, the development of the presidential elections in France, or just be informed if anything new takes place in Portugal or in the metal industry, for example. Given a news story, a TDT system would have to be able to attach it to any previous discussions about the event portrayed in the story – else the story would be regarded as new.

Topic detection and tracking is characterized by a tighter notion of relevance or "aboutness". In contrast to traditional subject-based topics, here the news stories are related to each other through a seminal real-world event. As an event is marked by location and usually actors or participants, event-based topics are more limited in scope. More importantly, events happen at some specific time, and while subject-based topics are hardly affected by temporal issues, they are crucial in TDT. We shall analyze the event definitions in Section 4.1.

TDT research began as a pilot project funded by the U.S. Government's Defense Advanced Research Project Agency (DARPA) in 1997. The pilot project was followed by yearly benchmark evaluations from 1998 to 2004. Five annotated text corpora were produced for the purpose of these evalu-

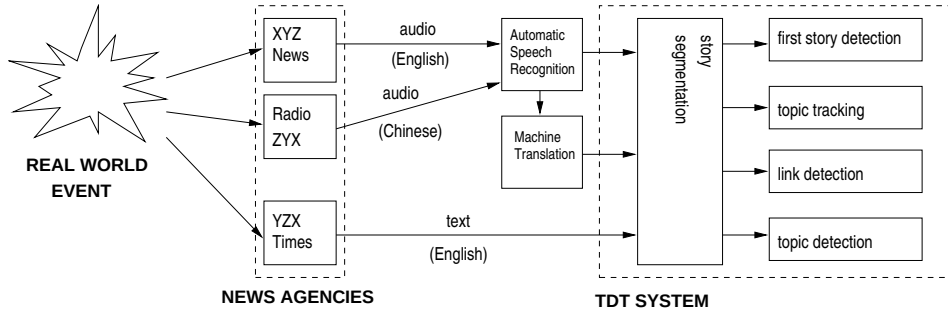

Figure 2.1: The tasks in TDT. A real world event is reported by various news agencies, possibly in different languages and media. The news data is possibly transcribed and translated to English, and then segmented to cohesive news stories and fed to the event-based organization tasks.

ations. Each corpus introduced a larger volume of data. The first corpus consisted only of 15,000 documents while the most recent one contains over 400,000 documents. Furthermore, from 1999 onwards the corpora were multi-lingual. Chinese news sources were introduced in 1999 followed by Arabic in 2002.

The pilot project experimented the use of traditional information retrieval techniques with some success. Then, the research turned to deal with problems characteristic to TDT: the small number of relevant documents, evolving vocabulary, and time-relatedness of topics. In 2000, Allan, Lavrenko and Jin presented a upper-bound for full-text similarity in spotting new topics [16]. After that the trends in TDT have shifted its efforts towards using semantical and contextual information. We will present the previous work in Section 3 with more detail.

2.2.2 Task definitions

There are five tasks in topic detection and tracking. They have been renamed and modified over the years. The tasks are illustrated by the dashed box in Figure 2.1. When something news-worthy happens in the world, it is reported by news agencies. A TDT system monitors the news agency data stream. If it is audio, it needs to be transcribed to text. In addition, many systems are monolingual internally, and thus non-English material needs to be translated to English by machine translation.

Story segmentation is a process by which the content of the input streams is divided into news stories. The news-wire material has the doc-

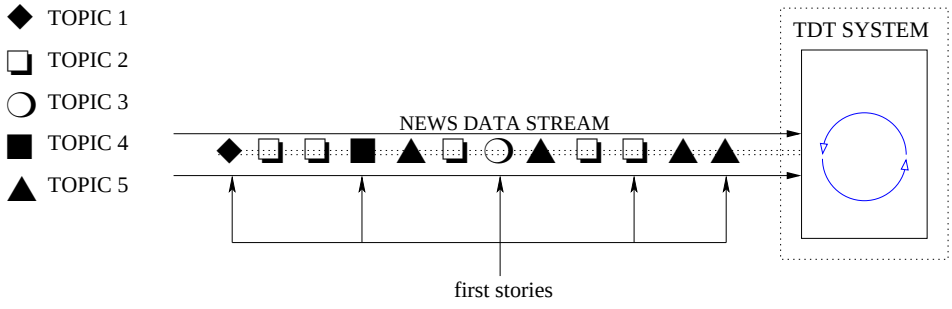

Figure 2.2: The task of spotting first-stories from a data stream. TDT system reads a news data stream composed of news stories. Here, the stories are illustrated by various shapes depending on to which topic they belong.

ument boundary markup inherently, but the transcribed speech is more or less a continuous stream of words without explicit breaks between stories. For evaluation purposes, the document boundaries are usually removed from the newswire material as well. The four other tasks require the document boundary information, so ideally the story segmentation is executed before any of the others. We are not investigating story segmentation, and therefore we make use of the annotated document boundaries.

The aim of *first-story detection* (FSD, also *new event detection*) is to recognize, whether the topic discussed in the document is reported for the first time or not. Figure 2.2 illustrates the idea of FSD. TDT system reads a data stream, and on arrival of a new story, a binary decision is made whether the story discusses some old, previously recognized topic or something new and previously unreported.

Since there is no a priori knowledge about the future topics, the notion of 'new' boils down to something sufficiently different from the 'old'. As a result, each incoming document is compared to all previous ones. FSD is thus a form of online clustering, where a new cluster is created if the new item is not sufficiently similar to any of the old. FSD has been characterized as 'query-free retrieval', because the information need cannot be expressed as an explicit query [11]. FSD is one of the problems we investigate in this work.

Topic tracking is a form of information filtering: provided a small number of sample documents, the system filters the relevant news stories from the input stream by labeling each incoming document as relevant or non-relevant. This resembles a situation, where the user encounters an inter-

esting news story and wishes to follow the development of the event it describes. The relevancy to the target topic is resolved by comparing the new document to the topic representation; if similar enough, the new document is considered relevant. If multiple topics are tracked simultaneously, they are run independently of each other. Thus, it is possible that a document is found relevant to more than one topics. Some recent evaluation settings for adaptive topic tracking have allowed relevance feedback. When the system makes positive judgment (deciding the document is on-topic), it is given the actual label of the document. This simulates actual use of the system, where the user is prepared to review the positive judgments [159].

In *cluster detection* (also *topic detection*), the tasks of first story detection and tracking are both run on-line. All the topics that FSD recognizes are tracked, even if no user has expressed an interest in them. The cluster detection simulates a large-scale news monitoring system, addressing the needs of various information specialists, business analysts and reporters. The evaluation is slightly different from FSD and tracking. The system is not severely penalized for missing the first actual story as long as the topic is detected at some point.

There have been two kinds of cluster detection experiments. *Online clustering* makes judgments as documents arrive. In *retrospective clustering* the system is presented with the whole document collection at once, the task is to organize the collection into topic clusters. In early experiments, clusters were assumed to be flat, but later *hierarchical clustering* conceded relationships between clusters [5].

The goal of *story link detection* (SLD) is to determine whether two given documents discuss the same topic or not. It is a classification task to test the pair-wise document similarity measures. As such, the task alone is not useful in practice, but the notion of topic-based similarity in FSD, cluster detection, and topic tracking is based on the problem similar to SLD.

2.2.3 TDT evaluation

In the TDT experimental setup there are no user-defined queries. All the tasks are about detection: given a stream of news documents, the system has to pick out relevant ones. Here, relevance does not hinge upon explicitly stated information need, but on the previous content of the stream. To this end, the focus of the evaluation in TDT shifts from measuring precision and recall to recording two kinds of errors, *misses* and *false-alarms*. Miss rate indicates the proportion of relevant documents that were not retrieved with respect to all relevant documents, and false-alarm rate the proportion of retrieved non-relevant documents with respect to all non-relevant

documents [54]. Based on Table 2.1, we can define

$$\begin{aligned}\text{miss} &= \tfrac{B}{A+B},\\ \text{false alarm} &= \tfrac{C}{C+D}.\end{aligned}$$

Similarly to precision and recall in Section 2.1.5, miss and false alarm rates can be understood as estimates of conditional probabilities: miss is the probability that a relevant document is not retrieved, $p\,(\overline{r}|\omega)$, and false alarm is the probability that given a non-relevant document, it is retrieved, $p\,(r|\overline{\omega})$.

The error-rates can be calculated in terms of topics or stories. The former, *topic-weighted* rate (macro average), means that the probabilities of misses and false-alarms are first determined for each topic, and then averaged over all of the topics. Each topic has equal weight regardless of how many documents they contain. The *story-weighted* rates (micro average) consider each document equally important, and averages directly over all of the judgments. As a result, there is a bias towards large topics: low error-rates in topics with a large number of documents imply good overall performance. Unless explicitly stated otherwise, we report the topic-weighted error-rates.

Similarly to precision and recall, miss and false-alarm rates have a trade-off. By assigning all documents as relevant, one achieves a zero miss rate, but a maximum false-alarm rate, and vice versa. For comparing the effectiveness of two TDT systems, two methods are widely used: the *detection cost function* [54] and the *detection-error trade-off* (DET) curves [102]. The detection cost function C_{det} is a linear combination of the two kinds of error rates,

$$C_{det} \;=\; c_m \cdot p\,(\overline{r}|\omega) \cdot p\,(\omega) + c_{fa} \cdot p\,(r|\overline{\omega}) \cdot (1 - p\,(\omega)), \qquad (2.2)$$

where c_m is the cost of a miss, c_{fa} the cost of a false-alarm, $p\,(\overline{r}|\omega)$ and $p\,(r|\overline{\omega})$ the conditional probabilities of miss and false-alarm, and $p(\omega)$ the *a priori* probability of a *target*, i.e., a document that is relevant to a given topic. By convention, c_m is set to 1.0, c_{fa} to 0.1 and $p(\omega)$ to 0.02 [3].

The detection cost is normalized by dividing it with the minimum of the expected cost when considering all documents either targets or non-targets:

$$(C_{det})_{norm} \;=\; \frac{C_{det}}{\min(c_m \cdot p(\omega),\; c_{fa} \cdot (1 - p(\omega)))}. \qquad (2.3)$$

[3] In early experiments, c_m and c_{fa} were both set to 1.0, which may impede comparing directly the reported results.

While a zero cost stands for a perfect effectiveness, a cost 1.0 would now present a performance that is no better than judging 'yes' or 'no' consistently in all judgments.

The TDT cost function has been criticized for being dependent on the size of topics rather than being a general function [98]. As the topics contain a small number of documents, the number of targets is by far smaller than non-targets. Though the function compensates the unbalance between targets and non-targets, it does not consider different sizes of the topics. In other words, as $p(\omega)$ is set to 0.02, it does not portray the real sizes of topics. For instance, in TDT3 corpus the true percentage of targets is 0.002 instead of 0.02. Yang et al. note that this leads the cost functions to favor recall-oriented systems, as this amplifies the penalty cost ratios between misses and false-alarms from $10:1$ to $100:1$ and for TDT5 an extreme of $583:1$ [159]. As a result, any optimization of parameters based on the detection cost function has bias arising from the test corpus. Still, we report $(C_{det})_{norm}$ costs as they are defined above to maintain a consistency between previous work.

For each input document, a TDT system produces a likelihood score that the document is a target. The scale of the scores can be anything but it should be consistent across decisions and greater values should indicate higher likelihood. By setting a threshold θ, one can convert the likelihood scores to binary decisions: scores greater than θ stand for "similar", and scores equal to or less than θ stand for "not-similar", respectively. Assuming the decision scores within targets and non-targets yield normal distributions, the corresponding curves would look something like in Figure 2.3. The distributions of targets, i.e., $p(x|\omega)$, and non-targets, i.e., $p(x|\overline{\omega})$, have means μ_ω and $\mu_{\overline{\omega}}$, and standard deviations σ_ω and $\sigma_{\overline{\omega}}$. In the figure, the deviations of the two distributions are equal. The lightly hatched area to the left of the threshold θ depicts targets that were regarded dissimilar, i.e., the probability of a miss. Similarly, the solid shaded area to the right of the threshold θ stands for non-targets that were erroneously found similar, so the area stands for the probability of a false-alarm. Now, the probabilities of miss and false alarm can be expressed as integrals over the two distributions,

$$\text{miss probability } p(\overline{r}|\omega) = \int_{-\infty}^{\theta} p(x|\omega)dx, \tag{2.4}$$

$$\text{false-alarm probability } p(\overline{\omega}|r) = \int_{\theta}^{\infty} p(x|\overline{\omega})dx. \tag{2.5}$$

Obviously, decreasing the value of θ reduces the probability of a miss, but it also increases the probability of a false alarm. To visualize the

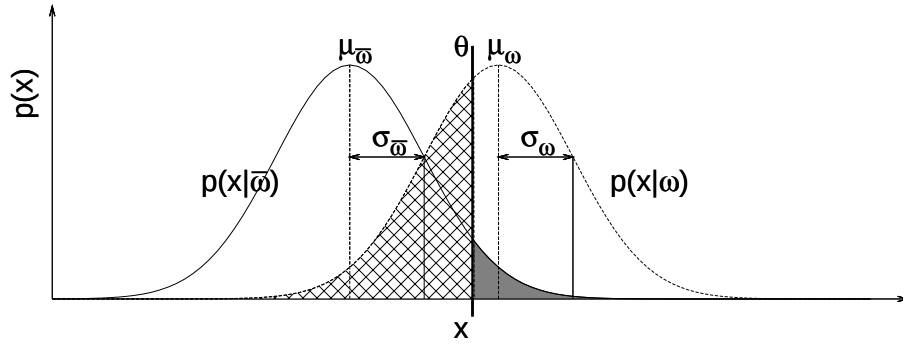

Figure 2.3: Examples of normal distributions for decision score x given classes of targets ω and non-targets $\overline{\omega}$. The score distribution of targets, $p(x|\omega)$, has a mean μ_ω and a standard deviation σ_ω (similarly, a mean $\mu_{\overline{\omega}}$ and a standard deviation $\sigma_{\overline{\omega}}$ for non-targets). The line at θ represents a decision threshold score. The shaded areas to the left and to the right of θ illustrate the probabilities of miss and false alarm, i.e., $p(\overline{r}|\omega)$ and $p(\overline{\omega}|r)$.

trade-off between the two kinds of errors, a DET curve combines the two distributions of Figure 2.3 into one by iterating the threshold θ through the space of decisions scores, and at each point estimating the two error probabilities, given the system output. Thus, a DET curve represents the effectiveness of a system throughout the space of decision scores, i.e., for all possible thresholds θ. As an example, Figure 2.4 depicts DET curves of two systems. Instead of plotting the miss and false-alarm probabilities themselves, DET curves show their standard deviates. The "toy" curves are ideal in the sense that they are linear. This means that the decision scores are distributed normally. Also, the plots have equal slopes due to assumed equal standard deviations.

The higher the effectiveness is, the fewer errors there are, and the closer the curves are to the lower left corner. Thus, in Figure 2.4, the effectiveness of System 1 is consistently better than that of System 2. In the same figure, the marked spot labeled θ denotes a decision threshold score for System 1 that produces a miss rate of 38 % and a false-alarm rate of 5 %. To attain the same miss rate, System 2 would have a false alarm rate of 67%.

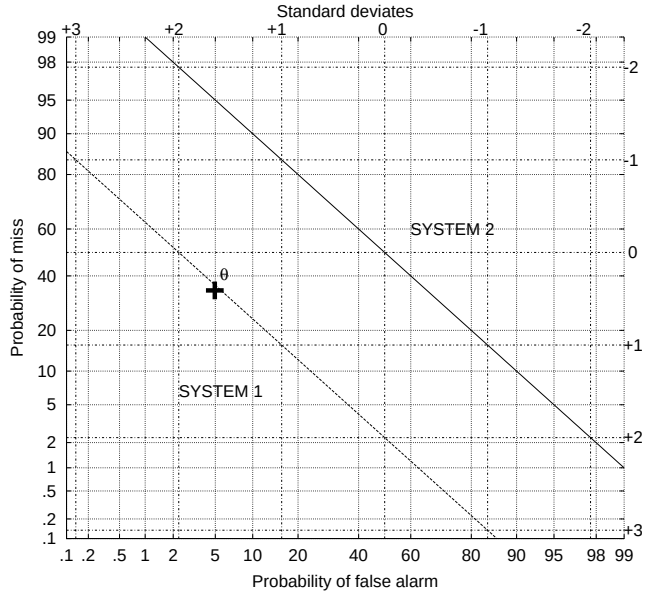

Figure 2.4: Examples of detection trade-off (DET) curves of two TDT systems [102]. The curves are obtained by iterating the threshold θ through the systems' output data, that is, the set of decisions scores. The probabilities of miss and false-alarm are estimated at each step and plotted on a standard deviate scale instead of a linear one. The probabilities of false-alarm and miss are on the left and bottom. The corresponding deviates are labeled on the right and on the top. The example threshold θ in the curve of System 1 represents the threshold of Figure 2.3 that yields 38 % miss rate and 5 % false-alarm rate.

In linear scale graphs, the plotted lines of well performing systems have a tendency to overlap. The advantage of the DET curve is that with deviate scale the plot spreads out and expands the high-performance area and thus better distinguishes between the systems. Moreover, as they show how false-alarms and misses vary with respect to each other across the space of decision scores, the overall performance is better displayed than with just one threshold. As it was said, a good recall (low miss-rate) can be achieved at the expense of low precision (high false-alarm rate). Suppose an analyst has a "knob" in his or her TDT client to regulate this trade-off. A DET curve plots the performance for different positions of this "knob" [56]. More importantly, the comparison of systems is not greatly affected by the choice of the threshold.

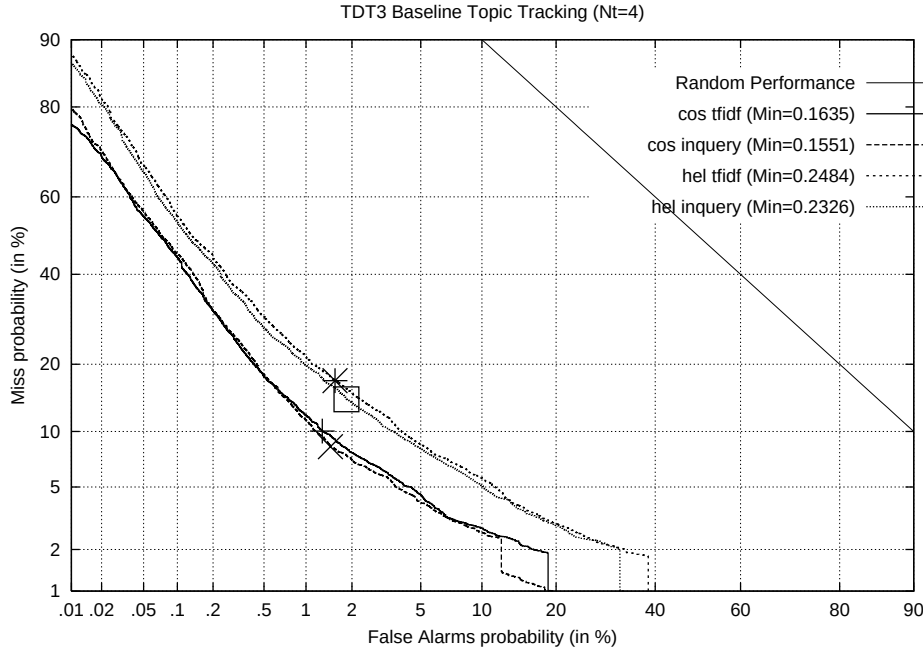

Figure 2.5: An example of four topic tracking DET curves. The straight line in the upper right corner represents the performance of a system that makes either YES or NO decisions consistently for all incoming documents.

Figure 2.5 illustrates actual DET curves for four topic tracking systems. Unlike in the examples above, the real data does not yield straight lines implying the underlying distributions are not strictly normal. In addition, the two axes are not of equal range. The miss probabilities are not plotted below 1% while false alarms go to one-hundredth of a percent. This is due to the asymmetrical portions of targets and non-targets, and therefore, of misses and false-alarms. The DET curve represents the topic-weighted cost. In TDT-2 corpus there are less than 100 and in TDT-3 barely over 100 topics, so there is no data to plot for topic-averaged misses below 1%. On the other hand, 1% false alarm rate in TDT-3 translates to 35,500 documents. We shall present the evaluation corpora in Section 4.2.

The plots contain a marked spot where the cost function reaches its minimum value. The minimum cost for each run is also indicated in the list of keys in the upper right-hand corner. The straight line labeled “random” traveling across the upper right-hand corner represents the performance of a system that answers either YES or NO consistently for all documents. So,

it is not really random, but rather the best performance without inspecting any of the documents.

The DET curve approach stems from *relative operating characteristic* (ROC). It was originally used in detection of radio signals in the presence of noise, and later in evaluation of information systems [141, 142].

Chapter 3

Topic tracking and first-story detection

> *I have omitted all those things which I have not myself seen, or have not read or heard of persons upon whom I can rely. That which I have neither seen, nor carefully considered after reading or hearing of, I have not written about.*
> – Gregorius Agricola, *De re metallica*

This chapter outlines the main directions of previous work in topic tracking and first-story detection. Section 3.1 describes one of the most widely used retrieval model, the vector-space model, and its application in topic tracking and first-story detection. In Section 3.2, we discuss the work on language models in TDT. There has also been a substantial body of work using semantical or contextual information in TDT, and we discuss that in Section 3.3. Finally, we lay out our approach in Section 3.4.

3.1 Vector-space model approaches

A digital document can be text, speech, image or multimedia, all of which are represented as byte-sequences. In this work, we are assuming the documents TDT system is dealing with are ultimately text, so transformation of the audio signal to text is outside the scope of our investigation. From whatever origin, a document of text is fundamentally a string of characters. Comparing documents by exhaustive string matching is not feasible. Therefore the text is transformed into a computationally efficient form.

The *vector-space model* associates each word in the text with a dimension in the vector-space [122]. It is a bag-of-words approach that neglects word order and syntax. Occurrences of a word in a document are assigned

weights based on the informativeness of the word and number of occurrences in the document. Written text displays a great deal of variation arising from morphology and typography, for instance. In order to decrease the number of dimensions in the vector-space, we reduce the variation by preprocessing the text.

If we deem information retrieval merely as searching and sorting, then sorting rests on suitable similarity comparison. In the vector-space model, the similarity is often thought of as a geometric distance; objects that reside in proximity to one another are thought to be more similar than those further apart. We have chosen the vector-space model as the baseline system for a number of reasons: it is robust, it is fairly simple to implement, and it has proven to be highly competitive compared to the more elaborate systems. The results by Allan et al. [16] provide an upper bound for first story detection systems based on full-text similarity.

3.1.1 Document preprocessing

Initially, a TDT system perceives a text document as a long character string. In order to transform the document into a vector, the system needs to identify and normalize words and then map them into a linear space. Salton suggested a five-step process for this purpose [123].

1. Identify individual words and reduce the typographical variation.
2. Remove non-informative words.
3. Reduce morphological variation.
4. Compute the term-weights.
5. Build the vector.

The phases of the process are illustrated in Figure 3.1. The first phase is called *tokenization*. It identifies words in the text by some word boundary (typically a whitespace or some non-letter character), removes punctuation and produces a list of *tokens*, i.e., instances of character sequences. The class of tokens containing the same character sequence is called a *type*. The tokens are *normalized* to remove the typographical variation by replacing upper-case letters with lower-case equivalents so that occurrences *the*, *The* and *THE* would be identified as the same type. In addition, we want to revert tokens like *U.S.A.* - *USA* and *cliche* and *cliché* to a single type, e.g., *USA* and *cliche*, respectively. In baseline case-folding, we simply lower-case everything except for abbreviations, e.g., *US*, *USA*, *NATO*, and *UN*, as

original text:
A painting by Claude Monet sold for a record 19.8 million pounds (dlrs 32.9 million) Tuesday, Sotheby's auction house said.
tokenized:
a painting by claude monet sold for a record 19.8 million pounds dlrs 32.9 million tuesday sotheby auction house said
stop-words removed:
painting claude monet sold record million pounds dlrs million tuesday sotheby auction house
stemmed:
paint claud monet sold record million pound dlr million tuesdai sothebi auction hous
simple term-weights:
paint:1 claud:1 monet:1 sold:1 record:1 million:2 pound:1 dlr:1 tuesdai:1 sothebi:1 auction:1 hous:1
simple termvector:
1:1 22:1 31:1 212:1 511:1 567:2 1920:1 3212:1 3390:1 12321:1 15929:1 15930:1

Figure 3.1: An example of document preprocessing. The original text is first tokenized, then stop-words are removed, then the words are stemmed transforming them into terms. The term-weights are simply frequencies in the text. The terms are then transformed into term-id - weight pairs, with the term-ids adopted from a larger lexicon.

some of them may be lost in later processing. For instance, lower-casing the token *US* results in type *us*. Since the pronoun *us* reverts to the same type, the two are confused and we lose the knowledge about the abbreviation. Identifying proper nouns will be revisited later on in Section 5.5.2.

There are highly frequent words occurring in all kinds of documents that are therefore irrelevant in representing the content. For instance, conjunctives (e.g., *and*, *or*, *but*), prepositions (e.g., *of*, *to*, *from*), articles (e.g., *a*, *an*, *the*) and pronouns (e.g., *that*, *she*, *it*) have a function in a sentence, but due to their high frequency and thus low specificity they work poorly in retrieval tasks. These words are called *stop-words* and a list of such words a *stop-list*. The second step excludes stop-words from the representation thus decreasing the number of dimensions in the vector-space, and making the document representations more compact and the retrieval process more efficient without a loss in effectiveness. Furthermore, we remove all numbers from the documents, because they have no impact on the effectiveness [13]. We do, however, store the publication date as metadata.

On the surface, the same word may vary as a result of inflection, e.g., *dog* - *dogs* and *bark* - *barked* - *barking*. In addition, there are closely related

words produced by word derivation, e.g., *president* - *presidential*. The purpose of the third step is to reduce this morphological variation for increased recall and more compact document representations. There are two alternative methods. *Stemming* is a simple and brutal heuristic that eliminates certain frequent suffices from the words. One of the most commonly used stemmer for English is by Porter [116]. *Lemmatization* reverts the word to its dictionary form, *lemma*, by syntactic and morphological analysis [100]. Regardless of which method we use, the result is what we call a *term*, a type without typographical or morphological variation.

In the fourth step, each term is assigned a real-valued weight. There are many ways to weigh the term, but for our baseline we shall adopt a commonly used scheme called TFIDF [124]. If a term occurs multiple times in a document, it is likely to be important. Thus, the term frequency (TF) represents the degree to which the term describes the content of the document. It can be a raw count or a logarithm

$$tf_d(t) = 1 + \log f_t, \tag{3.1}$$

where f_t is the number of times the term t occurs in document d. This count is often augmented with a coefficient called *inverted document frequency* (IDF) [136]. IDF associates the informativeness of a term with its specificity, which in turn is understood via the number of documents the term occurs in. Vague terms occur in a large number of documents; they are less specific and thus less valuable in determining the similarity of two documents. The inverted document frequency is a formula

$$idf(t) = \log\left(1 + \frac{m}{d_t}\right), \tag{3.2}$$

where d_t is the number of documents in which term t occurs and m is the total number of documents. TFIDF combines the term frequency in the document with the inverted document frequency of the term by multiplication as follows

$$tfidf_d(t) = tf_d(t) \cdot idf(t). \tag{3.3}$$

Naturally, there are numerous variants of TFIDF scheme [99].

Finally, each unique term is associated with a dimension in a vector-space. Now the documents can be represented as vectors.

3.1.2 Vector-space model

A *text corpus* (or corpus) is a collection of documents, and now after preprocessing the documents consist of terms. The compilation of all

terms occurring in the corpus is called a *term space*. Given a corpus $\mathcal{D} = \{D_1, D_2, \ldots, D_m\}$ of m documents and a term space $\mathcal{T} = \{t_1, t_2, \ldots, t_n\}$ of n terms, we can express the document collection as a $m \times n$ *document-term description matrix* $\mathbf{D}$, where each row $\mathbf{d_i}$ is an n-dimensional vector having non-zero values for terms that occur in the document D_i, i.e., $\mathbf{d_i}$ is the *document vector* $\mathbf{d_i} = (d_{i1}, d_{i2}, \ldots, d_{in})$ of document D_i. The value d_{ij} stands for the relative significance of the term t_j in document D_i, which in baseline system is expressed using a TFIDF weight.

At its simplest, the retrieval of information equals to returning a set of documents based on the given query expressed as a set of terms Q. Similarly to documents, the query can be expressed as an n-vector $\mathbf{q} = (q_1, q_2, \ldots, q_n)$. Now, the retrieval operation boils down to a multiplication

$$\mathbf{r} = \mathbf{Dq}, \tag{3.4}$$

where $\mathbf{r}$ is an m-vector expressing the degree of association or similarity between the query $\mathbf{q}$ and each document vector $\mathbf{d_i}$ [122]. The index of the maximal element in $\mathbf{r}$ refers to the most relevant document. In this simple model, the similarity of the document and the query is the inner-product of their vectors,

$$sim_{dot}(\mathbf{d_i}, \mathbf{q}) = \langle \mathbf{d_i}, \mathbf{q} \rangle = \sum_{k=1}^{n} d_{ik} q_k. \tag{3.5}$$

With any substantial amount of data, the use of the matrix $\mathbf{D}$ is impractical. With just 20,000 documents and 10,000 terms the cartesian product yields a matrix of 200,000,000 values. On the other hand, only a small portion of the terms occur in each document making the matrix very sparse. Figure 3.2 represents a data structure called *inverted index* that provides a computational short-cut (see, e.g., [151]). An inverted index is basically the transpose matrix $\mathbf{D}^\top$ of the document collection matrix $\mathbf{D}$. Each row of $\mathbf{D}^\top$ corresponds to a *posting list* of a term, i.e., a list of documents in which the term occurs and the corresponding term frequency. When computing the similarities between the query $\mathbf{q}$ and the document $\mathbf{d_i}$, it suffices to go through the posting lists of query terms instead of the whole of corpus.

The scores sim_{dot} may vary a lot. A comparison of vectors with many non-zero elements results in higher scores than comparing those with only a few. To eliminate this, the score is normalized by dividing the inner product with the vector lengths, i.e.,

$$sim_{cos}(\mathbf{d_i}, \mathbf{q}) = \frac{\langle \mathbf{d_i}, \mathbf{q} \rangle}{||\mathbf{d_i}||\,||\mathbf{q}||} = \frac{\sum_{k=1}^{n} d_{ik} q_k}{\sqrt{\sum_{k=1}^{n} (d_{ik})^2} \sqrt{\sum_{k=1}^{n} (q_k)^2}}. \tag{3.6}$$

Now, the score is equivalent to the cosine of the vectors [122]. This cosine coefficient assumes the similarity between two documents or between a query and a document is inversely proportional to the Euclidean distance of their vector representations. Thus, greater similarity yields higher scores. However, cosine takes values between 0 and 1, and thus value 0 denotes absolute dissimilarity and value 1 identity.

Algorithm 1 shows the evaluation of the cosine coefficient in an inverted index [151]. The algorithm gets query $\mathbf{q}$ as an input parameter. The algorithm processes a query term t at a time. First, the inverted document frequency is retrieved for the term t. Then, the posting list P is obtained for the term. The list consists of pairs of document-id d and term frequency f_t. For each encountered document-id d, a score is accumulated in structure A (which is called *accumulator*). The accumulator can be a simple hash map consisting pairs of document-ids and corresponding scores. Once all

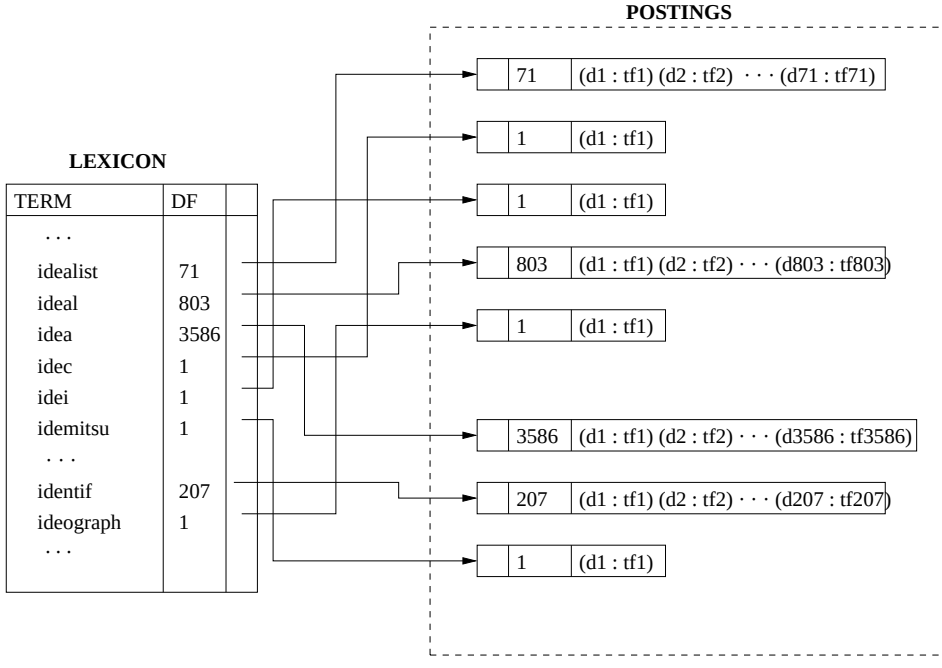

Figure 3.2: An inverted index consisting of a lexicon and a collection of posting lists. The lexicon contains the document frequency (DF) for each term and a pointer to the beginning of the posting list that contrary to the lexicon typically resides in persistent memory. The posting list simply comprises pairs of document ids and term frequencies.

the postings of all the terms in query **q** have been processed, the score for document-id d in the accumulator A equals to the inner product of query vector **q** and document vector of d. These products can be transformed to cosines by dividing them by vector length. In text retrieval, there are typically only a few query terms and so query term weights are all taken to be 1. Should q be a document, its weights must be included on lines 10 and 14.

Finally, the pairs in A are sorted in descending order such that the best matching document - indicated by high cosine coefficient - come first, and the least matching come last. Given a large index, the accumulator often contains a great number of low-relevance candidates due to sporadic co-occurrence of terms. If there is little to gain from sorting this sort of long tail, the scoring can focus on the top k results.

3.1.3 Topic tracking

In Section 2.2.2, we identified topic tracking with information filtering. The system is expected to "associate incoming stories with topics that are already known" [5]. This knowledge is provided in one or more sample stories discussing each tracked topic. The topics are tracked independently of each other, i.e., the arrival of a new story leads to a YES/NO decision for each topic. Ideally, a news story discusses only one topic, so there is

Algorithm 1 Evaluation of the cosine coefficient in an inverted index.

```
1:  A ⇐ {}.
2:  for all t ∈ q do
3:     idf_t ⇐ inverted document frequency for t
4:     P ⇐ posting list of t.
5:     for all (d, f_t) ∈ P do
6:        if A_d ∉ A then
7:           A_d ⇐ 0
8:           A ⇐ A ∪ A_d
9:        end if
10:       A_d ⇐ A_d + (tf_d(t) * idf_t)
11:    end for
12: end for
13: for all A_d ∈ A do
14:    A_d ⇐ A_d/|d|
15: end for
16: A ⇐ partial_sort(A, k)
```

only one correct YES answer.

In TDT pilot project, the team at University of Massachusetts (UMass) adopted the vector-space model [11] and have since continued to employ its various modifications for topic tracking [17, 19, 20, 21, 39, 40, 110, 111]. The approach has been used by others as well (see, e.g., [128, 49, 129]). UMass has used a variant of TFIDF term weights. The TF-component is normalized in the range $[0, 1]$ such that high term frequency in a short document yields higher value than in a long document,

$$tf_{comp}(t) = \frac{tf_d(t)}{tf_d(t) + 0.5 + \frac{len_d}{avglen}}, \tag{3.7}$$

where t is a term, $tf_d(t)$ is the term frequency of the term t in document d, len_d is the length of the document d, and $avglen$ is the average length of document in the collection. It is sometimes called the "Okapi TF", since it is an interpretation of Okapi weighting first used by Robertson et al. [119]. Likewise, the IDF-component ranges between 0 and 1,

$$idf_{comp}(t) = \frac{\log(m/d_t)}{\log(m+1)}, \tag{3.8}$$

where m is the number of documents in the corpus and d_t is the number of documents in which term t occurs. Following the nomenclature of Allan et al. [14], we shall define InQuery weighting scheme as a combination of Equations 3.7 and 3.8 as follows:

$$inquery_d(t) = tf_{comp}(t) \cdot idf_{comp}(t). \tag{3.9}$$

The main idea has remained the same: the sample stories are compiled into centroids, and then each incoming document is matched against each of them. In practice, if the topic centroids are stored in an inverted index, different topics can be tracked in parallel. As a new story arrives, it is executed as a query against the index. From all the possible topics, only those with matching terms are retrieved and scored. If the resulting similarity coefficient score exceeds a pre-defined threshold, the story is associated with the topic. The threshold can be obtained from series of test-runs with training data.

Another participant in TDT pilot project, the team at Carnegie Mellon University (CMU) started tracking with a k nearest neighbor (kNN) classifier [11, 157]. It is called instance-based (or lazy) classifier, as it does not generalize the training data into models but bases the classification judgments on the sample data directly. The number of instances is very

small for a classifier, and to address the sparsity, the kNN system of CMU generates a kNN classifier for each topic. When a new document $\mathbf{q}$ arrives, the k nearest sample stories (by cosine measure) are used in a similarity coefficient that takes the average of cosine coefficient scores between the document and positive samples and subtracts the average of cosine coefficient scores between the document and negative samples [33]:

$$score_{knn}(YES|\mathbf{q}) = \frac{1}{|P_k|} \sum_{\mathbf{d} \in P_k} sim_{cos}(\mathbf{d}, \mathbf{q}) - \frac{1}{|Q_k|} \sum_{\mathbf{d} \in Q_k} sim_{cos}(\mathbf{d}, \mathbf{q}) \quad (3.10)$$

Here, P_k stands for positive samples among all the k nearest to $\mathbf{q}$ and Q_k for negatives, respectively. If the proximity of the new document vector is dominated by negative samples, the score will be low. The team at CMU has developed variants of weighting positive and negative samples [156]. It has also employed Rocchio classifier [120] that is similar to kNN, only that it compiles centroid vectors from positive and negative samples [157, 160].

The cosine coefficient and most of its variants range neatly between 0 and 1, but there are notable differences between the score distributions between topics. The tracked topics vary in size, in scope and in vocabulary. As a result, the ranges of decision scores vary over different topics, and the decision threshold θ may suit some topics better than others. With the centroid-based approach the decision score can divided by the average similarity of the training samples and the centroid, which will unify the scores to work better with a universal θ. As each topic is defined by 1, 2, or 4 sample documents, the normalizing factor z_T for topic T is

$$z_T = \frac{1}{N_t} \sum_{\mathbf{d} \in R_T} sim(\mathbf{c_T}, \mathbf{d}), \quad (3.11)$$

where N_t is the number of training samples, R_T is the set of training samples for topic T and c_T is the topic centroid. With only one sample document, the normalizing factor equals to 1 [14, 18].

Reports of events in the world lead to changes in the vocabulary, and it is therefore beneficial to keep track of this change. On line 4 of Algorithm 1 the computation of the coefficient exploits global document frequencies of terms. Topic tracking runs online and knows nothing about the future, so the only way to adapt the term frequencies is incremental updating, i.e., *incremental document frequency* [18, 49, 128]. This means the document frequencies for terms are updated as they are encountered in the new documents. The updates can take place instantly, in batches of 100 or 1000 documents, or daily, for example.

In early experiments, it was noted that vectors of 20-50 best terms are preferable [11, 19], but it may have been a feature of the InQuery retrieval system [32] as it assumes the documents are to some extent of the same length. Its similarity coefficient, the weighted sum, was later outperformed in topic tracking by cosine that worked well with vectors up to 1000 terms and more [17, 20, 128].

3.1.4 First-story detection

First-story detection reduces to testing whether a new story is sufficiently different from previous data or not. Once something is recognized as a first-story, it is translated into a new topic and stored among the previous ones, i.e., it becomes 'old'. This kind of single-pass clustering approach is described by Algorithm 2. The decisions are done online, although some benchmark runs have allowed deferral of the judgment for a period of 10-20 documents, for instance.

Initially, the set of topic centroids C is empty. There is nothing to compare the first document $\mathbf{d}_1$ to, and so it is added to C. Then, the algorithm proceeds through the document collection or data stream up to N documents. Line 4 represents the execution of Algorithm 1 such that only the centroid with the highest score is returned (partial sort with $k = 1$). If the score does not exceed the given threshold θ, the document $\mathbf{d}_i$ is added to C as a new topic centroid. Otherwise, document i is assigned to topic k [148].

If topic centroids are updated, the process is called *agglomerative* as new vectors are 'glued' to the cluster [123]. If the topics remain untouched,

Algorithm 2 Single-pass clustering algorithm for first-story detection.

1: $m \Leftarrow 1$
2: $C_m \Leftarrow \{\mathbf{d}_1\}$
3: **for** $i = 2$ to N **do**
4: Find $\mathbf{c}_k : sim(\mathbf{d}_i, \mathbf{c}_k) = \max_{1 \leq j \leq m} sim(\mathbf{d}_i, \mathbf{c}_j)$
5: **if** $sim(\mathbf{d}_i, \mathbf{c}_k) < \theta$ **then**
6: $m \Leftarrow m + 1$
7: $C_m \Leftarrow \{\mathbf{d}_i\}$
8: **else**
9: $C_k \Leftarrow C_k \cup \{\mathbf{d}_i\}$
10: where necessary, update representatives
11: **end if**
12: **end for**

the clusters are never explicitly formed which makes it a case of nearest neighbor process. Both have been used in FSD.

In the pilot project, UMass and CMU treated the incoming documents agglomeratively as described above [11, 110, 111]. CMU experimented with group-average clustering (GAC) that 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 quadratic with respect to the number of documents. The efficiency has been increased by dividing the clusters into evenly-sized buckets similarly to scatter/gather-method [45] and applying GAC locally to a bucket before removing bucket boundaries. The single-pass approach constructs clusters in one go, and are therefore computationally less expensive. Although the experiments have no response-time requirements, the test corpora have grown in size over the years, and efficiency has become something of a concern.

Similarly to their topic tracking, UMass replaced InQuery later with cosine [17, 20]. After the initial work, more and more of the vector-space model FSD used 1-NN [14, 17, 18]. Brants, Chen and Farahat [29] presented several interesting and effective improvements to the run-of-the-mill vector-space model. It was found that Hellinger similarity coefficient performs better than cosine in first-story detection. Hellinger similarity coefficient is defined as follows:

$$sim_{hel}(\mathbf{d}, \mathbf{q}) = \sum_{i=1}^{n} \sqrt{\frac{d_i q_i}{\sum_{j=1}^{n} d_j \sum_{j=1}^{n} q_j}}, \tag{3.12}$$

where n is the size of the term-space. Brants, Chen, and Farahat reported higher performance with Hellinger than with cosine in the high recall area.

3.1.5 Observed problems

Inherently, news is about something new. When same event is reported, there is always new information in the news stories. Over time, this accumulates into a conceptual drift making original models or queries invalid at some point. As a result the recall drops. The first TDT corpus covered the Oklahoma city bombing event. Six days after the explosion, Timothy McVeigh was arrested. His arrest was reported in the 61st document relevant to the topic [19]. A natural solution is adaptiveness: if the similarity score for some story exceeds another, higher threshold, it is added to the topic description. Another proposed solution is to adopt a global scoring policy, i.e., tracking scores are dealt as a distribution, not simply as a 1NN

process, and dynamical term-weights, especially for proper names [56]. We will address this technique in Chapter 7.

Overall, the effectiveness of topic tracking algorithms has been somewhat satisfactory – at least compared to that of first-story detection. The task is analogous to information filtering that has been actively researched for years. There was one crucial difference: the traditional filtering tasks relied on high precision while topic tracking requires higher recall. The former has more relevant documents to deal with while the latter has only few. To this end, TDT system adopted Detection Error Trade-off curves instead of recall/precision graphs to evaluate the system performance [15]. The low number of positive samples has been an impediment for the use of machine learning techniques.

One of the problems in FSD was making a distinction between two events of the same type. The discussion about, say, two airplane crashes, tend to have a great number of common words. The difference is, ideally, in the names of airlines and officials, locations and time [11, 21]. This problem owes to the shift from content-based to event-based processing. The large volumes of data and the online nature of the setting do not accommodate for exhaustive similarity comparisons that could remedy the problem at least partly.

Summer school at Johns Hopkins University in 1999 set out to investigate the poor FSD effectiveness [13]. It dismantled the basic approach and rebuilt it piece-by-piece. As a result UMass moved from agglomerative clustering towards nearest neighbor (1NN), because more clusters usually leads to better performance. TFIDF was found better than plain term-frequency, and the use of stop-list was found beneficial. The results were ambiguous on the benefits of stemming, which may have been due to their stemmer (the benefits have been reported elsewhere, see, e.g., [153, 87]). Time decay did not improve the results, nor did named-entity collocations, but focusing on similar content did. Similar content means coinciding non-name terms. Despite small improvements, FSD remained a difficult problem.

If topic tracking were to run without errors, then, because recognizing something 'new' requires recognition of 'old', first-story detection should excel as well. Topic tracking is not perfect, and its failures will affect FSD. Allan, Lavrenko and Jin [16] made a probabilistic estimation of what to expect from first-story detection effectiveness given the state-of-the-art topic tracking. What they suggest as reasonable effectiveness would be 10% miss-rate with 1% false-alarm rate. For the first-story detection to attain that the topic tracking would have to be improved by a factor of 20 (from year 2000 levels), meaning about 1% miss-rate and 0.02% false-

alarm rate. The writers argue that tracking is not likely to make this leap in effectiveness. They encourage approaches that are not based on full-text similarity.

These results marked the end of the first phase of TDT which had focused a lot on testing traditional information retrieval techniques on these new problems. TDT research sought to overcome the inadequacies by enhanced document representations, new retrieval models and use of semantical and contextual information. Despite its faults, the vector-space model has provided a competitive baseline results.

3.2 Language model approaches

An alternative to the vector-space model called *language model* has gained currency in recent years in information retrieval. A language model is a statistical model for generating text. Instead of matching the representation of the information need (query) to the set of document representations (document vectors), it turns the perspective around and estimates the probability of the query having been generated from the document model [115]. Thus, a document is a good match to a query, if it is likely to generate the query.

Unlike vector-space model, which is basically a heuristic, language model provides a principled foundation to retrieval process.

3.2.1 Language models

A language model is a probability distribution over terms. In information retrieval language modeling means that for each document d there is a probabilistic language model M_d. Given a query q, the retrieval process ranks documents by $P(q|M_d)$, i.e., the probability of the document model M_d generating the query q. Although we do not know the actual model, we consider document d as a sample of text drawn from the model distribution, and so we shall use document d to estimate the language model M_d. Then, this model M_d is used to estimate probabilities of term sequences, e.g., that of query q [99].

In *query likelihood model* [115], the objective is to estimate the probability $P(d|q)$. Using the Bayes rule it can be written as:

$$P(d|q) = \frac{P(q|d)P(d)}{P(q)}. \tag{3.13}$$

Since our ultimate goal is ranking, $P(q)$ is redundant as it is same for all documents d. The prior probability of a document $P(d)$ could arise from

some quality of the document (e.g., number of incoming links, number of downloads, length, or author), but usually it is treated as a uniform probability distribution and so it is same for all documents d and can be eliminated. By the virtue of these simplifications, the ranking relies on $P(q|d)$. Again, document d is modeled with language model M_d, and so the ranking boils down to estimating $P(q|M_d)$.

There are many kinds of language models, but information retrieval applications typically apply *unigram models* which are bag-of-words models in that they neglect word ordering, and any ordering of the terms would yield the same probability. Unigram models are relatively easy to estimate, and yet they have been quite effective in information retrieval. This owes to the *multinomial* view of the generation process: terms occur independent of each other [86]. By the virtue of the independence assumption, we can employ, for instance, maximum likelihood estimation (MLE) that multiplies the individual term probabilities as follows:

$$\hat{P}(q|M_d) = \prod_{t \in q} \hat{P}_{mle}(t|M_d) = \prod_{t \in q} \frac{f_d(t)}{L_d}, \tag{3.14}$$

where $f_d(t)$ is the term frequency of term t in document d and L_d is the number of tokens in document d.

As a form of data, text is very sparse, and so the models are plagued by so called zero-frequency problem. In Equation 3.14, if there is at least one query term t that does not occur in the document d, the estimate yields $\hat{P}(q|M_d) = 0$. The remedy is *smoothing* which means on one hand discounting the encountered occurrences of terms, and on the other hand distributing that discounted probability mass to unseen terms. Smoothing has become a central piece in language model based retrieval, because it not only deals with zero probabilities but also implements considerable portion of term weighting [99]. A common technique is *linear interpolation smoothing*. It combines the probability from the document with a probability from a general background model M_c,

$$\hat{P}(t|d) = \lambda \hat{P}_{mle}(t|M_d) + (1 - \lambda)\hat{P}_{mle}(t|M_c). \tag{3.15}$$

Now the probability of term t occurring in document d is partly based on a larger collection of documents. The parameter $\lambda \in (0, 1)$ can be obtained from training with a test collection, for instance.

Language models have not been adopted without objections. Philosophically, the models identify information need representation with documents themselves. They lack explicit notion of relevance, and offer the pursue of the query generation process instead. This hampers efforts to

integrate relevance feedback and query expansion techniques, for example. The probability of term t occurring in a relevant document has been difficult to estimate, as there is no training data for ad hoc queries. Lavrenko and Croft present a technique that assumes an underlying *relevance model* from which both the query and the relevant documents are generated by random sampling [81]. Here, the query is not a random sample of a document but of the relevance model. In absence of training data, the authors argue that the query terms provide a good approximation of which terms would occur in relevant documents. So, given a query q, a set of documents is retrieved. From that set, high-ranking matches R_q containing most or all query terms are used to approximate $P(t|R)$, i.e, the probability that term t occurs in a document relevant to a query, where R represents the class of documents relevant to the query. This technique is a way to generate training data on the fly, and integrate a notion of relevance into language models by estimating $p(t|R)$ through $p(t|R_q)$,

$$P(t|R_q) = \sum_{d \in R_q} P(t|d)P(d|q), \tag{3.16}$$

where the probability $P(t|d)$ lends itself to the maximum likelihood estimate with linear interpolation smoothing of Equation 3.15. Relevance model approach has been supported with strong effectiveness in text retrieval as well as in topic detection and tracking.

3.2.2 Topic tracking

Early on, Dragon Systems employed a combined unigram language model and a beta-binomial language model in a topic tracking system [90, 147, 155]. The beta-binomial models represent a topic with a mixture of unigram distributions. Thus, to estimate the probability of encountering n_t occurrences of term t in a document d, the model employs a combination of binomial distribution and beta distribution. Later, the beta-binomial models were dropped, since unigram models appeared to fare better in the tasks demanding a high recall [153]. A discriminative model was built from a larger background collection. All models were smoothened with mixture models from a background text corpus. The tracking decision rested on log-ratio of the probability that the given story was generated from the topic model and the probability that the given story was generated from the discriminator model.

BBN Technologies combined three variations of language models in topic tracking [69, 83]. The respective influence of each model was resolved with logistic regression on labeled training data. BBN normalized

the scores over all topics by z-scoring them on a distribution estimate from the training data. The combined system outperformed the individual systems.

The team at TNO [1] used a single model based on maximum likelihood estimation and linear interpolation smoothing as described in Section 3.2.1. They also normalized scores somewhat similarly to BBN above yielding similar results [137]. In a similar vein, the group at LIMSI [2] adopted MLE approach: the decision score was log-likelihood of ratio between a topic model and a background model. The group also experimented with document expansion and unsupervised adaptation and found them both advantageous [87, 88, 89]. Also, The Royal Melbourne Institute of Technology (RMIT) employed a system nearly identical to that of LIMSI although did not achieve quite as high effectiveness [84].

He et al. at University of Maryland applied the Lemur Toolkit [3] for their language model that followed the ideas described above [67]. However, they wanted to achieve an effect similar to inverted document frequency and thus divided the interpolation of Equation 3.15 with $P(t|M_C)$ thus straying from language models as the result is no longer a probability. The results were not bad but not as good as other systems. The problem may have been in "sub-optimal" λ in the interpolation phase and a small sample size in the z-score normalization distribution construction. Later, the team adopted n-grams as the basis of their topic modeling [51]. Each topic contained two models, one for on-topic and one for off-topic stories, both of which were smoothened with background term distribution. Decision scores were log-ratios between the on-topic and off-topic probabilities that were modified by term-frequency factor. The system fared rather poorly, possibly due to pronounced sparseness of n-grams with 1 to 4 training documents.

The group at UMass utilized relevance models on topic tracking after having applied them on story-link detection [80, 18, 106]. The similarity between models was measured as the asymmetric clarity adjusted divergence, which is a Kullback-Leibler divergence modified by query clarity, i.e., "value to query based on how different its associated language model is from the corpus language model" [44]. The divergence is thus similar to length-normalized log-likelihood ratio. The purpose of the modification is to achieve an inverted document frequency effect for term weights. The

[1] Nederlandse Organisatie voor Toegepast Natuurwetenschappelijk Onderzoek (Netherlands Organisation for Applied Scientific Research)

[2] Laboratoire d'Informatique pour la Mécanique et les Sciences de l'Ingénieur, Centre national de la recherche scientifique (The Computer Sciences Laboratory for Mechanics and Engineering Sciences, French National Center for Scientific Research)

[3] http://www.lemurproject.org/

approach has been used in topic tracking in the form [40, 79]:

$$sim_{rel}(T, d) = \sum_{t \in d} P(t|M_T) \log \frac{P(t|M_d)}{P(t|M_c)}, \tag{3.17}$$

where T is the topic, d is the document, and M_T, M_d, and M_c are the models of topic, document and background collection, respectively. The results have been strong.

Although the score normalization and smoothing have turned out to be crucial parts of language modeling, and systems' performance rests on how carefully they are carried out, no major obstacles are reported in topic tracking. In 2004 benchmarking, a relevance model performed slightly better than the best vector-space model.

3.2.3 First-story detection

The gains language models have made in topic tracking have not yet translated into better performance in first-story detection. In fact, there have been only a few reported attempts. Dragon Systems employed their tracking approach to topic detection [90, 153, 155]. As a problem it is not quite the same as first-story detection. Likewise, BBN focused on topic detection and not FSD [83]. The reports, however, suggest that linear interpolation was a better smoothing approach than the back-off method, where only the zero probabilities are adjusted and non-zero are merely rescaled by a constant so that everything sums up to 1.

Summer School at Johns Hopkins experimented with language model based FSD and explored a variety of techniques to model the probability of first-stories. The participants built a system the performance of which was equivalent to a vector-space approach [13]. Ultimately, the techniques that were found to work well, were fairly analogous to those in vector-space model. Although the formulae were different, the ideas they portrayed were the same, e.g., term-frequency weighting and length normalization.

Zhang, Ghahramani, and Yang modeled FSD as a Dirichlet process with a growing number of topics. The approach produced a marginal improvement over their baseline, a version of CMU's group-average clustering, but the results are behind the vector-space models in general [161].

3.2.4 Observed problems

There seem to be two fundamental problems in applying language models to topic tracking. First, like in vector-space models, there is a need for normalization of decision scores over topics that is not necessarily motivated by

statistical reasoning. Second, in order to avoid the zero probability problem arising from sparse data, smoothing techniques are required. Reading through the past research, the language model methods display a great variance in the result, i.e., two systems using similar methods end up in the opposite ends of the performance scale. In the absence of a thorough failure analysis, we assume bad results are due to a failure in applying either one (or both) of the above techniques.

3.3 Use of semantical and contextual information

A prominent avenue that TDT research has taken since the results by Allen, Lavrenko and Jin [16] leads to natural language processing and exploitation of its products. Natural language processing (NLP) focuses on the interaction of computers and human (natural) language [100].

3.3.1 Named entities

A *named entity* (NE) is a word or a sequence of words that form a proper name like 'Bismarck', 'sir Walter Scott', 'U.N.', or 'Gulf of Mexico'. Names of people, organizations and places anchor the news into the real world making the stories meaningful. It is difficult to quantify their importance to readers, but news reporting without proper names would certainly decrease their informativeness.

The special role of certain kinds of terms was also noted early in TDT research. Papka associates the news medias' four W's – who? what? when? where? – with names of people, names of organizations, temporal expressions [4] and locations, respectively, and assumed one could construct a more effective representation by the virtue of harnessing these. An increase in the term-weights of named entities boosted the effectiveness, and a decrease lowered it somewhat. The gains were not consistent throughout the space of decision scores. One problem with large scale adoption of named entity recognition is the fact that much of the corpora is a product of automatically recognized speech, and as such lacks proper punctuation and capital letters [110].

Allan et al. [13] found that basing the decision score of first-story detection solely on the number of new named entities increased the effectiveness, if only marginally. At the time the first-story is published, the people involved might not be known. Another problem is that media tends to focus

[4] Frege argued that temporal expressions are proper names in that they denote entities much in the same way locations do [57].

on known people, politicians and celebrities who are in the 'spotlight' very often for a variety of reasons.

Lam et al. [78] built a document representation comprising three vectors instead of a typical one. One vector contained concept terms, i.e., important terms drawn from similar documents, another named entities and a third common terms. The overall similarity of two documents was a linear combination of the three kinds of similarities. The system was used for topic detection, and the performance was not competitively good.

Yang et al. [160] made an effort to tackle the problem with recognizing two topics of the same kind. Typically, the TDT systems find it difficult to distinguish between two different airplane crashes. They proposed a two-layer model, where documents were first classified into topic-type categories like 'elections', 'tornadoes', and 'airplane accidents', and then carried out the first-story detection with seven different types of named entities. The idea was that by belonging to the same topic-type, the documents already shared a lot of common words. Their difference would reside in people, places, times, and dates. In addition, the terms were re-weighted inside that topic. The classes of different NE's were weighted by their effectiveness, which here means an average χ^2 statistic with respect to topic and features. The actual detection employed Rocchio, so the method was based on the vector-space model. The experiments showed considerable gains over their baseline. It was also found that locations are the most informative NE's.

Carthy and Stokes [139, 138] augmented the traditional document representation with *lexical chains*, sequences of related words of the document. These chains were built from a lexical database, WordNet [104], and as a result the document representation comprised a term vector. The results in first-story detection were not as good as the cosine TFIDF baseline. Although the use of lexical databases has not been widely adopted, similar multi-vector approaches have appeared in numerous works (see, e.g., [36, 50, 56, 66]). The idea is that by comparing the combined representations of documents, the similarity or the dissimilarity of different aspects of the stories provide a more accurate estimate of the event-based similarity. Thus, stories about two plane crashes could be recognized as two separate instances of the same type of event.

Nallapati extracted three kinds of named entities (people, organizations, locations) and four parts-of-speech (nouns, verbs, adjectives, adverbs) into designated vectors in a story-link detection system [105]. The different classes of terms were compared with language models, and the resulting set of values was classified with a linear perceptron. Verbs and adverbs were attributed small weights indicating their insignificance in the document

similarity. Overall, the results were nearly identical to a unigram baseline.

Yet a similar route was taken by Kumaran and Allan [76]. Their first-story detection system first classified documents to topic-type categories, and then used three document vectors consisting of all terms, named entities and non-names terms to determine whether a story is new or not. Some topic-type categories are more sensitive to named entities, while others are to non-names. By exploiting topic-type based modeling the authors achieved a gain in performance. Zhang, Li, and Gang did a similar finding in measuring the average correlation between a named entity type and topic type [163]. For instance, reports on elections and legal cases were focused on people, sports and scandals on organizations, and disasters, war and finances on locations. When the topic-type was taken into account when updating the term-weights, the first-story detection effectiveness enhanced considerably.

Kumaran and Allan [77] also experimented using a support vector machine for finding proper weights for the term vectors without topic-categories. The first-story detection system triumphed over the baseline cosine TFIDF approach. However, they note that not all topics carry named entities in all relevant documents. A report on a disaster might only briefly mention the larger event and instead focus on suffering and hardships of an individual.

Connell et al. [40] used different types of named entities to adjust the common cosine TFIDF similarity coefficient in first-story detection. When vectors of a named entity type, say, locations, showed a low similarity, the cosine coefficient was decreased by a fraction. A sufficiently low similarity in all named entity types meant the cosine coefficient dropped to zero. The experiments favored simple cosine TFIDF approach over NE-application.

3.3.2 Locations

Location information has been used widely in information retrieval. It is the bread and butter of Geographical Information Retrieval (GIR) systems that are designed to answer queries on geographically referenced unstructured data. In resolving queries with focus on the locality of the user, the system makes use of geographic coordinates or ontologies [103]. In TDT, locations have been mostly used simply as named entities or terms. Thus, the comparison of location terms have not made much use of geography.

Jin, Myaeng, and Jung [70] note that in their collection of news material half of location terms occurred in the first 20 percent of the document. The location terms later in the document were found passing references to related events or themes. They used a path length in a geographical on-

tology for the similarity coefficient of two location terms. Similarly to [76], they increased or decreased the full-text similarity depending on the similarity in location terms. The tests suggest that use of location information was useful. The experiments were not carried out on a TDT corpus, so cross-system comparison is difficult.

3.3.3 Temporal information

The recognition and formalization of temporal expressions is not a novel idea. Mani and Wilson have presented an approach for resolving temporal expressions from news-feed with manually produced and machine-learned rules [97]. They could resolve expressions with explicit ('tomorrow') and positional ('next week') offsets based on the reference time and implicit ('Thursday', 'February') offsets based on reference time and the verb tense. While Mani and Wilson confined their experiments to core expressions and neglected prepositions, Schilder and Habel introduced prepositions in their finite-state automata [126] ('after last week', 'until Tuesday'). The formalized expressions have been applied mostly to ordering problems, i.e., whether something happened before, during, or after something else.

In information retrieval the use of temporal expressions has not been as widespread. Although Alonso et al. [22, 23], for instance, have used temporal expressions in clustering and ordering web search results, temporal expressions themselves are seldom used in text retrieval queries [108]. There has been, however, some work using time information in detecting events from the text. Smith employed spatio-temporal attributes in detecting historical events from a digital library [133]. His method employs place-time collocations. Koen and Bender augmented news documents with external temporal information and thus anchored the temporal expression occurring in the text into events more familiar to the reader [75].

TDT-related work has sought ways of combining real world time and terms. Swan and Allan have developed methods to track the co-occurrence of terms over time and build timelines of topics based on the unusual phrases and words [140]. Temporal summarization is an attempt to build chronologies of events in a topic [12]. The events are expressed by action sentences, and the problem is to find the most relevant sentences such that the progress of the topic is condensed. These methods do not use temporal expressions in the text but publication date of the documents.

In topic detection and tracking, the use of temporal information started with the recognition that the documents that discuss the same topic tend to be temporally proximate. This means the topics appear in 'bursts', and their publication dates are close to one another. Often, the vocabulary

reflects this: some terms and term combinations are highly topic-related appearing in bursts as well. This has been exploited by introducing a time penalty factor by which the weight of the terms decayed over time [21, 111, 29, 158, 157, 83]. Also, a term would be considered *surprising*, if it had not occurred recently [21]. In addition, source specific and event-based weighting models have been reported to increase the effectiveness [29].

Pons-Porrata, Berlanga-Llavori and Ruiz-Shulclope [113, 114] extracted temporal expressions from the document content and used them in building event hierarchies. Their document representation contained a temporal vector in addition to normal term vector. The temporal information was used to take advantage of the burstiness: if the closest dates found in the documents were too far apart, the document similarity was set to 0, otherwise to the cosine of the term-vectors. Much in the same way, Kim and Myaeng [74] used temporal expressions to frame a comparison window for documents. Li, Li and Lu [85] begin with the assumption that topics have a time of occurrence, because they are event-based. If incoming documents contain references to that time, their system increases the term similarity between the document and the topic.

3.3.4 Sentence-level approach

The persistent problems especially in first-story detection have raised questions about the suitability of the bag-of-words models in TDT. In previous sections the named entities were put into their own bags, which paid off in some gain in effectiveness. It is difficult to recognize important terms, because the same term can be detrimental to the topic model, depending on the context in which it appears.

Fukumoto and Suzuki [58] make the very important distinction between topics and events: topic is a collection of events. They classify terms into three categories: general terms that occur through-out the corpus, event terms that are characteristic to the news story in which they occur and occur through-out the news story, and finally topic terms that are characteristic to news stories relevant to the topic and are sensitive to the context (paragraph) in which they occur. If the topic was the 1995 earthquake in Japan, topic terms would include 'Kobe', 'Japan', 'earthquake', and 'fire', while event terms would consist of 'rescue', 'emergency', 'area', or 'worker'. Fukumoto and Suzuki present a method picking out these topic and event terms and a way to use them in topic tracking: All topic terms of the training stories are compiled into a single vector, and all individual training stories are left with only event terms. If a new story is more similar to the topic term vector than any of the training stories, it is considered on-topic.

The effectiveness thus reported is very good, although the data is the first and smallest of the TDT corpora.

Allan et al. experimented shifting the focus from word-level to sentence-level language models [18]. Sentences express ideas, and syntax binds the constituents, words, together. So, instead of estimating the probability a word was randomly drawn from a document or a topic model, they estimated the same probability for a sentence, an ordered sequence of words. Bundled with a unigram detector, it provided a small gain in story-link detection over a unigram baseline.

3.4 Our approach

Our main contributions are two-fold. Firstly, we present a framework for document similarity that can make use of ontologies. Secondly, we investigate how the term occurrences can be anchored on to a timeline through the temporal context. We run experiments with both static and adaptive systems.

In addition, we put forward a conceptual analysis that attempts to clarify the vague and inter-changeably used notions of event and topic. The analysis serves partly as a starting point for our time-related work as well as adaptive tracking.

3.4.1 Ontology-based document similarity

The TDT system performance has been plagued by high false alarm rate. The terms in two documents seem to be similar largely by accident. They discuss different events, but they share a great deal of terms. On the other hand, making the distinction between two different airplane crashes or train accidents has been difficult. The terms of two documents discussing the same *kind* of event tend to converge and therefore a term vector is not able to represent the delicate distinction between documents regarding similar but not same event [11]. However, Allan, Lavrenko and Papka suspect that only a small number of terms is adequate to make the distinction between different news events [19]. Intuitively, it would be temporal references, locations and names that would vary more than other terms.

A news document regarding an event reports at the barest what happened, where it happened, when it happened, and who was involved. Most topic detection and tracking approaches have tried to encapsulate these facts in a single vector. In order to attain the delicate distinctions mentioned above, to avoid the problems with the term-space maintenance and still maintain robustness, we divide the term-space into *semantic classes*,

i.e., groups of words that have meaning of the same type. The semantic class of PLACENAMES 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 name a logical object, that is, a point on a global time-line, and bind the document onto the time-axis. PERSONS and ORGANIZATIONS are proper names and tell who or what was involved. What happened is represented by normal words which we call TERMS.

Figure 3.3 illustrates an example of document representation with semantic classes. The representation comprises a semantic vector for each semantic class. These vectors reside in distinct vector-spaces so that they each can be assigned a dedicated similarity measure that can be based on ontologies. In computer science, an *ontology* is a formal representation of concepts and their relationships. A geographical taxonomy is an ontology defining the set of possible PLACENAMES and defines relationships between them. A time-axis is likewise an ontology, albeit only one dimensional: the timeline is an ordered (and usually for practical reasons discrete) set of points. The semantic classes need not limit to these; they can be any domain of terms.

In Figure 3.3 PLACENAMES are connected to a geographical ontology. The terms denoting a place are thus given an interpretation and they can be compared to other PLACENAMES through the ontology. Similarly, PERSONS can be assigned a database of people and thus we can overcome some of the problems arising from same person having different names, e.g., 'Steve Fossett' and 'James Steven Fossett'.

The comparison of two documents is carried out class-wise: the names in the one document are compared to the names in the other, the locations in one against the locations in the other, and so on. As a result, we have a vector of similarity values that we turn into a single yes/no decision. Such resolution can be attained with virtually any machine learning technique. We are employing support vector machines due to the robustness.

The ontology-based document similarity will be detailed in Chapter 5.

3.4.2 Temporal indexing

News documents contain a wealth of information coded in the natural language temporal expressions. Automatic processing of news often neglects these expressions for several reasons: temporal expressions are difficult to spot, their surface forms themselves are hardly of any direct use, the meaning of an expression is often somewhat ambiguous or at least very difficult to resolve, and the expressions do not lend themselves easily to any kind of comparison. However, this temporal information would be highly use-

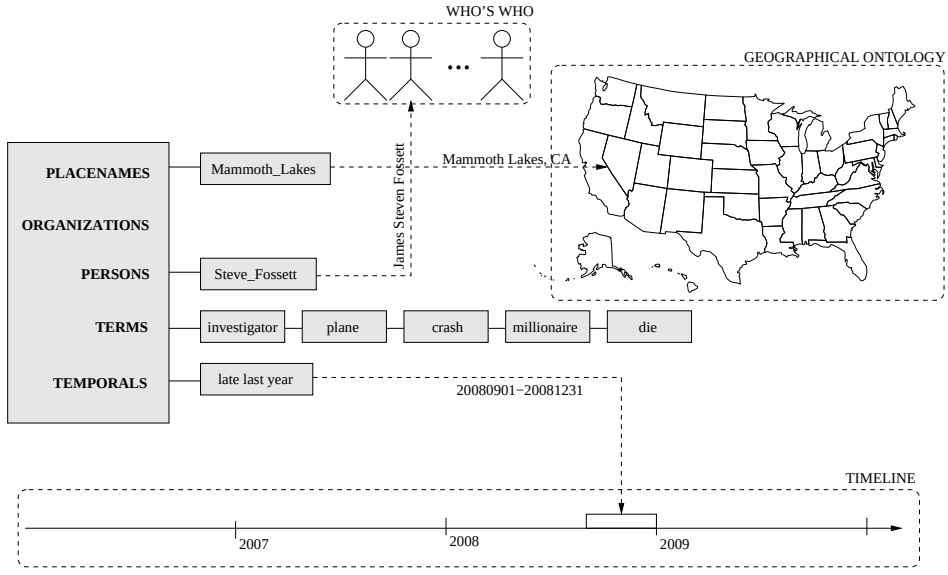

Figure 3.3: An example of a multi-vector consisting of five semantic vectors colored in gray. (*"Federal investigators have revealed new findings in their probe of the plane crash that killed millionaire adventurer Steve Fossett. They say it appears that Fossett died in the initial crash. Wreckage of Fossett's small plane was discovered near Mammoth Lakes late last year."* MSNBC, March 6, 2009)

ful for many areas of applications: (context aware) information retrieval, information extraction, question answering, document summarization, and topic detection and tracking, for example [130].

There are three problems one has to deal with before temporal information can be applied automatically in any of these tasks: *recognition*, *formalization* and *comparison* of the temporal expressions. First, the expressions have to be extracted from the text. Second, the expressions need to be provided with formal meaning. In addition to the expression, one often needs some context information, such as the utterance time and the tense of the relevant verb, in order to map the expression. And finally, there has to be a suitable method of employing the formalized expression.

In recognizing temporal expressions, we employ finite-state automata based on functional dependencies of the words. Once an expression is recognized, the terms it contains are converted to *shift* and *span* operations that move the utterance time to the past or to the future. We define these operations on top of a *calendar* [63] that defines the global time-line and

its various granularities of time. In addition, we provide semantics for expressions with considerable degree of indexical variety, e.g., *by early May* or *later this month*.

Unlike in some of previous approaches, we are not attempting to establish a chronological order for various actions occurring in the text of a news document. We shall employ the formalized temporal expressions in two ways. Firstly, we make a semantic vector out of them. The terms in the vector are temporal intervals, and their comparison yields a similarity score. Secondly, the term occurrences in documents are augmented with temporal references. In other words, we assume that all terms occur in the context of some temporal expression, at least the publication date. When comparing (semantic) term vectors of two documents, we can increase or decrease each pair-wise term comparison depending on the cohesion of their temporal contexts in the documents.

Temporal expressions and their integration to document similarity is presented in Chapter 6.

Chapter 4

Basic experimental setup

Our baseline system is based on the vector-space model. It is simple, efficient, and over the years has been a competitive system in topic tracking [128, 14, 18, 40] and especially in first-story detection [30, 31, 40]. The implementation is based on the work of UMass research group. Our semantic approach presented in Chapter 5 is an extension to the basic document vector representation.

In Section 4.1 we will take a closer look at news, events and topics, the core concepts of TDT, and clarify the picture of TDT. Then, in Section 4.2 we examine the three benchmark text corpora used in experiments. Section 4.3 deals with a baseline topic tracking system and its results. In a similar vein, Section 4.4 explores the baseline first-story detection system and results. Section 4.5 concludes the baseline work.

4.1 News and events

TDT research has rarely ventured into problematizing events and topics. Discussions about the nature of news data in general is virtually absent. Yet, events and news is what TDT is all about. So, we seize this opportunity to straighten out some of the basic concepts and slight incoherence in the terminology. In our view, these basic concepts govern the relevance in topic-based retrieval. For our part, we shall start filling this philosophical void with a conceptual analysis of both news and events in Sections 4.1.1 and 4.1.2, respectively. So, what is it the systems are supposed to find?

4.1.1 News as data

Heraclitus of Ephesos held that world is a constant flux, where things change all the time without ever re-occurring exactly the same [25]. In-

deed, we seem to be surrounded by a constant change from microscopic phenomena involving particles to the affairs of celestial bodies, from slow conceptual shifts to abrupt mental turmoil. Assuming an increasing universal entropy, there is no possibility of complete re-occurrence of any past state of affairs. Naturally, from this virtually infinite continuum of change, only a small portion bears significance to humans. Even smaller portion ever surfaces the public awareness through news media.

Whether an event is a product of a human intellect, i.e., a sort of shorthand for a particular change in the world, or if it is a metaphysical entity of its own, is an open question. Whatever the case, we do not perceive events per se, and meaningful experience is possible by the mind selecting, interpreting and organizing the sensory information into identifiable events. On the other hand, the sensory information is not necessary as we can contemplate on events that have happened in distant past or have not yet occurred or that are completely fictional [1]. However, we can say with some certainty that our linguistic devices, causal and temporal thinking, planning and actions are all well tuned to events and event structures [112]. The problem is, however, that we cannot express or communicate events directly. We can only describe them.

News is about describing events taking place in the world, and therefore the reader of news or the user of a TDT system does not perceive the events themselves, but descriptions of the events. So, *"news is not what happens, but what someone says has happened or will happen"* [132]. The descriptions are not delivered in the form of plain chronologies, but as *narratives* placing news reporting in the long tradition of story-telling [28, 34, 62, 150, 146]. In the industry lingo, a piece of news is a "news story". The difference between a chronology and a narrative is in unity and coherence: as an experience is organized into a narrative form, irrelevant facts are omitted and the event is provided with a context and meaning. The selection and interpretation of facts is possible, because the story-teller has the benefit of hindsight: she knows the course of the event and can focus on what is relevant and what is not with respect to her "angle".

Newsworthiness is not an inherent characteristic of an event. It is a judgment made by an editor under the influence of norms, conventions, medium, organizational pressures, competition, and temporal constraints. News agencies fill their programs and papers with a fairly constant amount of material regardless of what is taking place in the world. In analyzing the selection process of a news agency, Gans [60] pointed to *importance* and

[1] The difference between these three kinds of events remains outside the focus of this work.

interestingness; ideally a good news story yields both. Importance arises from timely issues in the local, national and international scene: activities of decision-makers, politics, abnormal behavior (e.g., strikes, demonstrations, crime), or events affecting a large number of people (e.g., disasters, war, employment). Interestingness has more to do with people: deviation from the norm, man-bites-dog role-reversals, or heroic deeds. Gans's study shows that newsworthiness is dominated by official sources and well-known people, i.e., political, cultural, and economical elite. A more recent survey supports this observation, but marks also entertainment and surprisingness as factors in the news selection [65]. Also, an event initially covered in the news is likely to have follow-ups.

There is a variety of story types from news briefs and reports to editorials, features, comments, and columns imposing a variety of format and style restrictions. The television format is composed mainly of short reports, expanding rarely to discussion or multiple viewpoints [41]. In contrast to compact broadcasted packages, newspapers tend to engage in an analytical approach with lengthier background and wider time-frames and areas [26]. Still, news agencies are mostly living in a constant present, and their reporting is oriented to the past twenty-four hours [127]. This is especially true for newswire.

News are manufactured representations of real world events. The medium, language, has a bias: the choice of words often carry intended or unintended valuations [55]. Although news agencies and reporters strive for objectivity, they do so within their own set of sociological, ideological, and literary constraints. This means news are not a perfect mirror of the world nor are they just plain facts [101].

4.1.2 Events and topics

Topic detection and tracking revolves around the notion of event. Events are understood to be phenomena of the real world, and a TDT system perceives the world through news stories, as discussed in the previous section. In topic detection and tracking, coining a solid definition for an event has proved to be difficult. Reaching a wide consensus has been equally difficult in humanities, history and political science, for instance [52, 61]. We shall attempt to weed out some of the inconsistencies from the TDT terminology.

In TDT pilot project, an event was understood as a unique occurrence of something in contrast to a *topic*, a more general class of events – a particular presidential election vs. elections in general. Initially, the definition read as follows [11]:

Definition 4.1 *An event is some unique thing that happens at some point*

in time.

This definition is intuitively sound as it leaves the scale of the event open. If taken literally, however, it seems to lend itself to Davidson's ideas that approach events from the sentences that describe them and regard an event as if a hidden variable of a sentence [46]. The variable is a place holder for an occurrence of an action expressed by a verb or a noun phrase. The sentence "*John is talking to Mary*" can be expressed in form:

$$\exists x : talk(x, \text{John}, \text{Mary}) \wedge time(x, now).$$

The variable x denotes an event, which in Davidson's ontology is a particular, i.e., a concrete, datable, locatable individual, and thus perfectly in agreement with Definition 4.1 [48]. This variable paves the way for augmenting the event with further details: John and Mary are standing in the corridor or sitting in the park, John is in a hurry, and Mary is not paying any attention. An event can have any number of descriptions stating a variety of (non-contradictory) facts. The sentence "*John is talking to Mary in the corridor*" could be formalized as

$$\exists x : talk(x, \text{John}, \text{Mary}) \wedge time(x, now) \wedge place(x, corridor).$$

This formalization refers to the same event as the one above, although comprising an additional predicate. Davidson's approach has been employed in artificial intelligence. A framework called *event calculus* models the world as a sequence of events using the logical form similar to the one above (see, e.g., [121]).

Definition 4.1 might work well with artificial intelligence systems that deal with formalized propositions. News stories, on the other hand, contain several action sentences and thus describe several of unique events. When a U.S. jet severed a funicular cable in February 1998 near Trento, Italy, the reports comprised a number of events: the jet is flying, the jet hits the cable, the cable is cut, the cable car crashes, the rescue teams are launched, and investigations carried out. When a fireman is interviewed, his act of responding is an event (*"..., he said"*). Logically, they are equally valid events, so how do we determine which one to track? Apparently, to address this problem, Definition 4.1 was later adjusted [38].

Definition 4.2 *An event is a specific thing that happens at a specific time and place along with all necessary preconditions and unavoidable consequences.*

Although events of Definition 4.1 need not be interpreted as logical events, Definition 4.2 clearly arises from logic. The idea is that the plane

flying too low is a necessary precondition and the cable car crashing is unavoidable consequence of the severed funicular cable. The plane severing the cable, the cable being cut and the car crashing are all specific things that happen, and, as such, they could be events to which other things connect via causality. The definition simply shifts the problem to a very difficult terrain of modal logic: what are the necessary preconditions for the jet to cut the cable, and which consequences are unavoidable. Moreover, how far do we extend the causality? Even if we could tread this terrain, we would have to rely on what we are told in determining the preconditions and consequences, and some descriptions make the causal relation more apparent than others [47]. The improved Definition 4.2 is not very helpful.

The conflicts in Kosovo and Macedonia relate to other political conflicts in the area of former Yugoslavia. The inflamed situation between the Israelis and Palestinians can be seen a series of actions and counter-actions, steps towards peace and drawbacks. The SARS epidemic comprised outbreaks of the flu in numerous places, issued quarantines, and quests for remedy. Definition 4.2 does not really recognize these large scale events, because they are geographically or temporally apart or scattered. Another attempt to establish a connection between the Davidsonian singular events put forward the following definition [110]:

Definition 4.3 *An activity is a connected set of actions that have a common focus or purpose.*

The definition is intuitively appealing. Political or military campaigns and disaster relief efforts would qualify as activities. According to Definition 4.1 actions are events: they are singular occurrences. Hence, activities are sequences of events involving human intention and planning. The purpose or focus might not be obvious right from the first story, so we may have no way of knowing a priori, if something happening is just an event or an activity.

On the other hand, natural phenomena like volcanic activity, series of earthquakes or global dimming can be seen as activities in the above sense, although lacking a direct human agent. Naturally, the actions grouped under an activity need to be relatively proximate in time and space; all tectonic shifts do not qualify as a connected set of actions.

Initially, the term *topic* was used to refer to stable categories of events, like 'volcanic eruptions', 'elections' or 'plane crashes'. Then, the notion was used to sow together the distinct events [38].

Definition 4.4 *A topic is an event or activity with all directly related events and activities.*

If an activity is a series of related events, then here it is redundant. In addition, the definition seems to slide from the world of particulars into the world of man-made connections between them. We may observe something happening, but we do not observe topics; we make them [2]. Events of Definition 4.2 are things that happen, but a topic is a human judgment, a way of grouping these events. Hence, a topic cannot be an event, or an event is an arbitrary human judgment, too. This definition does not seem to work well with Definition 4.2.

Yang et al. [157] understood topic as a dynamically changing event: initially the event is about the severed funicular cable, but is later extended to include the technical investigations, the public outrage and trials, for example. However, if an event is something specific that happens somewhere at a specific time, it does not evolve or change – it simply happens. The notion of event evolution is useful but not quite accurate.

Definitions 4.2 and 4.4 seem to take the smallest available unit of change and build their case in a bottom-up fashion. All definitions above are problematic in several ways:

1. The logically motivated definitions do not lend themselves to the narrativeness of news. News is not facts but story-telling. We can carry out only limited logical inferences based on what we are told.

2. The difference between the phenomenon and its description is glossed over. This distinction is fundamental to source criticism in journalism, literary criticism and humanities in general. TDT should follow suit, because it does not perceive the events themselves. Moreover, the source could well be a component in what the user deems relevant.

3. The definitions cannot cope with fictional or erroneous descriptions. They cannot address cases like Jessica Lynch's reported fire-fight near Nasiriyah, Iraq, in 2003 (which was later found fabricated), or the alleged massacre in Timisoara, Romania, in 1989.

4. There are events of varying magnitude with larger events comprising smaller ones. The news stories report and package the events into narrative form. The connections between the events need not be strictly causal or logical. They only need be relevant.

So, how do we address these problems? Pachter defined an event as something denoting a shift from a situation to another, "*the juncture be-*

[2] We identify with the nominalist tradition that claims the universals are mere names. We think that making of news topics is a form of *dynamic nominalism*, although Hacking was more concerned with the emergence of classifications and ontologies thereof [64].

tween two situations S' and S''" [109]. The scale of S' and S'' is irrelevant: a shattering of a wine glass is an event just as the fall of Roman Empire is. The scale merely restricts the level of phenomena that can be called causal and relevant. It is not all that different from Definition 4.1, it just highlights the choice of scale.

We shall make a distinction between events and news events. Events are what happen, news events are what the media says happened.

Definition 4.5 *A news event is a newsworthy real world event or series of events as reported and packaged by the news media.*

News stories are about events, and by story-telling the event is transformed into a news event. A news event is a literary product. It yields narrative unity and possibly thematic progression, i.e., the description first anchors into what is commonly known (theme) and then accumulates new information (rheme) [143].

The real world events are often grouped, and the group is often given a name. A label or a name such as 'Watergate', '9/11', or 'The Second World War' have come to denote series of events. They are packaged into cultural tradition by contemporaries and historians. In a similar fashion, reporters package real world events into news stories and narratives, and create catchy shorthands to refer to them.

Definition 4.6 *A topic is a name for a news event or a group of related news events.*

A topic is ultimately a choice. It is built around relevance, and is therefore prone to subjectivity. Wide, large-scale topics, like 1997 Asian financial crisis, are bound to remain debatable. The relative instability in research corpora, however, can be decreased by a careful labeling process as was discussed in Section 2.1.5.

In fairness, the original definitions were issued and refined partly to maintain the consistency in the annotators' work when the TDT corpora were produced. The philosophical footwork and new definitions presented here will obviously have no impact on the existing judgments in the corpora. Nevertheless, we have straightened some of the inconsistencies in the terminology.

We shall return to the narrativeness in Chapter 7, when we consider adaptive TDT systems. Furthermore, the Davidsonian analysis of action sentences gives rise to temporal indexing we present in Chapter 6.

4.2 TDT corpora

While the Pilot Corpus was created by the research participants involved in the initial pilot project, the later corpora were products of Linguistic Data Consortium. In addition to TDT corpora, we employ part of TIPSTER corpus for background statistics of English. Like TDT corpora, it is produced by LDC.

The TDT-2 corpus spans from the beginning of January to the end of June 1998 consisting of about 72,000 documents. Initially, the data originated 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 television broadcast sources (Cable News Network "Headline News" and American Broadcasting Company "World News Tonight") and two radio broadcast sources (Public Radio International's "The World" and the Voice of America). Later, this set was augmented with Mandarin radio broadcasts of Voice of America, Mandarin newswire from Xinhua News Service and Mandarin web news from Zaobao news agency.

The six-month period of TDT-2 is divided into training, testing and evaluation sections, each covering two months' worth of material. Table 4.1 presents the number of documents per source and per sections. There are about 10,000 documents with topic labels of 100 events distributed over the length of the corpus. We use training and testing sections to train the system, e.g., to induce the system parameters. The result is evaluated using the evaluation section.

The TDT-3 corpus is an extension to TDT-2 running from October 1998 through December 1998. There are two new English news sources, National Broadcasting Company (NBC) "World News Tonight" and MSNBC "News with Brian Williams" [38]. Table 4.2 presents the number of documents per source and section. The TDT-3 corpus uses TDT-2 as training and testing data. The evaluation section contains about 9300 labeled documents. There is a total of 120 exhaustively annotated topics.

In case of broadcast sources, the corpora TDT-2 and TDT-3 offer a choice of using either automatic or manual transcription. All the data is delivered as a stream of documents, and thus there is no necessity of story segmentation. The topic annotation relied on rules of interpretation defining the sphere of relevant events. The quality and cohesion of the judgments was evaluated in several stages by both system and human evaluators [37].

As of 2009, the TDT-5 is the latest and most comprehensive of the LDC's TDT corpora. It comprises 407,505 documents of three languages: 278,109 English documents, 72,910 Arabic and 56,486 Mandarin respectively. The corpus contains a total of 250 labeled topics. All documents

are newswire, so no automatic or manual speech recognition is involved. Table 4.3 lists the number of stories per source. Systems running tests on TDT-5 are trained with TDT-2 and TDT-3 newswire documents.

Table 4.1: News story counts of TDT-2 corpus by source and section. The types 'nw' and 'bc' stand for newswire and broadcast, respectively. The languages 'eng' and 'man' stand for English and Mandarin.

			News stories			
Source	*Type*	*Lang*	*Train*	*Test*	*Eval*	*Total*
Associated Press (APW)	nw	eng	4,309	4,222	4,229	12,760
New York Times (NYT)	nw	eng	4,057	3,569	4,169	11,795
ABC WNT	bc	eng	626	742	785	2,153
CNN HDL	bc	eng	4,565	5,591	5,629	15,785
Public Radio Intl (PRI)	bc	eng	956	986	971	2,913
Voice of America (VOA)	bc	eng	2,593	3,021	2,600	8,214
Voice of America (VOA)	bc	man	335	771	1,159	2,265
Xinhua News Service (XIN)	nw	man	3,633	3,865	3,779	11,277
Zaobao (ZBN)	nw	man	123	2,497	2,550	5,170
			21,197	25,264	25,871	72,332

Table 4.2: News story counts of TDT-3 corpus by source and section. The 'train' section is composed of TDT-2 corpus. The types 'nw' and 'bc' stand for newswire and broadcast, respectively. The languages 'eng' and 'man' stand for English and Mandarin.

			News stories		
Source	*Type*	*Lang*	*Train*	*Eval*	*Total*
Associated Press (APW)	nw	eng	12,760	7,338	20,098
New York Times (NYT)	nw	eng	11,795	6,871	18,666
ABC WNT	bc	eng	2,153	1,020	3,173
CNN HDL	bc	eng	15,785	9,216	25,001
Public Radio Intl (PRI)	bc	eng	2,913	1,588	4,501
Voice of America (VOA)	bc	eng	8,214	3,952	12,166
NBC	bc	eng	–	859	859
MS-NBC (MNB)	bc	eng	–	685	685
Voice of America (VOA)	bc	man	2,265	3,401	5,666
Xinhua (XIN)	nw	man	11,277	5,153	16,430
Zaobao (ZBN)	nw	man	5,170	3,817	8,987
			72,332	43,900	116,232

Table 4.3: News story counts of TDT-5 corpus by source. All sources are newswire. The languages 'arb', 'eng' and 'man' stand for Arabic, English and Mandarin.

News stories		
Source	*Lang*	*Total*
Agence France Presse (AFA)	arb	30,593
An-Nahar (ANN)	arb	8,192
Ummah (UMM)	arb	1,104
Xinhua (XIA)	arb	33,051
Agence France Presse (AFE)	eng	95,432
Associated Press (APE)	eng	104,941
CNN (CNE)	eng	1,117
LA Times / Washington Post (LAT)	eng	6,692
New York Times (NYT)	eng	12,024
Ummah (UME)	eng	1,101
Xinhua (XIE)	eng	56,802
Agence France Presse (AFC)	man	5,655
China News Agency (CNA)	man	4,569
Xinhua (XIN)	man	37,251
Zaobao News (ZBN)	man	9,011
		407,505

The corpora TDT-2 and TDT-3 were annotated exhaustively for topics such that all the on-topic documents were labeled. Given the volume of TDT-5, the labeling process was "search guided". Thus, the annotators used the first-story of a topic as a query, and iteratively labeled candidates and refined candidates until no on-topic documents were found. The process had a three-hour time limit per topic [7].

TIPSTER corpus is a generic text collection designed for the use of information retrieval evaluation. It contains a variety of document types from patent data to magazine articles. We, however, confine our use to two sources, Associated Press (1989-1990) and Wall Street Journal (1987-1992), both of which pre-date TDT data. The selection endows us with nearly 400,000 documents, as described in Table 4.4.

Table 4.4: News story counts of TIPSTER sources. All documents are English newswire.

News stories			
Source	*Type*	*Lang*	*Total*
Associated Press (AP)	nw	eng	242,916
Wall Street Journal (WSJ)	nw	eng	149,612
			392,528

4.3 Topic tracking

4.3.1 System

The document preprocessing employs tokenization, stop word elimination, and stemming as detailed in Section 3.1.1. The evaluation is done with the TDT scoring tool [3] provided by NIST. It was developed for the yearly benchmarks and is tightly integrated to LDC's TDT corpora and their document formats.

Each topic is defined by $N_t \in \{1, 4\}$ sample documents – subscript t stands for 'training'. The documents up to the last sample are not part of evaluation for that topic. Topics emerging later in the stream are obviously evaluated with fewer documents. Should a topic have fewer than N_t documents, it is not evaluated.

Our system is composed of a document repository, a lexicon and an inverted index. The document repository contains the preprocessed document vectors. Unlike normal retrieval systems, we do not store the news stories themselves, as TDT experiments have no use for them. The lexicon contains all the terms in the collection, their document frequencies, and a pointer to the posting list of the term as illustrated in Figure 3.2. The posting lists are stored in a blocked binary file, i.e., all access to these files is done through fixed size blocks. The postings lists are stored using byte-aligned compression [151].

The sample documents are compiled into a centroid vector that then represents the topic. When a new document enters the system, in the evaluation it is compared against all of the different topics independently. So, if the system tracks 50 topics, then the system has 50 independent and separate topic filters. Ideally, a news story discusses only one topic, but

[3] http://www.nist.gov/speech/tests/tdt/resources.html

independently run topic filters could find the story similar to several topics.

The topic tracking runs were executed in two parts: First, the system ran on the training set in order to find a good decision threshold θ. As discussed in Section 2.2.3, the choice of the threshold affects the trade-off between the misses and false-alarms. Thus, the topic tracking was run on the training set using a variety of thresholds. Then, the system ran on the evaluation set making tracking judgments based on the threshold θ that worked best on the training data.

4.3.2 Baseline run

We ran topic tracking with both cosine (Equation 3.6) and Hellinger (Equation 3.12) similarity, both TFIDF (Equation 3.3) and InQuery (Equation 3.9) term-weighting, and with and without incremental document frequency recording (see Section 3.1.3). All runs employed centroid vectors for topic representation. When incremental document frequency was used, the term weights were recalculated after each document. Based on experiments by Allan et al. [14, 20], the number of terms in the document vector was set to 1,000, i.e., 1,000 best terms based on the used term-weighting were selected to represent the document. We used normalization of Equation 3.11.

Table 4.5 shows the results from TDT-2, TDT-3, and TDT-5 corpora. The table lists miss and false-alarm rates for each run as well as the normalized detection cost of Equation 2.3. The results are topic-weighted, which means - as was discussed in Section 2.2.3 - that the error rates and costs are first determined for each topic and then averaged over all of the topics. The values on the left result from running the system with the trained similarity threshold θ. We use values $\theta = 0.10$ for $N_t = 1$ runs and $\theta = 0.12$ for $N_t = 4$ runs. The minimum values on the right were obtained by sweeping the threshold across the space of decision scores and choosing the optimal. This minimum is a by-product of the NIST TDT evaluation tool. The purpose of presenting such minimum is the same as with DET curves: to eliminate the speculation regarding the choice of threshold. The minimum merely wraps the effectiveness into a single number.

In Table 4.5, the systems with 'i' at the end of the identifier were run with incremental document frequency (see Section 3.1.3). Over all, it seems to decrease the detection cost, but not very much. This observation is similar to that of Braun and Kaneshiro regarding first-story detection [31]. Considering the computational cost it carries, the gains are hardly worth the effort.

Table 4.5: The baseline topic tracking results. System 'cos' uses cosine and 'hel' Hellinger similarity coefficient. The results on left employ threshold θ that is acquired from a training run. The results on the right represent the minimal detection cost obtained from the output data.

		Topic-Weighted at θ			Minimum Topic-Weighted		
System	N_t	$p(\overline{r}\|\omega)$	$p(r\|\overline{\omega})$	$(C_{det})_{norm}$	$p(\overline{r}\|\omega)$	$p(r\|\overline{\omega})$	$(C_{det})_{norm}$
TDT2							
cos tfidf	1	0.1063	0.0069	0.1401	0.0982	0.0073	0.1341
cos tfidf i	1	0.0935	0.0084	0.1349	0.0937	0.0082	0.1304
cos inquery	1	0.1024	0.0067	0.1353	0.0893	0.0083	0.1298
cos inquery i	1	0.0858	0.0081	**0.1256**	0.0755	0.0095	**0.1220**
hel tfidf	1	0.1449	0.0109	0.1982	0.1153	0.0146	0.1868
hel tfidf i	1	0.1481	0.0099	0.1964	0.1238	0.0128	0.1864
hel inquery	1	0.1389	0.0089	0.1828	0.1130	0.0109	0.1664
hel inquery i	1	0.1433	0.0084	0.1843	0.1120	0.0112	0.1669
cos tfidf	4	0.0953	0.0117	0.1526	0.1009	0.0089	0.1446
cos tfidf i	4	0.0909	0.0125	0.1523	0.1055	0.0079	0.1443
cos inquery	4	0.1223	0.0071	0.1570	0.1010	0.0092	0.1461
cos inquery i	4	0.1109	0.0079	**0.1498**	0.0955	0.0097	**0.1431**
hel tfidf	4	0.1410	0.0187	0.2326	0.1528	0.0153	0.2279
hel tfidf i	4	0.1426	0.0180	0.2307	0.1511	0.0158	0.2286
hel inquery	4	0.1429	0.0162	0.2221	0.1556	0.0124	0.2163
hel inquery i	4	0.1438	0.0163	0.2239	0.1560	0.0122	0.2159
TDT3							
cos tfidf	1	0.1071	0.0114	0.1630	0.0971	0.0133	0.1623
cos tfidf i	1	0.0885	0.0156	0.1651	0.0991	0.0127	0.1613
cos inquery	1	0.1107	0.0098	0.1586	0.0895	0.0131	0.1536
cos inquery i	1	0.0963	0.0118	**0.1539**	0.0870	0.0133	**0.1522**
hel tfidf	1	0.1350	0.0172	0.2195	0.1332	0.0175	0.2188
hel tfidf i	1	0.1437	0.0155	0.2199	0.1437	0.0155	0.2196
hel inquery	1	0.1204	0.0162	0.2000	0.1216	0.0158	0.1991
hel inquery i	1	0.1268	0.0154	0.2022	0.1282	0.0146	0.2000
cos tfidf	4	0.1185	0.0104	**0.1694**	0.1008	0.0130	0.1645
cos tfidf i	4	0.1267	0.0090	0.1710	0.1004	0.0129	0.1635
cos inquery	4	0.1438	0.0074	0.1803	0.0981	0.0122	0.1580
cos inquery i	4	0.1441	0.0072	0.1794	0.0836	0.0146	**0.1551**
hel tfidf	4	0.1576	0.0188	0.2497	0.1712	0.0156	0.2477
hel tfidf i	4	0.1899	0.0127	0.2524	0.1469	0.0207	0.2484
hel inquery	4	0.1710	0.0133	0.2363	0.1421	0.0185	0.2326
hel inquery i	4	0.1671	0.0142	0.2366	0.1426	0.0184	0.2326
TDT5							
cos tfidf	1	0.1671	0.0112	0.2219	0.1222	0.0156	0.1987
cos tfidf i	1	0.1291	0.0171	0.2127	0.1546	0.0088	0.1976
cos inquery	1	0.1257	0.0220	0.2335	0.1426	0.0129	0.2059
cos inquery i	1	0.1225	0.0184	0.2125	0.1507	0.0083	**0.1913**
hel tfidf	1	0.1350	0.0172	0.2195	0.1332	0.0175	0.2188
hel tfidf i	1	0.1437	0.0155	0.2199	0.1437	0.0155	0.2196
hel inquery	1	0.1204	0.0162	**0.2000**	0.1216	0.0158	0.1991
hel inquery i	1	0.1268	0.0154	0.2022	0.1282	0.0146	0.2000
cos tfidf	4	0.1172	0.0250	0.2395	0.1528	0.0154	0.2285
cos tfidf i	4	0.1209	0.0237	**0.2370**	0.1578	0.0123	**0.2181**
cos inquery	4	0.1177	0.0274	0.2521	0.1522	0.0148	0.2249
cos inquery i	4	0.1177	0.0274	0.2521	0.1522	0.0148	0.2249
hel tfidf	4	0.1682	0.0221	0.2765	0.1979	0.0141	0.2671
hel tfidf i	4	0.1557	0.0278	0.2918	0.2019	0.0137	0.2692
hel inquery	4	0.1416	0.0278	0.2777	0.1784	0.0154	0.2537
hel inquery i	4	0.1457	0.0271	0.2785	0.1886	0.0138	0.2561

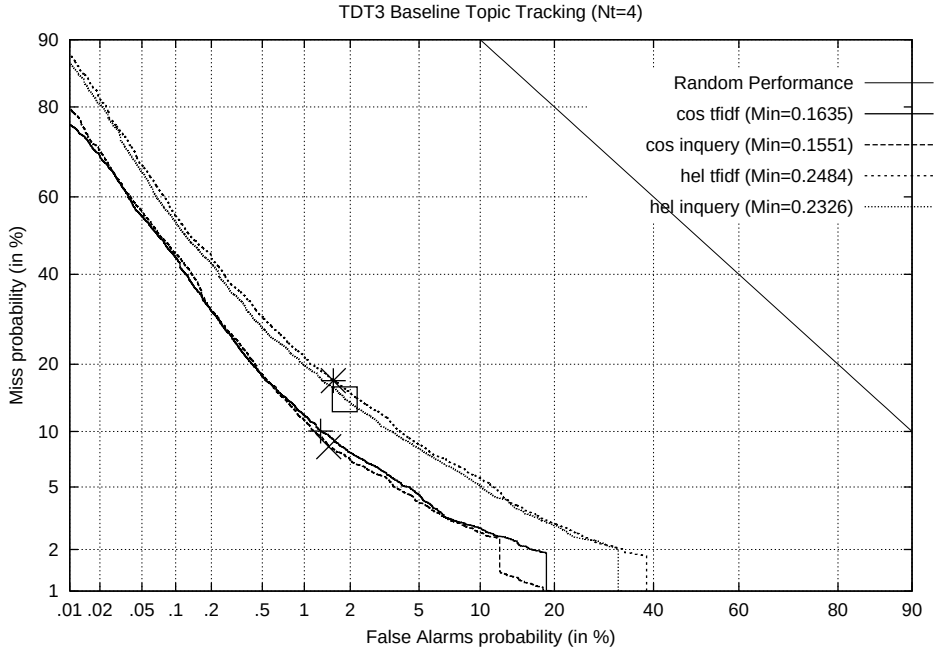

Figure 4.1: The DET curves of TDT3 baseline topic tracking ($N_t = 4$) with minimum costs marked.

Based on the results, it is safe to say that cosine is more suitable as a similarity coefficient for topic tracking than Hellinger. Consistently, in all corpora, the error-rates for Hellinger are higher. Figure 4.1 represents the DET curves of TDT-3 tracking runs with four sample documents. The gap between the curves of the two similarity coefficients is a clear one. For cosine, when miss-rate is at 1-2%, the difference in false-alarm rates is around 15-20 percentage points. Similar phenomenon is reported by Chen, Farahat, and Brants [35] with regarding story-link detection. Based on comparison of score distributions, they suggest that cosine works better in the high precision area while Hellinger triumphs in high recall area of the score distribution. Story-link detection and apparently topic tracking are biased towards high precision and first-story detection towards high recall, respectively.

In general, the InQuery term-weighting effectiveness exceeds that of TFIDF in Table 4.5. This seems to be opposite to the results by Allan et al. [14, 20]. They were using raw term frequency, which in our experience performs worse than the logarithm variation we are using. The DET curves

of Figure 4.1 are a bit more ambiguous about the gains: the curves overlap for a great part. The difference might not be statistically significant.

The difference between the optimal effectiveness and the effectiveness at threshold θ is in most cases reasonably small. Obviously, there are some minor differences in the training and evaluating corpora and topics. The threshold obtained from TDT-2 using four sample documents was not a very good one for InQuery. This may be due to small number of topics with four sample documents in the evaluation set of TDT-2.

In these baseline runs, there is a slight decrease in effectiveness as the number of sample documents is increased from 1 to 4. This seems counter-intuitive, as one would expect the error-rates to drop as the system has more information. Apparently, increasing the number of sample documents increases the term-weights of the common terms. It also increases the overall number of terms as not all news document contain 1000 terms. Thus, when comparing the $N_t = 4$ centroid to a short broadcast news document containing some of the same, possibly otherwise rare terms, the comparison yields higher scores than each of the sample document would yield alone.

4.3.3 Failure analysis

In order to identify causes of tracking errors, we examined 200 errors – 100 randomly drawn misses and 100 randomly drawn false-alarms – from TDT-2 evaluation data manually. We settled for subsets as the analysis of all errors would have required inspection of several thousand documents. The system was given one document as a sample (i.e., $N_t = 1$), and the similarity was measured with cosine similarity coefficient and TFIDF weights. In topic tracking, a miss boils down to not having a high enough similarity score. The greatest single cause for missing relevant documents was automatic speech recognition in Mandarin Chinese. We counted 46 misses where crucial named entities were lost due to erroneous speech recognition. For example, the documents discussing the Israeli-Palestinian peace negotiations in London contained two important figures, prime minister Netanyahu and PLO leader Arafat. In all the Mandarin news stories these were 'tower Nepal Asia Hu' and 'Alfate'. All the main characters in the Monica Lewinsky scandal apart from President Clinton were similarly garbled in translated documents.

Another source of problems was in the changed focus of the reporting. In 18 misses the report discussed the same topic but a different aspect. Hence, there were few common terms. For instance, the first story reporting a train accident near Hamburg, Germany, starts with Prime Minister Blair

sending his condolences. Yet, the relevant documents quite soon move onto investigations and domestic debate over blame in Germany.

In twenty-odd misses either the document vector was short or contained no common named entities with the topic. In five cases we would disagree with the annotation. The rest had common terms, but just not enough. Some topics were simply difficult to track. The documents discussing the incentive programs companies are establishing to attract skillful employees displayed a variety in vocabulary describing benefits and perks. Since few named entities were involved, the miss-rate remained high.

False-alarms were not plagued by automatic speech recognition (ASR) problems. A false-alarm occurs when the similarity score is too high, and mistranslations are more likely to decrease the score than to increase it. Slightly over 20 false-alarms resulted from different instances of similar events or topics. Human rights issues were discussed in relation to Ethiopia and China, and the system would confuse the two different instances.

Two lengthy documents covering political decision making or debate are likely to share a great deal of vocabulary. Often, they have some common people (e.g., President Clinton), some common organizations (e.g., Democrats, Republicans or Congress), and same places (e.g., Washington, D.C.). Even if these names occur often, they do not occur as often as common nouns or verbs, so their information content measured by inverted document frequency is fairly high. In a sparse vector-space, the comparison of such documents is likely to result in positive judgment. We found about thirty such cases in areas of politics and culture. Also, long documents might make brief references to other events; we attributed four false-alarms to these passing references.

On the other hand, short documents were also problematic. Since the cosine similarity coefficient consists of a length normalization, terms occurring in a short vector gain more emphasis than same terms occurring in longer document. Mere appearance of place names and persons like 'Clinton' or 'Kofi Annan' along with some general terms resulted in a false positive judgment in about ten cases. With false-alarms, we would dispute only one annotation.

To better understand the role of named entities, we compared the intersection of terms in positive on-topic and off-topic comparisons, i.e., true and false positives or, equivalently, hits and false-alarms. We listed up to 20 most frequent terms per topic the centroid and the document had in common to see what kind of terms constitute positive judgments. The idea was to pin down the "topic terms", i.e., the kind of terms that documents in the same topic will share or "strong" terms that are likely to

Table 4.6: A listing of term types occurring in both the centroid and the document in on-topic and off-topic comparisons. The type 'adjective' refers to nation, religion and region, e.g., 'Turkish', 'Muslim', and 'Asian'.

type	*on-topic*	*off-topic*
location	55	41
person	44	29
adjective	19	29
organization	13	21
other	8	3
total	139	123
all terms	834	1131

cause false-alarms.

We tagged each term manually to types listed in Table 4.6. Some topics had only few positive samples, and their intersection would not yield 20 terms. Hence, the lower total count in on-topic column.

About 17% of the terms thus selected are named entities in on-topic category and about 11% in off-topic category. References to locations and people seem to dominate the on-topic named entities. Adjectives and organizations appear less useful.

Another way to examine topic-terms is through document frequency. If a term has a high document frequency, it occurs in many documents across the topics and is less informative and less useful in document similarity. Table 4.7 describes the document frequency distribution for the terms of Table 4.6. The total number of documents was about 440,000: a combination of term occurrences in TIPSTER and TDT2-training corpora. In on-topic comparisons, 29% of the terms occur less than 10 000 times in the corpus (vs. 17% in off-topic). Most of the matching terms are relatively common.

What does all this tell us? The topic-terms are composed of relatively common terms, which is likely to induce false-alarms. It has been noted that increasing the term weights of named entities in the document vectors does not decrease the detection error-rates [110]. Table 4.6 suggests that although locations and names are characteristic to on-topic documents, they are so only with a margin.

In topic tracking, the question of relevance may hinge on the presence or absence of some particular place name or person. The reports of riots in Yemen and Indonesia share a great deal of general terms that outweigh

the difference in named entities. If we could override the common term similarity with the lack of geographical support, we could discourage the association between documents that are about similar but not same events. In Chapter 5, we will investigate an approach of processing and comparing named entities separately from the common terms.

4.3.4 Comparison to previous work

Comparing the scores directly to the works of others is hampered by changes in the cost-function, deferral periods, changes in the data, and variant experimental setups. Many reports simply plot the DET curves without anchoring any points either at threshold or the topic-weighted minimum.

Table 4.8 lists the best results from the yearly TDT benchmark evaluations organized by NIST [2, 3, 4, 6]. We restricted the selection to those using manually defined (reference) boundaries, English content, and (translated) multilingual sources.

From the table we see that our baseline seems on par with TDT-2 results. However, our evaluation contains all topics occurring in the evaluation set, not just the 33 topics designated for evaluation, because otherwise $N_t = 4$ experiments would have very few topics. By restricting to the evaluation topics only, the cost-function of our system drops below 0.10 for $N_t = 1$ and below 0.12 for $N_t = 4$.

In earlier TDT-3 experiments, the basic cosine TFIDF approach is outperformed by language models. The LIMSI system used unsupervised adaption as well as document vector expansion which instead of decreasing the number of terms augmented the topic model with term probabilities from related documents. However, the TDT-3 results presented here used only 60 topics (60 for 2000 and another 60 for 2001), while we have used all 120

Table 4.7: A listing of document frequency ranges in on-topic and off-topic comparisons.

document frequency	*on-topic*	*off-topic*
$10 \leq$ df	3	0
$10 <$ df ≤ 100	9	4
$100 <$ df $\leq$ 1,000	34	22
1,000 $<$ df $\leq$ 10,000	193	165
10,000 $<$ df $\leq$ 100,000	477	665
100,000 $<$ df $\leq$ 1,000,000	118	275
	834	1131

Table 4.8: Topic tracking results by other groups (LM = language modeling, TFIDF = cosine TFIDF). TDT2 results have been normalized and modified to use $c_{fa} = 0.1$ instead of the contemporary 1.0 to enable comparison, and so the results might not portray the best performance. Note: These are results at threshold θ, not optimal decision trade-off points.

team	*year*	*system*	N_t	$(C_{det})_{norm}$
		TDT2		
Dragon [154]	1998	LM	4	0.1294
UPenn [128]	1998	TFIDF	4	0.1313
BBN [69]	1998	LM	4	0.1347
		TDT3		
LIMSI [87]	2001	LM + adapt + expand	1	0.1213
IBM	2000	vector-based	1	0.1238
UMass [20]	2001	TFIDF	1	0.1655
IBM	2000	vector-based	4	0.1021
LIMSI [87]	2001	LM	4	0.1415
		TDT5		
CMU [159]	2004	rocchio + adapt	1	0.0707
UMass [40]	2004	TFIDF + adapt	1	0.1545
CMU [159]	2004	rocchio + adapt	4	0.0645

topics. There was no information on IBM system in the NIST data, nor have the authors published research papers on their findings.

In TDT-5 the best results have been obtained using a system with Rocchio classifier that makes use of both positive and negative samples and adaptation. If the system determines an incoming document is on-topic, the document is added to the pool of positive samples and the topic representation is re-computed.

4.4 First-story detection

4.4.1 First-story detection system

The document preprocessing and system evaluations are the same as for topic tracking. However, by convention the first-story detection is run with native English data only [1, 5].

Our FSD system uses a simple single-pass clustering based on Algorithm 2: on encountering a new document, the closest match is retrieved from the previously encountered documents. If it is not similar enough,

the document is considered new. Otherwise, it is considered old. Thus, the system is a 1-nearest neighbor: The topics are represented by previous documents themselves; no topic models or centroids are compiled.

4.4.2 Baseline run

Table 4.9 presents the baseline first-story detection results. The results are topic-weighted and comprise error rates and detection cost of Equation 2.3 for both the threshold θ obtained from training runs with the training set and the minimum cost obtained from iterating through the system output.

In most cases, the minimum topic-weighted scores are higher for Hellinger coefficient than for cosine coefficient – even after careful examination of the implementation and a re-run. This is contrary to previous work by Brants, Chen, and Farahat [29] that showed considerable gains in effectiveness with Hellinger compared to cosine in FSD. However, in Table 4.9, the TDT-5 results show slight gain in favor of Hellinger. Similarity, in Table 4.5, the difference between Hellinger and cosine topic tracking costs was lower with TDT-5 ($N_t = 1$) than with the other corpora. The impact of the size of the corpus and the average document length to Hellinger similarity coefficient is left for future work.

Like in topic tracking, the runs with incremental document frequency are labeled with letter 'i' in the system identifier. It provides small gains, but nothing that would justify the computational cost it imposes. InQuery term-weights work slightly better on TDT-2 data, but on the other data sets the TFIDF prevails.

A striking phenomenon emerges from cross-comparison of results between data sets. The effectiveness declines as the size of the data set increases. Larger amount of data seems to increase the likelihood of sporadic match with a non-relevant but sufficiently similar document. The vector-space becomes more "dense". As a result, the performance on TDT-5 is considerably worse than on the other two. The detection cost at threshold is above 1.0, which means we would be better off by just deciding "NO" for all documents. One way to tackle the problem is to confine the search space to, say, 30,000 latest documents.

Figure 4.2 illustrates the DET curves of FSD with TDT-3 data. The curves are more choppy than in topic tracking. The reason is that they represent topic-weighted averages, and there are very few first-stories to average over. As a result, when the decision threshold is iterated through the space of decision scores, the error-rate estimates evolve in large steps, especially in zones of high probabilities of miss or false-alarm. In TDT-3 there are 105 first-stories, so missing one of them increases miss rate by

about 1%.

Table 4.9: The baseline first-story detection results showing the miss and false-alarm probabilities and normalized detection score.

	Topic-Weighted at θ			Minimum Topic-Weighted		
System	$p(\overline{r}\|\omega)$	$p(r\|\overline{\omega})$	$(C_{det})_{norm}$	$p(\overline{r}\|\omega)$	$p(r\|\overline{\omega})$	$(C_{det})_{norm}$
TDT2						
cos tfidf	0.2321	0.1242	0.7981	0.3929	0.0149	0.4658
cos tfidf i	0.2143	0.1253	0.7807	0.3929	0.0149	0.4658
cos inquery	0.1964	0.1061	0.6845	0.4107	0.0091	0.4552
cos inquery i	0.2500	0.0984	0.7139	0.4107	0.0085	**0.4523**
hel tfidf	0.3929	0.0341	**0.5251**	0.3929	0.0341	0.5602
hel tfidf i	0.3929	0.0345	0.5266	0.3929	0.0342	0.5605
hel inquery	0.2143	0.1184	0.7689	0.4821	0.0140	0.5508
hel inquery i	0.2143	0.1156	0.7550	0.4821	0.0140	0.5508
TDT3						
cos tfidf	0.3714	0.0583	0.6628	0.4095	0.0310	**0.5614**
cos tfidf i	0.4571	0.0281	**0.5889**	0.4095	0.0312	0.5622
cos inquery	0.4095	0.0469	0.6365	0.4286	0.0324	0.5874
cos inquery i	0.4095	0.0468	0.6360	0.4476	0.0285	0.5871
hel tfidf	0.4857	0.0385	0.6745	0.5143	0.0312	0.6672
hel tfidf i	0.4857	0.0392	0.6767	0.5143	0.0319	0.6704
hel inquery	0.4000	0.0596	0.6945	0.5333	0.0150	0.6070
hel inquery i	0.4095	0.0566	0.6881	0.5333	0.0150	0.6070
TDT5						
cos tfidf	0.8095	0.0156	**0.9129**	0.7857	0.0191	0.8793
cos tfidf i	0.8095	0.0168	0.9185	0.7540	0.0265	0.8838
cos inquery	0.8333	0.0163	0.9343	0.8492	0.0024	0.8608
cos inquery i	0.8333	0.0159	0.9322	0.8492	0.0034	0.8659
hel tfidf	0.6905	0.0597	1.0196	0.7937	0.0275	0.9286
hel tfidf i	0.6905	0.0602	1.0224	0.7937	0.0284	0.9330
hel inquery	0.8571	0.0141	0.9602	0.7302	0.0243	**0.8492**
hel inquery i	0.7857	0.0160	0.9150	0.7698	0.0164	0.8504

While in topic tracking, the estimates of "optimal" cost trade-offs were in mid-section of the curve, here they are in upper parts near or at the 50% miss-rate level. So, in the best case (as measured by the cost-function), about a half of the first-stories are lost.

4.4.3 Failure analysis

As there are very few target documents in FSD, it is possible to examine all the decisions made in the TDT-2 corpus. In the 25 misses we encountered with TDT-2 evaluation, eight targets were lost because there were good enough hits by chance. In four cases, we determined the topics overlapped

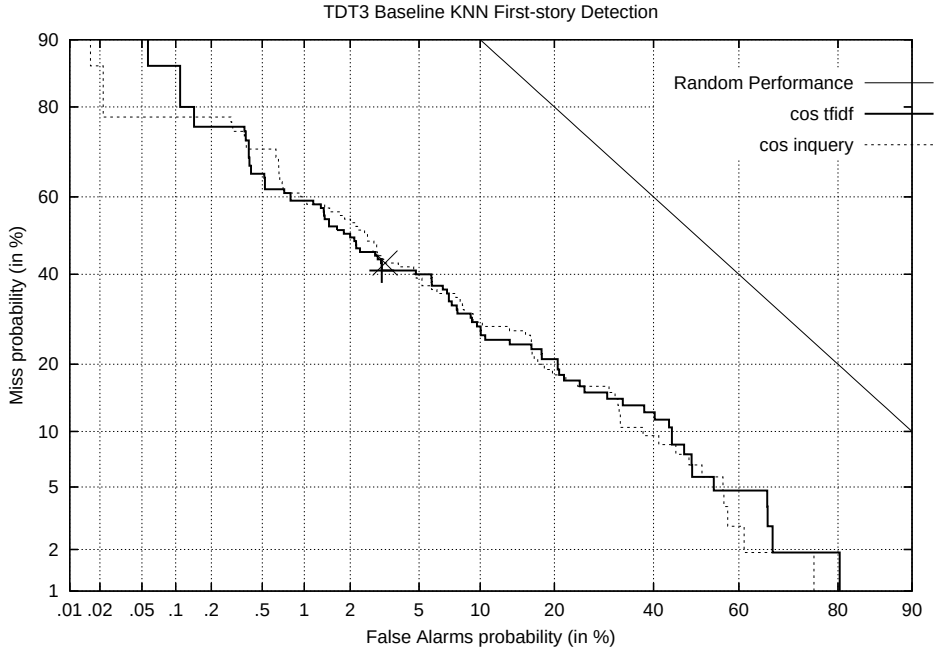

Figure 4.2: The DET curves of TDT3 first-story detection.

each other, e.g., different aspects of related acts of violence in Indonesia, and it would be difficult even for humans to make the consistent judgments. Seven of the misses discussed the same people yet who were involved in different events. For instance, President Clinton was in Eugene, Oregon, to meet victims of a school shooting and the same day in Oregon his plane landed at an airport where a possible bomb was found. In three cases, the topics were instances of the same topic type, and in two cases we would disagree annotation. Overall, the problems of FSD misses are the same as with tracking false-alarms. ASR problems were fewer since the FSD evaluation uses only native English material.

A false first-story is a tracking failure: the document is related to something old, but the system decides otherwise. In the 18 false-alarms of TDT-2 evaluation, 7 were due to a low number of otherwise general terms between a long and a short document. In 11 cases, there was simply not enough common terms. The documents could have discussed different aspects of the same topic, and thus share only few terms.

4.4.4 Comparison with previous work

Table 4.10 presents results from some of the previous experiments by others. Most approaches combine cosine coefficient with semantical representations of the documents. Language models have not transferred their good performance in topic tracking to first-story detection.

Although TDT system evaluation is seldom reported using TDT-2 and TDT-3 these days, our system appears to outperform the other systems, which is surprising for a run-of-the-mill cosine-TFIDF system. Again, the difference may be due to evaluation topics sets. TDT-3 evaluations in 2000 and 2001 used only 60 topics each while we have utilized all 120. With TDT-2, we have used all labeled topics present in the data, not just those designated for evaluation.

Our system does not do as well with TDT-5 corpus. Other systems have managed to maintain a higher level of effectiveness with the increased number of documents and topics. CMU [162] and IBM [6] both employ some sort of windowing, i.e., their systems confine the scope for candidate documents. UMass [40, 77] have combined named entities and other features to modify the cosine coefficient score. Stottler Henke [31] employed co-occurring sentences and locations in pushing down the error rates. Furthermore, the benchmark evaluation configuration of 2004 allowed a deferral period of 10 documents, and so the decisions were made "10 documents later".

4.5 Conclusions

There has been little discussion in TDT about the core concepts of events and topics. The concepts are often used interchangeably neglecting the fundamental difference between, what happens and what somebody says will or has happened. In this chapter, we put forward more accurate definitions for events and topics. An event is a phenomenon in the real world. Some events are reported in the news, and the media compiles them into news events that are literary products. Topics are groups of news stories that are found related on the basis of events they discuss.

We presented a baseline system for topic tracking and first-story detection. It is based on vector-space model, and utilizes run-of-the-mill preprocessing with tokenization, stemming, and stop-lists. In topic tracking, the effectiveness was similar to those of vector-space systems reported by others suggesting our experiments are set up correctly. Furthermore, we found incremental InQuery term-weights with cosine similarity coefficient to perform the best. A failure analysis uncovered problems with automatic speech

Table 4.10: First-story detection results by other groups. Results with year indication are from NIST benchmark evaluations.

team	*year*	*system*	$(C_{det})_{norm}$
		TDT2	
UMass [77]		semantic cosine variant	0.5400
Papka [110]		cosine + time-decay	0.6262
		TDT3	
PARC [29]		cosine variant	0.5874
CMU&UCL [161]		language model	0.6901
UMass [14]	2001	TFIDF	0.7729
UMass [14]	2000	TFIDF	0.8110
		TDT5	
UMass [77]		semantic cosine variant	0.6610
Stottler Henke [31]	2004	cosine + loc. linkage + sent. linkage	0.7155
UMass [40]	2004	cosine inquery + NE	0.7603
CMU [162]	2004	cosine TFIDF + window	0.8359
IBM [6]	2004	cosine TFIDF + MaxEnt + window	0.8601

recognition that garbled at least many of the named entities. Other sources of error included shifting focus of the topics, overlapping vocabulary, similarity of the reported events and varying lengths of the documents. We also verified that a named entities are important in determining the document similarity.

In first-story detection the effectiveness of our baseline was slightly better than that of other systems – except for experiments on TDT-5. The other systems were more elaborate than just plain cosine-TFIDF. They have used adaptiveness, named entities and document windows to push down the error rates that generally seem to increase for FSD with larger corpora. Although pointing out specific sources of errors is more difficult in FSD than in topic tracking, the failure analysis of our baseline first-story detection misses seem to coincide with causes of topic tracking false alarms: similar, yet not the same events being reported, short documents with strong named entities, sporadic references to other events.

Chapter 5

Ontology-based document similarity

We have already seen that this notion of 'only playing' in no way excludes the possibility of realizing this 'only playing' with extreme seriousness.
– Johan Huizinga, *Homo Ludens*

In previous chapter, we noted the importance of named entities in topic detection and tracking. Now, we lay out a framework for document similarity that incorporates named entities – or any group of terms – and their ontologies, which builds up on vector-space model.

In Section 5.1 we present an ontology-based document similarity approach that splits the term space into semantic classes. Each of these classes can be assigned its own similarity measure that can be based on an ontology, for instance. In Section 5.2, we discuss ways of comparing terms of the semantic classes. Section 5.3 presents ways to compare two semantic vectors of the same class, and Section 5.4 discusses approaches to combine the class-wise similarities into a single score. Section 5.5 presents experiments, and Section 5.6 is a conclusion.

5.1 Similarity of hands

In the vector-space model, the similarity is based on the term-term identity, that is, non-identical but related terms are not considered. In the light of the results by Allan et al. [16], we wish to investigate the possibility of basing the similarity of two documents on the co-occurrence of *similar* or *related* terms, and thus relaxing the term-term similarity from binary relation to a continuous scale. The term-term similarities can be derived

from an external ontology or ontologies.

The idea of isolating named entities and comparing them separately is not novel. In Section 3.3 we outlined the previous work in this area. Much of the previous work shares the same basic ideas. Our objective is to generalize and formalize the numerous approaches, and then take one step further by introducing designated similarity coefficients for each isolated type of entities.

5.1.1 An analogy

For a moment, let us consider two hands of cards H_1 and H_2 drawn from two different decks illustrated in Figure 5.1. How similar are they? The most obvious way would be to look at the common cards. With the hands having three cards in common, we might not consider H_2 similar to H_1, but obviously there are hands that are less similar to H_1, and hands that are more similar to H_1.

suits	H_1	H_2	$H_1 \cap H_2$
♠	2, 3, K	3, 7, Q	3
♢	6, 7, 10	5	
♡	4, 6, 7, A	4, 7, K	4, 7
♣	3, 4, 10, J, Q	9	

Figure 5.1: Two hands of cards, H_1 and H_2, drawn from two different decks.

People do not view the similarity of hands only in terms of card-card identity. As an evidence, there is a long tradition of card games each of which presents similarities between two cards based on suit, color or value. For instance, in Freecell solitaire two cards are interchangeable, if they are of the same color and value. In Euchre, the relationships between the four jacks depend on the trump suit. In Poker, two hands can be equivalent although they might not have any common cards. Consider, for example, two flushes:

$$\{\spadesuit 6, \spadesuit 5, \spadesuit 4, \spadesuit 3, \spadesuit 2\} \equiv \{\heartsuit 6, \heartsuit 5, \heartsuit 4, \heartsuit 3, \heartsuit 2\}$$

In Poker, all suits are of equal weight and ties are broken with high cards. Thus, the two hands are equivalent, that is, literally, 'of equal value'.

The idea is that the cards are not taken merely as tokens. They have a *meaning* defined by the rules of the game. The meaning of a card does not depend directly on its name (e.g., ♣7 or ♡3), but on how it is related to

other cards in the game. A game [1] defines the similarities between cards; the role of colors, suits and values may vary. The cards are similar, if they can be played in a similar way.

Words can be viewed as "cards" of sort. Wittgenstein argued that *"The question "What is a word really?" is analogous to "What is a piece in chess?""* [152]. The meaning of a word is in the way people use it to express ideas successfully. The communication between people is composed of *language games* in which each word has an inherent logic that defines valid and invalid uses much like valid moves of chess pieces. The rules of the language game are fundamentally a convention known to members of the culture. The sentence *"John moved to France"* makes sense to anyone sufficiently familiar with English. The sentence would still be understandable, if we replaced 'France' with 'Paris' or 'Burgundy'. All these words denote a geographic place or an area, and their inherent logic is similar. Instead of 'John' we could say 'Company Ltd.', and instead of 'moved to' we could say 'drove to'. We are not pointing to the syntactical similarity of the words in particular, but the ontological likeness of the things they denote. If they were cards or chess pieces, they could be played in a similar way.

Much like hands of cards, two news stories can be similar in many ways. The vector-space model examines the intersection of terms and transforms it into a metric distance function that has some obvious mathematical and computational advantages. However, the similarity itself needs not lend itself to symmetry or triangle-inequality [145]. Determining similarity of two documents can make use of similarity of certain kinds of words, the lack of similarity in people they discuss, and presence of strong geographical overlap. We explore the document similarity as a card-game of sort.

5.1.2 Semantic classes

Assume there are groups of terms that have the same *kind* of meaning and that in language game, the terms would have similar inherent logic and therefore similar valid moves. We organize the space of all terms into these groups, which we call *semantic classes*. For each class, we assign a term-term correlation that expresses the semantical similarity between two terms.

Definition 5.1 *Let $\mathcal{T} = (t_1, t_2, \ldots, t_n)$ denote the termspace. A semantic class i is a unary relation $S_i \subset \mathcal{T}$.*

[1] The term 'game' is used in a broad sense and not in relation to game-theory.

Virtually any subset of terms can be a semantic class. The similarity function defines a value for each pair of terms within the semantic class – this can be seen as a crude form of semantics. Such similarity function could tell us that 'Paris' and 'Lyon' are somewhat related terms as are 'dog' and 'cat'. In a sense, a high similarity between terms would correlate with high interchangeability in the analogy presented above. One can always use the *binary* similarity that yields 1 for all identical terms and 0 for all the rest implying that there is interchangeability if and only if the terms are identical. One could also adopt a distance in a phylogenetic tree [135], a taxonomy of concepts, music genres or word senses [59, 118], a geographical distance, a distance on the time-axis, or a distributional similarity [82] such that any pair of terms in the same group can be assigned a (*non-binary*) real value denoting the similarity between the meanings of the terms. We think of a semantic class as a sort of "suit" of the term and the term itself as the "value" within that suit.

The notion of similarity can become overloaded in this terrain. We follow the nomenclature of Salton [122] and use term-term correlation to denote the pairwise similarity of terms.

Definition 5.2 *The term-term correlation of semantic class i is a function $cor_i : S_i \times S_i \to \mathbb{R}$.*

The pairwise correlation of terms $cor_i(a, b)$ can be expressed as an $n \times n$ matrix $\mathbf{T}$. The retrieval operation of Equation 3.4 would then be modified to

$$\mathbf{r} = \mathbf{DTq}, \tag{5.1}$$

where $\mathbf{D}$ is an $m \times n$ document matrix and $\mathbf{r}$ is a m-vector expressing the degree of association between the query $\mathbf{q}$ and each document vector $\mathbf{d_i}$ [122]. Augmenting the retrieval operation with term-term similarities $\mathbf{T}$ effectively augments the query vector to contain non-zero values for related or similar terms in addition to the original ones. This is similar to *query expansion*, which adds related terms (e.g., per thesaurus) to the query vector [99].

The introduction of semantic classes imposes changes on the document representation. As each semantic class is assigned a term vector, the document comprises multiple vectors transforming into what we call a *multi-vector*. Each vector is compared with a similarity coefficient such as cosine defined in Equation 3.6, Hellinger in Equation 3.12, or asymmetric clarity adjusted divergence in Equation 3.17, for example.

Definition 5.3 *A similarity coefficient $sim_i : S_i^n \times S_i^n \to \mathbb{R}$ represents the similarity between two vectors of semantic class i.*

In Equation 5.1 the inner-product of Equation 3.5 is modified to include the term-term correlations. It can be written out as follows:

$$sim_i^{mod}(\mathbf{d_i}, \mathbf{q}) \quad = \quad \sum_{k=1}^{n} \sum_{j=1}^{n} cor_i(d_{ik}, q_j) d_{ik} q_j. \tag{5.2}$$

Now the semantically augmented inner-product iterates over all pairs of terms instead of just the term-space. We get the standard inner-product by placing identity relation as the correlation cor_i, i.e., $cor_i^{id}(a, b) = 1$ when $a = b$, else it is 0. This would be equivalent of matrix $\mathbf{T}$ being an identity matrix, $\mathbf{T} = \mathbf{I}$, in Equation 5.1.

When determining the similarity of multi-vectors, the comparison is carried out class-wise with corresponding similarity coefficient. While the comparison of two traditional document vectors produces one real-valued similarity score, the result of class-wise comparison of two multi-vectors, given k semantic classes, is a score vector $\mathbf{x} \in \mathbb{R}^k$. In order to produce a single coefficient on which to make judgments, we employ *score models*, i.e., functions that transform score vectors $\mathbf{x}$ into a single real value. This real value is the decision score. In previous work, the class-wise similarity scores have been combined into single decisions scores by classifiers [105, 77] and various heuristic functions [40, 36, 50, 56, 66, 70, 113, 114, 74].

Definition 5.4 *A score model,* $score : \mathbb{R}^n \rightarrow \mathbb{R}$*, transforms a class-wise similarity score vector* $\mathbf{x} \in \mathbb{R}^k$ *of* k *semantic classes into a single similarity score.*

The semantic classes, their similarity functions, and the score model can be almost anything just as the rules in a card game. When people talk about the similarity of two books, they may be referring to the shape, color, author, structure, typography, genre, or theme; books might be found similar because the same customers bought them. There is no one single universal approach to determining the likeness of books, because in a discussion the similarity can be based on *ad hoc* criteria. Thus, it would seem people can play different language games that allow different concepts of similarity. The use of semantic classes paves the way for introducing these various dimensions to similarity or, if you like, language games.

Definition 5.5 *A signature* Σ *consists of the semantic classes, their similarity functions, and the score model to combine them*

$$\Sigma = \{S_1, S_2, \ldots, S_n, cor_1, cor_2, \ldots, cor_n, sim_1, sim_2, \ldots, sim_n, score\}.$$

Defining a signature Σ is really just a shorthand for a set of semantic classes and their similarity measures. The terms occurring in the document are not divided to just any subsets, but to those defined by the language Σ. A document representation employing these classes and functions is a Σ-structure.

Definition 5.6 *Let $\mathcal{T} = (t_1, t_2, \ldots, t_n)$ denote the termspace. $\mathcal{D}$ is a Σ-structure $\mathcal{D} = (\mathcal{T}, \Sigma)$ of document D.*

The introduction of Σ-structures is just another way of saying that the documents are represented the same way: the terms in the documents are partitioned by the same unary relations, i.e., semantic classes, and the similarity functions defined within these classes are also the same. The comparison of two such representations is meaningful, because they have the same structure defined by the signature Σ.

The presented approach is *similarity of hands*. It is motivated by the card game analogy, and is aimed to explore the document similarity beyond vector-space model and its inadequacies. So, on one hand, its purpose is to identify and name the features and mechanisms shared by the semantic TDT approaches discussed in Section 3.3. On the other hand, it introduces a framework in which to integrate ontologies into document similarity that were only vaguely present in the previous work.

5.2 Term-term correlations

5.2.1 Resnik

In order to determine the semantic similarity of terms in an 'is-a' taxonomy, Resnik [118] proposed the use of *information content* that arises from information theory. The information content of a term a is the negative log-likelihood, $-\log p(a)$. If the term is highly probable (frequent), its information content is low, and vice versa.

The probabilities of terms are simple relative frequencies, $\hat{p}(a) = f(a)/T$, where $f(a)$ is the frequency of the term a in the background corpus and T the number of all term occurrences. When calculating the probabilities, the occurrence of a term is also counted as an occurrence of all terms above it in a taxonomy all the way to the root. For example, in case of a geographical ontology of Figure 5.2 the occurrence of *Paris* is considered also an occurrence of *France* and of *Europe*. This way, the frequency of the root node ('world') equals to T and thus its information content is zero. Furthermore, the probability can only increase (and the information decrease) as one moves up the taxonomy.

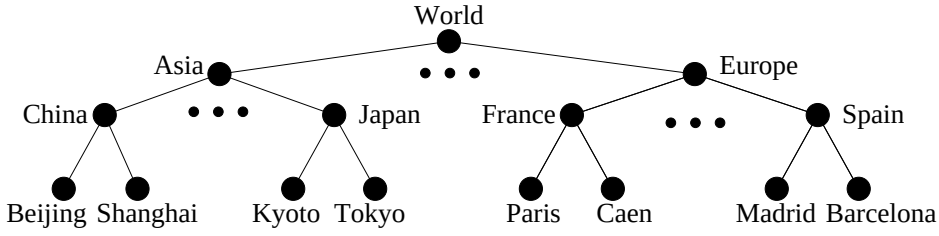

Figure 5.2: A simple geographical taxonomy.

The similarity of two terms in a taxonomy is based on the common node furthest from the root, or equivalently, the common node with the lowest probability and thus the highest information content. Resnik term-term similarity is a function

$$cor^{resnik}(a,b) = \max_{c \in C(a,b)} [-\log p(c)], \tag{5.3}$$

where $C(a,b)$ is the set of terms in the common path of terms a and b.

5.2.2 Cover

Another way to approach geographical similarity is to simply compare the placenames occurring in the documents and find the lowest common ancestor in a taxonomy. Instead of modeling the occurrence of individual terms, as was done in the previous section, we merely look at the *level* of the term in the taxonomy. In Figure 5.2, the root node 'world' corresponds to level 0, the continents to level 1, countries to level 2, and cities to level 3. The *cover* term-term correlation multiplies the level of the lowest common ancestor of terms a and b with a constant ϵ,

$$cor^{cover}(a,b) = \epsilon \max_{c \in C(a,b)} level(c), \tag{5.4}$$

where $level : \mathcal{T} \rightarrow \mathbb{N}$ returns the taxonomy level of term $t \in \mathcal{T}$, and $C(a,b)$ is the set of terms in the common path of terms a and b. If the terms a and b are on different continents, their lowest common ancestor is on level 0 and thus the correlation yields 0. The values of ϵ are discussed in Section 5.5.3.

5.3 Similarity coefficients

Cosine coefficient is one way to determine and normalize the intersection of two semantic vectors, but there are alternatives.

Overlap, for instance, sums the smaller of each pair of elements as follows [122]:

$$sim_i^{overlap}(\mathbf{d}_i, \mathbf{q}_i) = \frac{\sum_{k=1}^{n} \min(d_{ik}, q_{ik})}{\min(\sum_{k=1}^{n} d_k, \sum_{k=1}^{n} q_k)}. \tag{5.5}$$

There are sometimes great differences in document lengths and term frequencies, when comparing a Sunday edition New York Times article and a brief CNN news broadcast, for instance. Accumulating the smaller of the two term-weights for each term reduces the impact of higher term frequency of long documents. A variant of overlap called *asymmetric overlap* [122] normalizes against the length of the document and is interested only in the parts of the query that agree with the document,

$$sim_i^{asym}(\mathbf{d}_i, \mathbf{q}_i) = \frac{\sum_{k=1}^{n} \min(d_{ik}, q_{ik})}{\sum_{k=1}^{n} d_k}. \tag{5.6}$$

The named entity based semantic classes usually have very short term vectors. For these classes, we shall experiment this asymmetric version to measure the extent to which a new document overlaps with a topic centroid.

Coefficient *max-pair* examines all pairs of terms from the term vectors and returns the highest correlation,

$$sim_i^{maxpair}(\mathbf{d}_i, \mathbf{q}_i) = \max_{jk} cor_i(d_{ij}, q_{ik}), \tag{5.7}$$

where $1 \leq j \leq |\mathbf{d}_i|$ and $1 \leq k \leq |\mathbf{q}_i|$. Documents often contain numerous placenames, possibly in several sub-regions or on two or more continents. Max-pair is used for placenames with cover term-term correlation to measure the level in the ontology at which the documents coincide. Max-pair does not make a strong feature by itself, and it is therefore used to devalue the scores of other semantic classes. If similar events are reported to take place in different parts of the world, max-pair is used to decrease or increase the other similarity coefficients depending on (rough) proximity of referred places. In cases where topic centroids do not contain any geographical named entities, max-pair remains agnostic and yields a score of 1.0.

5.4 Score models

Score models transform the class-wise similarities into one real-valued score.

5.4.1 Support-vector machine

As a result of comparing documents, there are two classes of score vectors $\mathbf{x}$: ones that result from comparing document representations discussing the same topic, and those resulting from comparison of documents not discussing the same topic. For convenience, we shall call these types positive vectors and negative vectors, respectively. Assuming there is a difference between these two types of vectors, we can approach the score model transformation as a classification problem.

In the cosine-based baseline system, there was only one similarity score and thus need for only one decision threshold. Ideally, with k semantic classes, we need a k-dimensional threshold, a decision surface $h = (\mathbf{w}, b)$ that separates the positive score vectors from the negative. The weight vector $\mathbf{w} \in \mathbb{R}^k$ is perpendicular to the decision hyperplane, and $b \in \mathbb{R}$ is the distance of the decision hyperplane from the origin. Figure 5.3 illustrates an ideal situation with two semantic classes. The negative vectors are concentrated closer to origin, while positive vectors reside further out, as in the ideal case the similarity coefficient of locations and proper names yield higher scores when documents discuss the same topic.

For finding a separating hyperplane we adopt a robust machine learning method called *support-vector machine* (SVM, see, e.g., [43]). The method is given labeled positive and negative vectors as training data, $X = ((\mathbf{x_1}, y_1), (\mathbf{x_2}, y_2), \ldots, (\mathbf{x_n}, y_n))$, where a labeled training data point $(\mathbf{x}, y)$ is composed of a score vector $\mathbf{x}$ and its label y (-1 for negative and +1 for positive points). SVM produces a decision surface h that separates

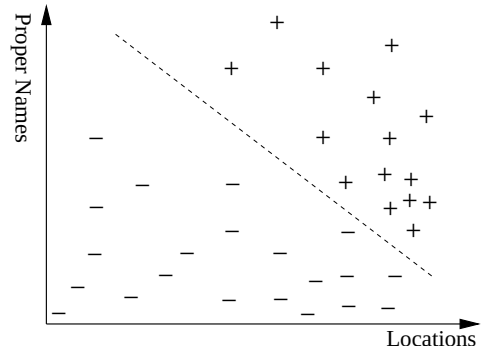

Figure 5.3: The space of vectors $\mathbf{v}$. Plus denotes a positive score vector, and minus a negative. The dashed line represents a hyperplane separating the two types of vectors.

the two types of vectors with maximal margin, i.e., is furthest away from both closest positives and closest negatives.

The classification function $f(\mathbf{x})$ takes the sign of the distance of point $\mathbf{x}$ from decision hyperplane h. It is simply the inner-product of the plane normal $\mathbf{w}$ and the vector $\mathbf{x}$ given the distance b,

$$f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b. \tag{5.8}$$

The margin γ_i of point $(\mathbf{x}_i, y_i)$ with respect to $(\mathbf{w}, b)$ is $\gamma_i = y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b)$. The data point $\mathbf{x}_i$ is classified correctly, when the margin is positive, i.e, $\gamma_i > 0$. The margin of the hyperplane h with respect to data set X is the distribution of margins of the data points. The greater this margin is, the fewer errors h yields.

Finding the optimal hyperplane h^* equals to minimizing the *structural risk*, i.e., finding the hyperplane that provides the lowest number of errors on the unseen data. Support vector machines are based on the hypothesis that the structural risk is minimized when there is maximum margin γ between the two sets of data points. The idea of hyperplane h^* and the margin is depicted in Figure 5.4. While the thin dotted line in the figure would separate the data points correctly, the thick solid line does it with maximal margin and is thus less likely to make errors on unseen data. We can think of the weight vector $\mathbf{w}$ as a linear combination of the training examples and rewrite the hyperplane in its dual form:

$$f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b = \langle \left(\sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i \right), \mathbf{x} \rangle + b \tag{5.9}$$

$$= \sum_{i=1}^{n} \alpha_i y_i \langle \mathbf{x}_i, \mathbf{x} \rangle + b, \tag{5.10}$$

where α_i is proportional to the number of errors done on x_i. Now, finding the optimal hyperplane h^* boils down to optimizing α, which can be done with quadratic programming.

Thus far we have assumed that the classification problem is linearly separable, i.e., there exists a linear hyperplane that correctly classifies all data points. In reality, this assumption is largely unwarranted, but soft-margin support-vector machines build a classifier that divides the hyperspace as cleanly as possible. Furthermore, by replacing the inner-product in Equation 5.10 with some other similar function, a *kernel*, the data points can be mapped to a higher-dimensional space where a better solution may be found. Now the classification function can be written as

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b, \tag{5.11}$$

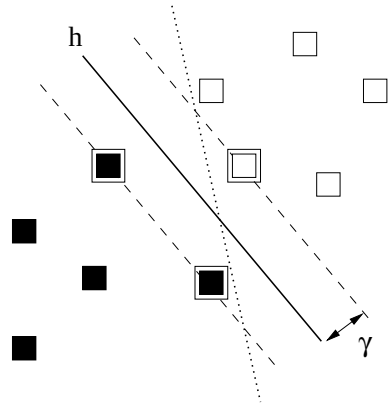

Figure 5.4: An example of a decision surface h and its margin γ. The data points on the dashed margins represent support-vectors.

where K is a mapping $K(\mathbf{x}, \mathbf{z}) = \langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle$, where ϕ maps the score vectors to a new feature space.

We are using three different kernels. The simplest is the linear kernel, which is straight-forward inner-product

$$K_{lin}(\mathbf{x}, \mathbf{z}) = \langle \mathbf{x}, \mathbf{z} \rangle, \tag{5.12}$$

where $\mathbf{x}$ and $\mathbf{z}$ are both score vectors. The radial basis function (RBF) can be used in a non-linear kernel

$$K_{rbf}(\mathbf{x}, \mathbf{z}) = \exp(-\beta ||\mathbf{x} - \mathbf{z}||^2), \tag{5.13}$$

where β is a regulatory width parameter.

The third kernel is called *all-subsets kernel* [131]. It extends the original set of k semantic classes with a new semantic class ϕ_A for every subset $A \subseteq \{1, 2, \ldots, k\}$. The function ϕ_A simply multiplies the original class-wise similarity scores,

$$\phi_A(\mathbf{x}) = \Pi_{i \in A} x_i. \tag{5.14}$$

For computation, the kernel does not need to produce the subsets explicitly.

$$\begin{aligned} K_{all}(\mathbf{x}, \mathbf{z}) &= \langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle & (5.15) \\ &= \sum_{A \subseteq \{1,\ldots,k\}} \prod_{i \in A} x_i z_i & (5.16) \\ &= \prod_{i=1}^{n} (1 + x_i z_i). & (5.17) \end{aligned}$$

The motivation for all-subsets kernel is in combining semantic classes into new semantic classes. For instance, in addition to having LOCATIONS and PERSONS, it might be beneficial to combine them into LOCATIONS-PERSONS feature, that would have a high value when ever both LOCATIONS and PERSONS have a high value. Similar combining and augmenting of features has been applied in TDT (see, e.g., [105]), but previous approaches have not made use of a kernel such as this.

To summarize, when comparing two Σ-structures A and B, we obtain the score vector $\mathbf{x}$ from class-wise comparisons of semantic classes, $\mathbf{x} = (sim_i(A_i, B_i))_{i=1}^n$, where sim_i is the similarity coefficient of semantic class i, A_i and B_i are the semantic vectors of class i in A and B, respectively, and n is the number of semantic classes in the signature Σ. To transform the score vector $\mathbf{x}$ into a real value, we employ a score model f, $f(\mathbf{x}) \in \mathbb{R}$. The decision whether A and B discuss the same topic or not, is based on this outcome of $f(\mathbf{x})$. If it is greater than 0, A and B are deemed on-topic; otherwise, off-topic.

5.4.2 Eliminative score models

As a machine learning problem, modeling the positive score vectors is frustrated by the small number of positive examples in the midst abundance of negative examples. For instance, in TDT-2 training data, there are 6,594 positive pair-wise comparisons ($N_t = 1$) and 1,927,576 negative ones, respectively. This imbalance affects the models in that the recall remains somewhat poor.

On the other hand, the baseline vector-space model solves some 80-90% of the tracking problem. As we saw in Section 4.3.3, many of the remaining errors had to do with events being similar yet not the same, and thus there is considerable overlap in vocabulary. In what we call an *eliminative score model*, the system relies on basic vector-space model up to a threshold, and for scores above that, it passes the decision to the semantic models. This "piggybacking" confines the examples to a range where positives are not overwhelmed. In addition, it runs semantic modeling on pairs of documents that have some terms in common. The classification function $f(\mathbf{x})$ of Equation 5.11 is re-defined as follows:

$$f_{elim}(\mathbf{x}) = \begin{cases} -\infty, \text{if } x_{bvsm} \leq \theta_{elim}, \\ \sum_{i=1}^n \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b, \text{if } x_{bvsm} > \theta_{elim}, \end{cases} \tag{5.18}$$

where θ_{elim} is a threshold for using the semantic classes, x_{bvsm} is the similarity coefficient of a basic vector space model.

We obtain elimination thresholds θ_{elim} from trials with the training data. First, pair-wise document comparisons in the training data are run to produce pairs of score vector and corresponding label $(\mathbf{x}, y)$. Then, elimination threshold θ_{elim} is iteratively increased from 0 upwards. At each iteration, two random samples are drawn from the training output: first 30 000 vectors for training the model and another 30 000 vectors for evaluating it. The best performing models are used for topic tracking or first-story detection.

On the whole, a similar strategy is used in question answering systems. A set of candidate documents is first retrieved with a basic retrieval approach, and then the candidates are more elaborately processed.

5.5 Experiments

5.5.1 Ontologies

Our ontology consists of two parts: a *gazetteer* for names of people, organizations, and geographical features, and a taxonomy for the geographical features. Neither is fitted to the text corpora, i.e., the ontology contains many names that never occur in the corpora. The gazetteer consists of a manually compiled list of 9,700 companies and organizations with abbreviations and name variants. Many entities, like governmental organizations and politicians, have multiple names. Thus, 'UN' and 'United Nations' refer to the same named entity. In addition, the gazetteer contains a list of about 40,000 politicians, athletes, musicians, actors and authors collected semi-manually mostly from English Wikipedia [2]. In addition, the ontology contains common English first and last names and adjectives for nationality.

Furthermore, the gazetteer comprises about 280,000 geographical names. It is compiled from the GeoNames geographical database [3] by selecting populated places with more than 500 inhabitants, regions, rivers, and lakes. Table 5.1 shows the number of different feature types in the gazetteer. Many of the features are referred to with multiple names. For instance, 'Eire', 'Republic of Ireland', and 'Ireland' all refer to the same geographical entity, the country of Ireland. Most of these names come from GeoNames, but we have done some manual work with synonyms and regions.

We have a five-level taxonomy for geographical features. The meaning of the levels depends on the type of the feature. As to land, cities reside on the bottom level, i.e., level 5; above them there are administrative districts

[2] http://en.wikipedia.org/
[3] http://www.geonames.org/

(level 4), countries (level 3), regions (level 2)and finally continents (level 1). Mountains, lakes, glaciers, valleys, and rivers are located in countries ignoring administrative regions and thus they are placed on level 4. Danube, for instance, is placed under several countries as it flows through across half of Europe. The maritime features are organized only to three levels that roughly correspond to oceans (level 1), seas (level 2) and bays (level 3).

In our geographical taxonomy, Balkan is partly eastern, partly southern Europe. Thus, features in Balkan countries have two paths in the taxonomy: one through Balkan and another through eastern or southern Europe. The situation is similar with other region-level entities like Levanto, Maghreb, and African Horn, for instance.

5.5.2 Preprocessing

Preprocessing the documents for the use of semantic classes follows the general outline in Section 3.1.1. Step 1 is expanded with named entity recognition and normalization, and step 5 builds a term vector for each of the semantic classes instead of just one document vector. In the following, we present the details of named entity recognition and normalization.

Spotting missing capital letters

A considerable portion of TDT-2 and TDT-3 corpora consists of automatically transcribed speech. Although some of the punctuation is in place, many of the documents lack some or all capital letters. Since the uppercase characters are important in recognizing the named entities, the text

Table 5.1: The types of features in the gazetteer. Many of the features have multiple names.

Populated places		Other	
type	*#features*	*type*	*#features*
continents	7	deserts	20
regions	22	glaciers	6,072
countries	232	isles or islands	35,171
adm. districts	3,970	lakes	52,287
cities	101,203	mtns. and mtn. ranges	34,177
		oceans/seas	96
		rivers	32,252
		valleys, canyons and plains	13,924

needs to be rewritten with capital letters in their proper place.

In order to achieve this, we employ a simple heuristic process. The TIPSTER corpus is composed of newswire material with capital letters in place. From that data we calculate probabilities for a token t starting with an upper-case letter ($P(first|t)$), being all upper-case ($P(all|t)$), and being all lowercase ($P(none|t) = 1 - P(first|t) - P(all|t)$). If the likelihood ratios of either $\frac{P(first|t)}{P(none|t)}$ or $\frac{P(all|t)}{P(none|t)}$ exceeds a given threshold, the token is written with starting upper-case or all upper-case; preference on the latter alternative. If the threshold is not exceeded, the token remains lower-case.

There are tokens like 'river' that occur in upper and lower case somewhat equal number of times, but when succeeding the token 'Potomac', it has a capital letter for certain. To capture these multi-word names, we use a window of two and three consecutive words to check up against the ontology described in the previous chapter. In case there is a match, the tokens are re-written in upper-case.

We evaluated this approach with 4,166 documents of Associated Press data from TDT-2 corpus. It is newswire data, so we assumed its capital letters to be correct. The text was lower cased, processed and the result was compared to the original token-by-token. The results are listed in Table 5.2.

Table 5.2: The performance of spotting the missing capital letters with 4,166 documents of Associated Press newswire data.

	upper-case first	*upper-case all*
tokens	2,551,774	2,551,774
targets	394,381	18,572
precision	0.942	0.841
recall	0.800	0.793

For proper names and words starting a sentence the precision of upper-casing was fairly high, 94%. Considering the large number of targets, i.e., true positives, this is a fairly good result. As to recall, there seems to be place for improvement. In recognizing all upper-case abbreviations, like 'NATO' and 'UNESCO', the system seems to have fared less well. It appears that many of the misses were a result of the original text being all upper-case. Some of the false alarms are due to lowercase abbreviations in the Associated Press material.

On the transcribed audio data, lack of punctuation is likely to weaken the accuracy. This evaluation gives us a rough estimate of the performance, and although there is room for improvement, the performance is satisfactory

for the time being. We will leave adjustments and further development to later work, as we now proceed to named entity recognition.

Named entity recognition (NER)

For the purpose of spotting named entities, we have adopted Stanford Named Entity Recognizer [4] that employs Conditional Random Field (CRF) sequence models for extracting organizations, persons and placenames from text [53]. The reported F_1-score, i.e., the harmonic mean of precision and recall (Equation 2.1), is around 0.85, which is quite good. However, given the sensitivity of TDT tasks and the need for disambiguation, we have built a simple post-prosessing for the recognizer output.

We evaluated the results of named entity recognition for 196 documents of native English news documents from TDT-2 training corpus. The translation errors in the Chinese data hinder the evaluation, because named entities are often garbled into strings of incoherent words, and so determining misses or false-alarms becomes very difficult. Furthermore, using regular expressions, we removed the phrases like "I'm Anthony Keith James, and this is CNN Headline Sports" and " You're listening to 'The World' on PRI, Public Radio International" as they are frequent in documents of some sources and contain irrelevant named entities. The results for Stanford Recognizer are shown in Table 5.3. The table lists token counts, not counts for named entities. The recognizer labels each token separately, i.e., 'New York City' is composed of three tokens and is reported as three placename tokens in the results.

Table 5.3: The results of the Stanford Named Entity Recognizer on 196 native English news stories of TDT-2. There were 84,934 tokens in total.

	placename	*organization*	*person*
entity tokens	1,789	1,806	2,922
precision	0.807	0.876	0.965
recall	0.926	0.838	0.882
F_1	0.863	0.856	0.922

Some of the errors were due to false capital lettering. Adjectives denoting nationality tended trigger false-alarms. Overall, the named entity recognition F_1-score is in line with the previous results with Stanford Recognizer.

[4] http://nlp.stanford.edu/software/CRF-NER.shtml

Named entity normalization

Recognizing named entity tokens is only a half of the problem. We need to convert the strings of tokens into names and to find possibly varying references to the same named entity. This normalization process verifies the recognition output with the ontology, and ensures the same named entity has the same label in all occurrences in a document. In addition, it normalizes named entities in a document, i.e., the initial occurrence of 'Steve Fossett' and subsequent occurrences of 'Fossett', for instance, all are considered to refer to the same named entity.

In normalizing placenames, organizations, and persons we employed the ontology described in Section 5.5.1. A string of tokens was considered a placename only if it was found in the gazetteer. Thus, locations smaller than a city were not included. Person and organization tokens were matched against the ontology for name alternatives, e.g., 'IBM' and 'International Business Machines' refer to the same company, just like 'Newt Gingrich' and 'Newton Leroy Gingrich' refer to the same person. The ontology was not considered exhaustive in terms of persons and organizations, and so tokens not present in the ontology were not discarded like locations. We did not, however, include mere first-names. Whatever tokens were excluded from classes of named entities, were considered normal terms.

Table 5.4 presents the results of the normalization phase. Since many named entities consist of multiple tokens, there are fewer named entities than there were recognized tokens. Despite the use of ontology with which to verify the labels, the effectiveness of the normalization turned out to be worse than that of recognition except for placenames.

Table 5.4: The results of named entity normalization on 196 native English news stories of TDT-2.

	placename	*organization*	*person*
named entities	1,558	1,106	2,172
precision	0.925	0.869	0.917
recall	0.899	0.716	0.879
F_1	0.912	0.785	0.898

The named entity normalization of organizations suffered from NER errors: 55% of misses and 33% of false alarms were due to incorrectly recognized label. About 2% of misses can be attributed to spelling mistakes. The rest is just failure to match multiple occurrences of the same entity.

Person names were similarly plagued by recognition faults: 41% of misses and 28% of false-alarms were a result of NER errors. About 38% of the misses were a result of having mere first-name. Typically, these problems arose in sentences like: "Korda said his wife, Regina, and daughter, Jessica, were the reasons he was inspired to ...". The tennis-player, Petr Korda, is referred to by his last-name and his wife and daughter by their first-names. The system fails to normalize the latter two. Furthermore, in direct quotes people were referred to by first-names. In transcribed audio the reporters address each other by first-names. In addition, nicknames were the cause of failure in a few cases: 'James' becomes 'Jim', 'Robert' becomes 'Bob', and 'Kenneth' becomes 'Ken', for instance.

The normalization of location entities fared the best. The effectiveness in fact improved from the recognition. Still, some 30% of misses and 15% of false alarms were due to errors of the Stanford recognizer. The choice of granularity, i.e., omitting the locations not present in the gazetteer, was the cause in 37% of the misses. Many false-alarms were due to splitting names of organizations like 'U.S. Department of Defense' or 'U.S. Treasury' into a placename and an organization.

The task proved to be somewhat more difficult than anticipated. Overall, the results are good enough for now, but there are several things to improve in future work.

Locations

Placenames sometimes denote the 'where', but often denote an agent, the 'who' as in an excerpt "England inflicted Germany's first defeat in Berlin for 35 years thanks to John Terry's header six minutes from time" [5]. The placenames 'England' and 'Germany' refer to national football teams, and only 'Berlin' refers to an actual location, where something happens. To recognize this difference, we consider two kinds of geographical named entities: placenames, such as all three in the above example, and *locations* that explicitly bind the event expressed by the sentence to some specific place, such as 'Berlin' in the above excerpt.

To recognize locations, we used the Connexor [6] functional dependency parser output. For each recognized placename, we processed the functional dependency tree like the one illustrated in Figure 5.5: if the placename had a source, goal, path or location dependency link to the dependency parse tree or if it had a modifier indicating proximity or direction from, e.g., 'near', 'south of', and 'outside', it was considered a location. In the

[5] Guardian, November 20 2008

[6] http://www.connexor.com

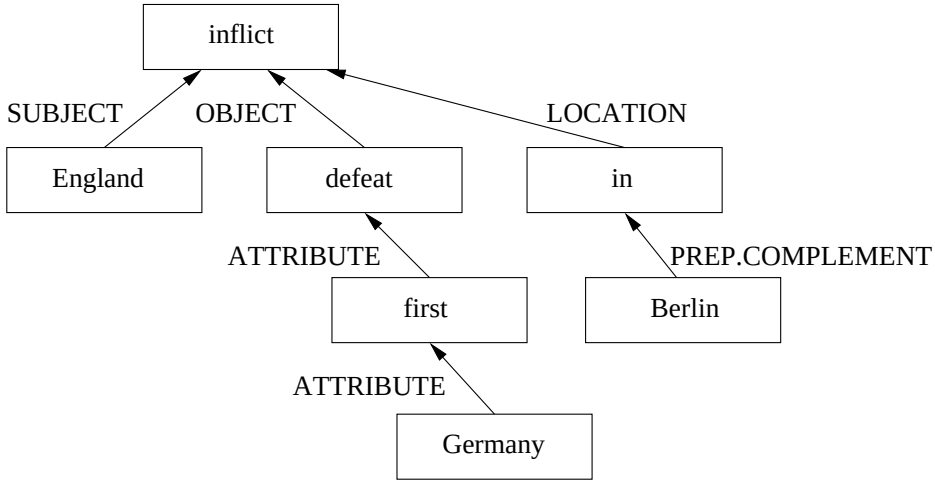

Figure 5.5: An example of simple dependency tree for sentence "England inflicted Germany's first defeat in Berlin". The arrows represent dependency links between nuclei that correspond to Tesnière's basic syntactic elements [68]. In the figure, the nuclei are simply word-forms. Connexor parser provides syntactic information that has been omitted here.

example, only Berlin qualifies as a location, because England is the subject of the sentence and Germany is merely an attribute to 'defeat'.

Named entity disambiguation

In our ontology, there are 18 cities or towns named Kingston. In the world, there are probably more. In order to make use of the geographical ontology, we need to disambiguate the placenames occurring in the news and resolve which Kingston, for instance, is in question. Each geographical entity is defined by a path in the geographical ontology. Thus, in the ontology, there can be only one entity called Kingston in Tasmania and only Kingston in Ontario, Canada. The disambiguation is carried out by first examining the placenames that match to only one entity in the ontology. If there are references to Toronto, we make a note of implicit references to North America, Canada, and Ontario, i.e., of entities that reside on the path of the Toronto. If the unambiguous placenames in a news story center in Ontario, Canada, we then the assume the ambiguous placenames are likelier to refer to entities Ontario than, say, the Caribbean or Australia.

So, given a news story, its placenames are organized into two groups:

the unambiguous placenames (U) that match to only one entity in the ontology - these are typically provinces, countries, subregions and continents - and the *ambiguous* placenames (A) that match to more than one entity in the ontology. The disambiguation process is an attempt to associate each ambiguous placename with the correct geographical entity.

The disambiguation procedure is described in Algorithm 3. First, the entities in the paths of the unambiguous placenames are considered resolved and their occurrences are stored in F (rows 2-4). Thus, an occurrence of placename Toronto would count as an occurrence of North America and Ontario as well.

For each ambiguous placename $a \in A$, the algorithm goes through the matching entities C in the ontology. Each entity $c \in C$ assigned a vector $\mathbf{x}$ consisting of the frequencies found in F for the entities in the path of entity c (rows 9-13). Then the entity c is assigned a weighted sum of a weight-vector of found frequencies and α that represents the relative importance of the ontology-levels in $\mathbf{x}$ (row 14). The weight vector α can be trained with a perceptron, for instance. Finally, the entity c and its score are stored in the results R (row 15). Ultimately, results R contain scores for each entity c for each ambiguous placename a.

Due to low initial precision, we augmented $\mathbf{x}$ with the geographical type and the population of candidate c, if it was available. Thus, populated places were awarded an advantage over geographical features like mountain peaks, lakes, and rivers. More importantly, larger cities gained more emphasis. Practically all European capitals have namesakes in the United States, and as much of the news focus on the U.S., this helps to recognize the scattered references to European towns.

We evaluated the disambiguation with 1,056 documents. The named entity recognition spotted 7,371 instances of placenames, 4,597 of which were unambiguous. In resolving the remaining 2,774 ambiguous placenames, the algorithm failed on 545 instances yielding an accuracy of 80.3% for the disambiguation. It seems many of the failures were a result of confusing a city and a province by the same name. For instance, the cities of Seoul, New York, and Hong Kong reside within province with the same name. Our approach seems to incorrectly score the provinces higher. Although it is not a devastating error in terms of the distance in the geographical taxonomy, the province and the city are two different entities that yield a zero term-term correlation if an ontology-based correlation is not used.

Algorithm 3 A disambiguation algorithm for placenames occurring in a news story. The input consists of two sets of placenames: the unambiguous (U) that have only one interpretation in the ontology, and ambiguous (A) that can refer to more than entities in the ontology. In addition, the algorithm makes use of geographical ontology. The result R stores a score for each entity matching an ambiguous placename $a \in A$.

```
1: F ⇐ {}
2: for all u ∈ U do
3:    add to F the number of occurrences of all the entities in path of
      placename u
4: end for
5: R ⇐ {}
6: for all a ∈ A do
7:    C ⇐ retrieve entities matching ambiguous placename a from the
      ontology
8:    for all c ∈ C do
9:       x ⇐ 0
10:      P ⇐ the entities in the path of entity c
11:      for i = 1 to |P| do
12:         x_i ⇐ F_{P[i]}
13:      end for
14:      R_a^c ⇐ Σ_i α_i x_i
15:      R ⇐ R ∪ R_a^c
16:   end for
17: end for
```

5.5.3 Topic tracking

A dry-run

In order to see how named entities behave in targets and non-targets, we have done a post-hoc analysis on the baseline tracking results shown in Table 5.5. We ran cosine InQuery tracking system on TDT-2 training data and recorded the individual comparisons between topic centroids (both $N_t = 1$ and $N_t = 4$) and documents. Although the system was baseline, the document representations were augmented with semantic classes of ORGANIZATIONS, PLACENAMES, PERSONS, and LOCATIONS, which were recorded but not interfered with the tracking process. Table 5.5 lists the percentage of comparisons having at least one named entity in common among targets (hits and misses) and non-targets that were considered targets (false-alarms). The correct negatives, that is, non-targets that were

Table 5.5: Ratios of hits/misses (target) and false-alarms (non-target) having at least one named entity in common in given classes in two baseline runs $N_t = 1$ and $N_t = 4$. The figures are obtained from post-hoc analysis of a baseline topic tracking results.

	$N_t = 1$		$N_t = 4$	
	hit or miss	*false-alarm*	*hit or miss*	*false-alarm*
ORGANIZATIONS	0.381	0.381	0.534	0.458
PLACENAMES	0.594	0.337	0.654	0.326
PERSONS	0.431	0.181	0.572	0.214
LOCATIONS	0.304	0.133	0.355	0.142

correctly recognized as such, were left out, because given their large quantity, the number of common named entities was close to zero.

From the table we see that common named entities tend to be more frequent among hits and misses than among false-alarms. Naturally, as targets discuss the same topic, this is what one would expect: the occurrence of the same terms. The interesting thing is that with $N_t = 1$ only 30-40% of the targets share any organizations, people, or locations. Shared references to places are more frequent with about 60% of the targets having at least one placename in common with the topic centroid. This certainly limits the potential benefits from named entity based semantic classes. Furthermore, about 30-40% of false-alarms share at least one organization and one placename. Emphasizing the role of named entities in tracking will also give a boost to non-targets.

The situation is somewhat more consolatory, when the topics are defined with four sample stories. The percentage of targets sharing a named entity is considerably increased, while the percentage for non-targets is increased only moderately. Apparently, one sample story is unable to cope with the narrative progression we discussed in Section 4.1.2. The news do repeat the same story over and over, but portrays development of events.

In the above results, locations were compared with binary term-term correlation. To gauge the potential of the ontology-based location comparison, we executed the same baseline run as above and sought the best level where the location terms of the topic centroid and documents matched in the ontology. If they had matching cities, the best level would be city. If one had a city and the other the country where the city is located, the best level would be country. The results listed in Table 5.6.

For $N_t = 1$, the percentage of city level is roughly equal among hits and

Table 5.6: Ratios of best match levels in location terms in two baseline runs $N_t = 1$ and $N_t = 4$. The ratios are obtained from post-hoc analysis of a baseline topic tracking results.

	$N_t = 1$		$N_t = 4$	
	hit or miss	*false-alarm*	*hit or miss*	*false-alarm*
city	0.109	0.111	0.257	0.159
adm. district	0.165	0.179	0.320	0.226
country	0.318	0.434	0.600	0.488
region	0.364	0.474	0.601	0.517
continent	0.439	0.518	0.681	0.551
none	1.000	1.000	1.000	1.000

misses versus false-alarms. Then, as the best level becomes less specific, i.e., denotes larger geographical area, the false-alarms show consistently greater percentage at each level than hits and misses. While about 50% of false-alarms have at least a continent level similarity in location terms, i.e., the documents refer to locations on the same continent, only 44% of targets do same. This goes to show, the ontology-based location similarity does not work well with $N_t = 1$.

Again, the situation is improved when topics have more sample stories. The best matching levels for $N_t = 4$ are consistently higher for hits and misses than for false-alarms.

To summarize our findings, the use of named entity based semantic classes are not likely to eliminate the tracking errors completely. On the average in each topic, only about a half of the on-topic stories have matching named entities in the classes of PERSONS, ORGANIZATIONS, and PLACENAMES. Whatever improvement there will be from using these semantic classes, its impact is going to be dampened by the absence of common terms in on-topic documents as well as the presence of common terms in off-topic documents resulting in false-alarms.

System

The tracking system employing semantic classes is similar to the baseline system. The system processes Σ-structures instead of simple document vectors, and each semantic class has a dedicated inverted index for posting lists and a dedicated lexicon with semantic class specific document frequencies.

The tracking process is basically same as the baseline system in Section 4.3.1 that was based on the cosine similarity in Algorithm 1, but there

are three important differences. First, the accumulator A stores score vectors $\mathbf{x}$ instead of scores. Second, the retrieval phase (rows 2-12) is carried out for each semantic class i separately. And three, the scoring (row 11) can utilize whatever similarity coefficient has been associated with the semantic class. After the retrieval phase, the score vectors $\mathbf{x}$ in accumulator A are transformed with the score model f into decisions: if $f(\mathbf{x}) \geq 0$, the decision is YES, otherwise NO.

The score models are trained with Joachim's SVM-light [71], which is a robust support vector machine implementation. For each model, the training data comprised 30,000 labeled samples drawn from comparison data that was obtained from the training sets of each corpora. SVM-light has a parameter for the weighting errors on positive samples versus negative samples. We used three values which are encoded as low, medium, and high. When the cost-factor is set to low, misclassifications on positives and negatives are of equal value. When it is set to medium (high), misclassifying a positive - which is a miss in our terminology - costs twice (thrice) the misclassification of a negative, a false-alarm. By choosing a higher cost-factor, we obtain a higher recall in the classification at the cost of precision. In many models, this is a way to prevent the positive samples from drowning in the midst of negative samples.

Σ-signatures

We employed six different semantic classes. The class ALLTERMS was the baseline term vector comprising all terms without named entity recognition. Thus, 'New York City' would be three distinct terms 'new', 'york', and 'city' like in baseline system. The class *terms* comprised normal terms, i.e., verbs, nouns, and adjectives. In effect, the class consisted of terms of the above class except for named entities. The classes PERSON, ORGANIZATION, PLACENAME, and LOCATION were obtained as specified in Section 5.5.2.

The experimental setup is made up of a considerable number of variables. To alleviate the lengthy notation, we shall adopt a short hand

$$\Sigma_{all}^{cos} = \{allterms, cor_{all}^{id}, sim_{all}^{cos}\}$$

to refer to the semantic class of ALLTERMS with identity term-term correlation and cosine similarity coefficient. In similar vein,

$$\Sigma_{loc}^{resnik} = \{locations, cor_{loc}^{resnik}, sim_{loc}^{cos}\}$$

refers to the semantic class of LOCATIONS with Resnik term-term correlation and cosine similarity coefficient. These classes are then combined into Σ-signatures. We have ran considerable number of experiments, but here we

report the following. The basic run uses cosine similarity for all six semantic classes and an eliminative score model

$$\Sigma_{basic} = \Sigma_{all}^{cos} \cup \Sigma_{ter}^{cos} \cup \Sigma_{org}^{cos} \cup \Sigma_{pla}^{cos} \cup \Sigma_{per}^{cos} \cup \Sigma_{loc}^{cos} \cup \{f\}.$$

A signature using asymmetric overlap for named entities is specified as

$$\Sigma_{asym} = \Sigma_{all}^{cos} \cup \Sigma_{ter}^{cos} \cup \Sigma_{org}^{asym} \cup \Sigma_{pla}^{asym} \cup \Sigma_{per}^{asym} \cup \Sigma_{loc}^{asym} \cup \{f\},$$

and a signature with Resnik term-term correlation on *locations* is specified as

$$\Sigma_{resnik} = \Sigma_{all}^{cos} \cup \Sigma_{ter}^{cos} \cup \Sigma_{org}^{asym} \cup \Sigma_{pla}^{resnik} \cup \Sigma_{per}^{asym} \cup \Sigma_{loc}^{resnik} \cup \{f\}.$$

Finally, a signature using max-pair similarity coefficient for locations and asymmetric overlap for the other named entities is

$$\Sigma_{maxpair} = \Sigma_{all}^{cos} \cup \Sigma_{ter}^{cos} \cup \Sigma_{org}^{asym} \cup \Sigma_{pla}^{asym} \cup \Sigma_{per}^{asym} \cup \Sigma_{loc}^{maxpair} \cup \{f\}.$$

Max-pair uses cover term-term correlation of Equation 5.4, and to modify the path lengths, we use $\epsilon = 0.3$ such that matching PLACENAMES of cities (level 5) yield a correlation of 1.5, matching countries (level 3) 0.9 and matching continents (level 1) 0.3.

Results

We ran semantic topic tracking system with TDT-2, TDT-3, and TDT-5 corpora. The TDT-3 runs were trained with the whole of TDT-2. TDT-5 was trained with newswire portion of TDT-3, because it contains no broadcast news sources. In order to decrease the execution time and to remedy the problems from documents of varying length, the document and topic representations used 100 terms for ALLTERMS and TERMS vectors regardless of the actual length of the document. The other classes remained unrestricted. The terms were chosen simply on the basis of term-weight for each document separately. Thus, this did not reduce the term space.

The error probabilities and detection costs (see Section 2.2.3) for semantic topic tracking are listed in Table 5.7. The table shows only the best run for each signature Σ given the corpus and the number of samples. With the smaller corpora, we simply explored system performance by testing systematically different kernels, cost-factors, and elimination thresholds θ_{elim}. With TDT-5, we experimented only with a few of the most promising settings.

Table 5.7: The semantic topic tracking results. The column 'kernel' specifies SVM kernel used ('all' stands for all-subsets kernel of Equation 5.17, 'rbf' for radial basis function of Equation 5.13) and the corresponding cost-factor ('l' for low, 'm' for medium, 'h' for high). The column θ_{elim} refers to threshold for eliminative score models. If it is non-zero, an eliminative score model was used; a plain SVM otherwise. The column p gives the p-value by Student's t test as compared to the corresponding best baseline value in Table 4.5.

				Topic-Weighted at θ			Minimum Topic-Weighted			
signature	*kernel*	θ_{elim}	N_t	$p(\overline{r}\|\omega)$	$p(r\|\overline{\omega})$	$(C_{det})_{norm}$	$p(\overline{r}\|\omega)$	$p(r\|\overline{\omega})$	$(C_{det})_{norm}$	p
TDT-2										
Σ_{asym}	all-l	0.08	1	0.0839	0.0116	0.1405	0.0726	0.0081	0.1123	
Σ_{basic}	all-l	0.08	1	0.1410	0.0033	0.1571	0.0714	0.0084	0.1128	
$\Sigma_{maxpair}$	all-m	0.08	1	0.1099	0.0058	**0.1382**	0.0618	0.0109	0.1154	
Σ_{resnik}	all-l	0.14	1	0.0412	0.0261	0.1690	0.0693	0.0087	**0.1111**	0.049
Σ_{asym}	all-l	0.02	4	0.0732	0.0154	0.1486	0.0825	0.0098	0.1306	
Σ_{basic}	rbf-l	0.04	4	0.1295	0.0054	0.1558	0.0829	0.0084	**0.1261**	0.059
$\Sigma_{maxpair}$	all-l	0.10	4	0.1065	0.0076	**0.1436**	0.0912	0.0077	0.1287	
Σ_{resnik}	all-m	0.08	4	0.1054	0.0091	0.1525	0.0895	0.0094	0.1378	
TDT-3										
Σ_{asym}	all-l	0.16	1	0.1257	0.0104	0.1767	0.0765	0.0150	0.1501	
Σ_{basic}	rbf-m	0.18	1	0.0952	0.0124	**0.1560**	0.0765	0.0150	0.1500	
$\Sigma_{maxpair}$	all-l	0.14	1	0.1070	0.0156	0.1832	0.0878	0.0142	0.1573	
Σ_{resnik}	rbf-l	0.14	1	0.0943	0.0138	0.1620	0.0764	0.0150	**0.1499**	0.363
Σ_{asym}	all-l	0.08	4	0.1377	0.0136	0.2044	0.0828	0.0136	0.1427	
Σ_{basic}	all-l	0.08	4	0.1069	0.0121	0.1660	0.0879	0.0111	0.1422	
$\Sigma_{maxpair}$	all-m	0.08	4	0.1325	0.0070	0.1670	0.0860	0.0109	0.1394	
Σ_{resnik}	all-m	0.08	4	0.1272	0.0050	**0.1519**	0.0868	0.0104	**0.1378**	0.002
TDT-5										
Σ_{asym}	all-l	0.16	1	0.2176	0.0110	0.2775	0.1839	0.0127	0.2524	
Σ_{basic}	all-l	0.16	1	0.2267	0.0021	0.2371	0.0923	0.0092	0.1427	
$\Sigma_{maxpair}$	all-m	0.08	4	0.1829	0.0127	0.2524	0.0936	0.0158	0.1749	
Σ_{resnik}	all-l	0.10	1	0.1176	0.0075	**0.1539**	0.0728	0.0108	**0.1314**	0.014
Σ_{asym}	all-l	0.08	4	0.1981	0.0064	0.2296	0.1344	0.0130	0.1979	
Σ_{basic}	all-l	0.08	4	0.1112	0.0069	**0.1451**	0.0915	0.0085	**0.1332**	0.005
$\Sigma_{maxpair}$	all-m	0.08	4	0.5380	0.0036	0.5554	0.4854	0.0092	0.5309	
Σ_{resnik}	all-m	0.08	4	0.1239	0.0093	0.1694	0.0863	0.0145	0.1575	

There appears to be a substantial improvement in the minimum topic-weighted costs from the baseline results of Table 4.5 across the board. We tested the statistical significance using paired Student's t test [96]. It is simple to implement and still it is in agreement with the more elaborate randomization and bootstrap methods [134]. The null hypothesis is that the minimum detection cost results in Table 5.7 and the corresponding baseline runs in Table 4.5 are random samples from the same normal distribution. The listed results are topic-weighted averages of per topic detection costs, and so comparing the minimum $(C_{det})_{norm}$ costs is effectively comparing means. The null hypothesis assumes the difference of means is zero. For the best results on each corpus, we paired the topic-wise $(C_{det})_{norm}$ costs with those of the baseline run, and calculated the one-tailed p-value. As

a result, the best $N_t = 4$ TDT-2 run and the best $N_t = 1$ TDT-3 run fail to refute the null hypothesis, and are thus not statistically significant improvements to 95% confidence.

When looking at the tracking performance with threshold θ obtained from training run, the costs show improvement less uniformly, in some cases none at all. The decision scores of the baseline systems ranged neatly between 0 and 1, so with normalization the trained threshold was close to the optimal threshold produced by the evaluation software. The decision scores of support-vector machine score models are signed distances from the decision hyperplane that do not have absolute similarity and dissimilarity. It appears that a trained threshold for such distance can be quite far from the optimal threshold, if it is used on another corpus. Moreover, there are differences in which elimination thresholds worked for which corpus. Given one sample story, the Σ_{basic} works best on TDT-2 with all-subsets kernel with $\theta_{elim} = 0.08$, while the threshold is higher for the best Σ_{basic} runs on the other corpora.

All of the listed runs are for eliminative score models. Given the score vectors, it would seem that the plain SVM score models are not able to deal with the imbalance between positive and negative samples very well. Similarly, the most of the listed models employ the 'low' cost-factor indicating that the higher recall from increasing penalties for missing positives in the training phase is delivered with too high a false-alarm rate.

Except for two cases, the most effective runs use all-subsets kernel. Since the kernel expands the score vectors with new features as discussed in Section 5.4.1, it must be concluded that combining the scores of different semantic classes is beneficial.

In TDT-2 and TDT-3, the differences between the minimum scores of the different signatures Σ are small. Overall, Σ_{resnik} seems to triumph with a narrow margin. The runs on TDT-5 show greater differences with Σ_{basic} and Σ_{resnik} as the most effective. The benefit of using a geographical ontology is thus not conclusive.

Figure 5.6 represents the DET curves of TDT-3 semantic topic tracking with four samples, i.e., $N_t = 4$. Here, the geographical ontology turns out to perform poorly in the high recall area as the curves shoots off to 30% false-alarm rate at 2% miss rate. The other systems remain consistently (if narrowly) below the baseline DET curve.

The signature Σ_{resnik} using the geographical ontology is clearly outperformed by the basic run, Σ_{basic}. Its effectiveness degrades very rapidly in the high recall/low miss rate area of the graph, a trend shown by all curves but these two in particular. It may be that the score models that worked

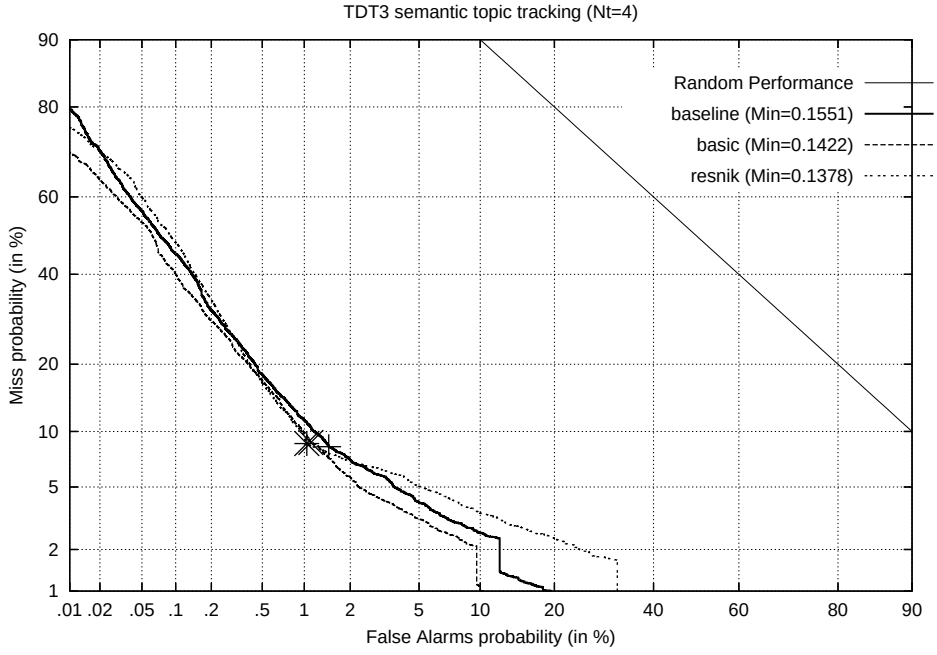

Figure 5.6: The topic-weighted DET curves of TDT-3 semantic topic tracking ($N_t = 4$).

well on TDT-2 and TDT-3, do not work well on TDT-5.

Figure 5.7 represents the DET curves of TDT-5 semantic topic tracking with four samples, i.e., $N_t = 4$. The signatures Σ_{basic} and Σ_{resnik} provide some gain over the baseline run. The use of geographical ontology does not reach miss rates below 2%, so there are documents that are never found similar with any similarity threshold θ. This is probably a problem in the trained score model. The signature Σ_{basic} consistently outperforms the baseline as well as the Σ_{resnik} signature - except for the high miss rate region in the upper left corner of the graph.

5.5.4 First-story detection

System

The first-story detection system with semantic classes uses the same document preprocessing as semantic topic tracking that was described in Section 5.5.2. In addition, the score models with SVMs are trained using the same process as in topic tracking: 30,000 random samples from score vector

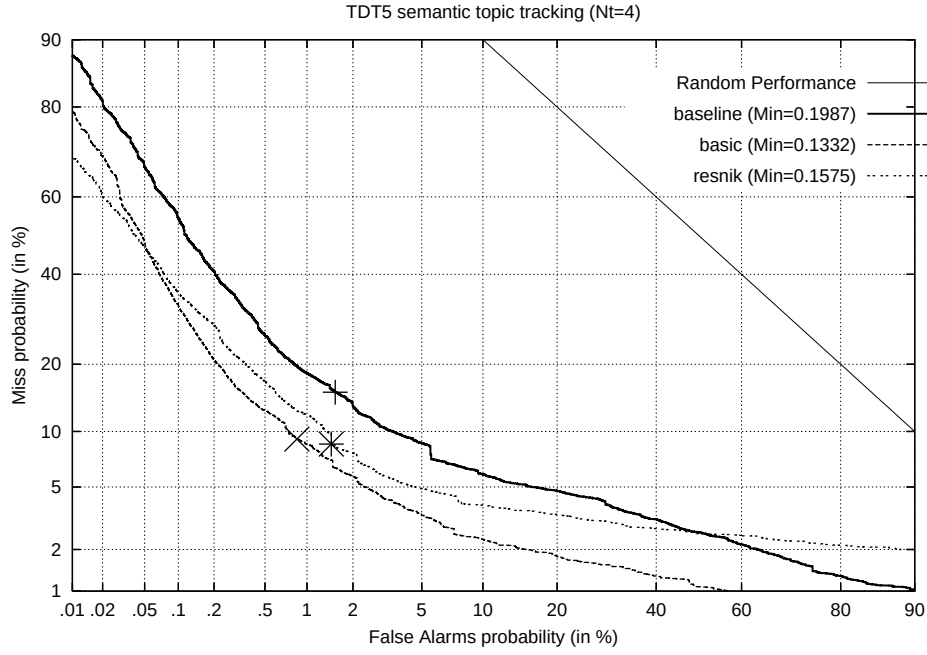

Figure 5.7: The topic-weighted DET curves of TDT-5 semantic topic tracking ($N_t = 4$).

output with training corpus.

Similarly to baseline FSD in Section 4.4.1, the semantic FSD system is still fundamentally a single-pass clustering system. Given a new document, it retrieves candidates from the inverted indices, ranks them and examines the closest match. Nevertheless, the introduction of semantic classes requires some minor modifications in Algorithm 2. The similarity functions on row 4 needs to make use of score model f to transform the score vector $\mathbf{x} = (sim_i(\mathbf{A_i}, \mathbf{B_i}))_{i=1}^n$ into a decision score. Thus, for semantic first-story detection row 4 would read

$$\text{Find } \mathbf{C}_k : f((sim_i(\mathbf{D}_i, \mathbf{C}_{ik}))_{i=1}^n) = \max_{1 \le j \le m} f((sim_i(\mathbf{D}_i, \mathbf{C}_{ij}))_{i=1}^n),$$

$\mathbf{C}_k$ is best match in the previous documents, $\mathbf{D}$ the new document under inspection, $\mathbf{D}_i$ and $\mathbf{C}_i$ the semantic vectors of documents $\mathbf{D}$ and $\mathbf{C}$, sim_i the similarity coefficient of the semantic class i and f the score model.

Results

Again, by convention the first-story detection is run with native English data only [1, 5]. Table 5.8 shows the results of first-story detection using semantic classes. It is a mixed baggage: there is dramatic improvement with TDT-2, less dramatic with TDT-3, and a something of set-back with TDT-5 minimum topic-weighted detection costs. For the former two corpora, we explored a space of parameter values: kernels, cost-factors, and θ_{elim} thresholds. For TDT-5, we ran only the models that did well on the other corpora, and apparently those models did not work as well with larger corpus.

On the other hand, the effectiveness at θ shows lower detection costs compared to the baseline detection costs in Table 4.9. The situation was the opposite with topic tracking in Section 5.5.3. On the whole, the gap between the cost from trained threshold and the cost from optimal threshold is narrower than in topic tracking. Looking at the p-values, only TDT-2 result yields 95% confidence. With p-value less than 0.1, the best TDT-5 result attains 90% confidence. Despite the large corpus, there is a fairly low number of topics to average over, and so a statistically significant improvement is hard to produce.

Overall, Σ_{asym} yields lowest detection costs with all corpora. The similarity coefficient used to compare named entities is not symmetric, but measures only the degree to which the new document matches an earlier document. This seems to work better in FSD than cosine. The non-binary term-term correlations maxpair and resnik are not helpful here.

Figure 5.8 illustrates the DET curves of semantic FSD on TDT-3. The curves overlap to great extend, especially in the mid area, where there are more data points and where the curves are less "choppy". None of the semantic FSD curves stay below the baseline curve suggesting that even the best performing signature Σ_{asym} does not provide consistent improvement. This may be because listed runs are selected by minimum detection costs, not their DET curves.

The situation is similar with TDT-5 curves depicted in Figure 5.9. The points of minimum detection costs reside in the upper part of curve above the 60% miss rate line. Compared to the baseline, the semantic methods are able to decrease the false-alarm rate in the high precision area of the curve. The first-story detection relies on tracking in spotting something new, and tracking is, as discussed in Section 4.3.2, biased towards high recall.

The semantic TDT approach we have presented appears to be geared towards topic tracking. The eliminative score models in topic tracking are

Table 5.8: The semantic first-story detection results. The column 'kernel' specifies the SVM kernel used ('rbf' for radial basis function of Equation 5.13, 'lin' for linear kernel of Equation 5.12) and the corresponding cost-factor ('l' for low). The column θ_{elim} refers to threshold for eliminative score models. The column p gives the p-value by Student's t test as compared to the corresponding best baseline value in Table 4.9.

			Topic-Weighted at θ			Minimum Topic-Weighted			
signature	*kernel*	θ_{elim}	$p(\overline{r}\|\omega)$	$p(r\|\overline{\omega})$	$(C_{det})_{norm}$	$p(\overline{r}\|\omega)$	$p(r\|\overline{\omega})$	$(C_{det})_{norm}$	p
TDT-2									
Σ_{asym}	lin-l	0.12	0.2143	0.0593	**0.4644**	0.3750	0.0123	**0.3570**	0.045
Σ_{basic}	rbf-l	0.20	0.3393	0.0620	0.6063	0.3929	0.0081	0.3589	
$\Sigma_{maxpair}$	rbf-l	0.10	0.2143	0.0934	0.6533	0.3571	0.0180	0.4452	
Σ_{resnik}	rbf-l	0.18	0.3393	0.0630	0.5689	0.3214	0.0235	0.4367	
TDT-3									
Σ_{asym}	rbf-l	0.12	0.4095	0.0373	0.5894	0.4381	0.0221	**0.5462**	0.228
Σ_{basic}	rbf-l	0.18	0.5058	0.0215	0.6044	0.3714	0.0403	0.5647	
$\Sigma_{maxpair}$	rbf-l	0.12	0.4190	0.0437	0.6343	0.4476	0.0232	0.5615	
Σ_{resnik}	rbf-l	0.06	0.3619	0.0467	**0.5833**	0.4381	0.0254	0.5623	
TDT-5									
Σ_{asym}	lin-l	0.06	0.9603	0.0005	0.9752	0.6984	0.0184	**0.8174**	0.068
Σ_{basic}	rbf-l	0.18	0.9127	0.0033	0.9435	0.7698	0.0073	0.8357	
$\Sigma_{maxpair}$	rbf-l	0.10	0.9921	0.0004	0.9930	0.8016	0.0218	0.9251	
Σ_{resnik}	rbf-l	0.18	0.8730	0.0063	**0.9127**	0.8730	0.0051	0.9066	

more likely to increase the precision rather than recall. As we discussed in the context of Hellinger coefficient in Section 4.3.2, topic tracking works better in the high precision area of the decision score distribution. FSD is biased towards high recall, as the test corpora. Therefore the performance gains in tracking have not translated to increases in first-story detection effectiveness.

Yet, the results listed in Table 4.10 show fairly good performance with analogous systems. Kumaran and Allan, for instance, used SVM to combine similarities from various kinds of semantic classes (e.g., sets of hand-picked general event-related 'cue' words) and they report considerably good performance [77]. Connell et al. employed a heuristic function that used named entity similarity to modify the ALLTERMS cosine coefficient [40], and the results were far better than our baseline.

5.6 Conclusions

We have presented a framework for combining different kinds of ways of measuring document similarity. The document representation is split into semantic classes, each of which can be assigned a specific term-term corre-

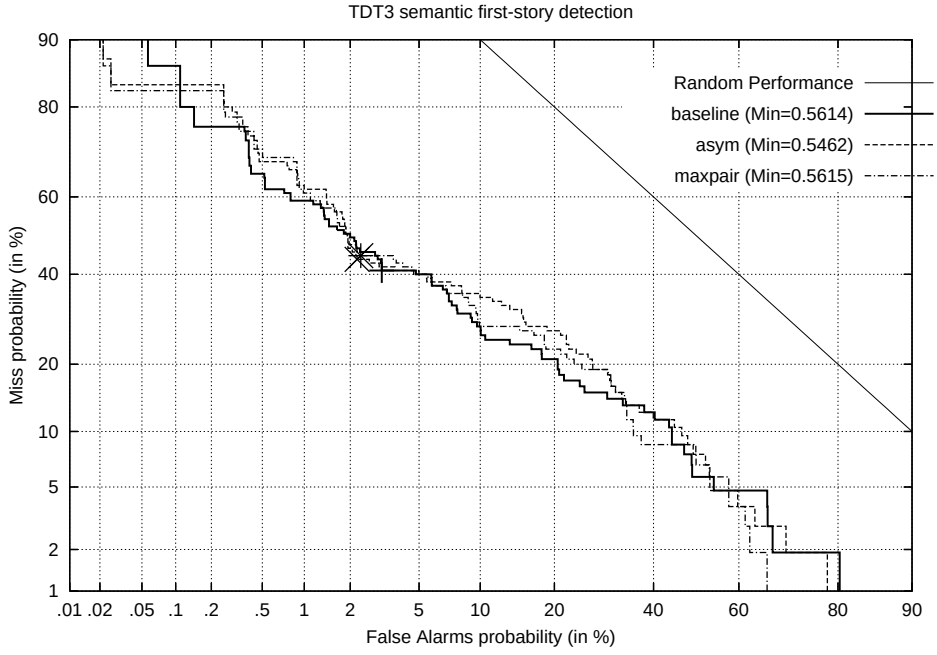

Figure 5.8: The topic-weighted DET curves of TDT-3 first-story detection.

lation. Unlike in previous work, this enables integration of external knowledge, e.g., geographical ontology or taxonomy of concepts, into the document comparison process. The class-wise similarities are combined into decision score by score models. To this end, we employed support-vector machines.

The adoption of semantic classes requires tools with which to recognize, normalize, and disambiguate named entities. Our named entity recognition was built on top of Conditional Random Field -based software by a team at Stanford University. Its F_1-scores ranged between 0.85 and 0.92. Our normalization of named entities, i.e., converting sequences of labeled tokens into actual names and unifying the variety of ways used to refer to the same person or organization, made use of partly manually built gazetteer. The F_1-scores of the normalization were slightly lower than in named entity recognition. Finally, we have presented a technique for disambiguation of placenames that associates geographical entities with placenames. The technique uses the unambiguous placenames occurring in the document in ranking the candidate locations. The evaluation showed an accuracy of 80%.

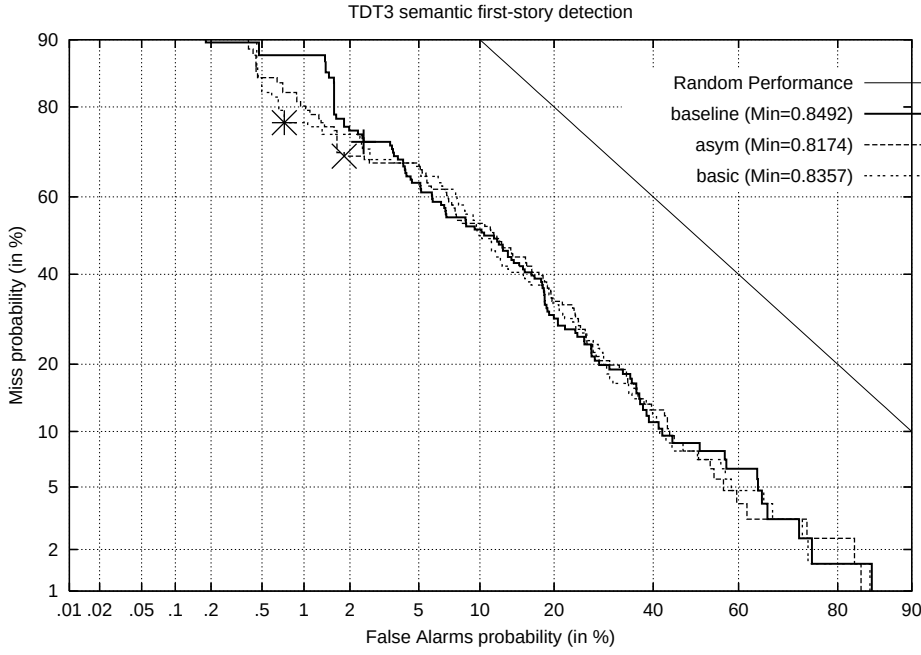

Figure 5.9: The topic-weighted DET curves of TDT-5 first-story detection.

Topic tracking using semantic classes improved the system's effectiveness significantly. Incorporating a geographical ontology into the location term comparison increased the effectiveness, but not consistently throughout the space of decision scores. The binary term-term correlations approaches displayed better overall performance.

The semantic class approach turned out not to be as useful in first-story detection. We failed to produce statistically significant improvement to 95% confidence with the larger corpora. We suspect our current FSD system is geared towards topic tracking and high precision, and as such does not accommodate for high recall required by first-story detection. From the different combinations of semantic classes, the ones employing asymmetric overlap for comparing named entities seemed to consistently perform better than others.

Chapter 6

Temporal information in document similarity

In the previous chapter, we put forward a framework for document similarity that can make use of relatedness of terms in an ontology. Here, we add yet another component: *time*. In Section 6.1, we pave the ground for exploiting temporal information by outlining ways to recognize and formalize temporal expressions automatically. Section 6.2 covers three different ways of using the temporal information in comparing documents. We describe the effectiveness with which we preprocess the documents to extract the temporal information, and present experiments in which we employ the extracted temporal information in Section 6.3. Section 6.4 is a conclusion.

6.1 Processing temporal expressions

A temporal expression is a sequence of tokens that denote time. There are different kinds of temporal expressions. Some temporal expressions are *explicit* [126], e.g., *2008-03-01*, *on the 24th of August 2003*, or *in October 2001*. No additional information is required in resolving them. There are also *implicit* expressions that contain a varying degree of indexical features like *last February*, *two weeks ago*, or *on Tuesday*. In order to formalize these expressions, we need to know the reference time and the verb tense. Yet, there are *vague* expressions such as *after several weeks* and *before June* the formalization of which is difficult.

We map all the expressions as an interval, i.e., a pair of dates, on a global time-line, *calendar*. This is accomplished by moving either the *utterance time* or the *reference time* according to the operators occurring in the expression. The utterance time is, quite simply, the time at with the

temporal expression is uttered. The reference time is a reference point defined by context. The processed temporal expression in a sentence expresses *event time*, the time at which the action expressed by the sentence takes place [117]. Consider following sentences:

> *The winter of 1974 was a cold one. The next winter will be even colder.*
> *The winter of 1974 was a cold one. The next winter was even colder.*

In the first sentence, the expression *next winter* (event time) refers to the next winter with respect to the time of utterance. In the latter, due to the past-tense of the verb, *next winter* refers to the following winter with respect to reference time (winter of 1974), i.e., the winter of 1975.

6.1.1 Recognition

Our recognition approach relies on automata that use dependency functions similarly to location recognition presented in Section 5.5.2. We use the dependency functions as categories, and map the terms of a temporal expressions to those categories. A list of categories is presented in Table 6.1. We consider the baseterms to be more like sub-categories denoting a basic unit of time. For example, the baseterm *day* contains terms referring to a day: morning, noon, day, afternoon, evening, night. Similarly, *weekday* consists of weekdays from Monday to Sunday.

Table 6.1: A sample list of categories of terms occurring in temporal expressions.

category	*terms*
baseterm	day, week, weekday, month, monthname, quarter, season, ...
indexical	yesterday, today, tomorrow
internal	beginning, end, early, late, middle
determiner	this, last, next, previous, the
temporal	in, on, by, during, after, until, since, before, later
postmodifier	of, to
numeral	one, two, ...
ordinal	first, second, ...
adverb	ago
meta	throughout
vague	some, few, several
recurrence	every, per
source	from

We employ manually produced finite-state automata in recognizing temporal expressions. Figure 6.1 portrays an automaton that recognizes temporal expressions with *monthname* as the baseterm. The input comprises a natural language sentence that is examined a word at a time. The automaton remains in the initial state unless a *temporal*, a *determiner* or an *ordinal* is encountered. The automaton runs for as long there are valid transitions or an end state is reached. In case an end state is reached and there are valid transitions, the automaton does not stop. If there is no valid transition from a given state, the automaton returns to the initial state, unless there have been previous end-state. In Figure 6.1, the valid end-states have double circles.

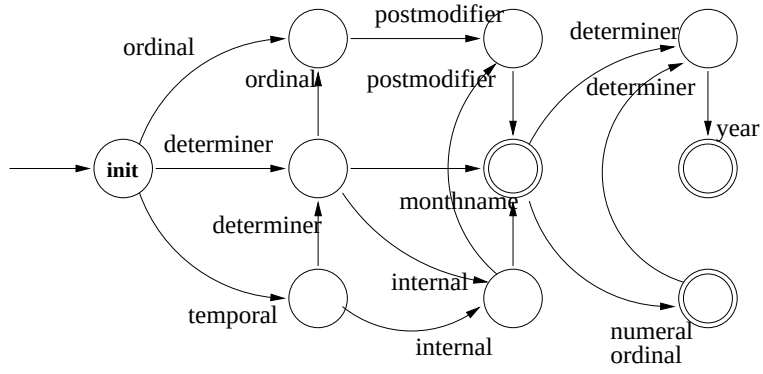

Figure 6.1: An automaton recognizing *monthname* related temporal expressions. The double circles represent valid end-states for the automaton, i.e., valid sequence of instances of categories to form a temporal expression.

The automaton can deal with input 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*". The expressions with a *vague* term, *a few weeks ago*, are recognized, but are omitted from further processing at this stage, because they are difficult to formalize.

Given the dependency parse tree, it is fairly easy to determine the tense of the relevant verb, since the parser typically finds a link between the verb and the first term of the temporal expression. In cases the link is not found, we simply examine the verbs in the same clause.

6.1.2 Calendar

In order to formalize the structure of the time-line, we adopt the calendar model from Goralwalla et al. [63] to construct an algebra for temporal

expressions. At the heart of it, we have the *time-line* composed of ordered "points of time".

Definition 6.1 *A global time-line $\mathcal{T}$ is a point structure with precedence relation $<_{\mathcal{T}}$. An interval $[t_i, t_k] \subset \mathcal{T}, t_i < t_k$ is a set of instants $\{[t_j, t_j] | \, i \leq j \leq k\}$.*

A *calendar* provides a space, where elements can be compared, ordered and transformed, i.e., where elements are in relation with other elements. Without a calendar, expressions like "last April" and "two weeks ago" would hardly make sense.

Definition 6.2 *A calendar C is a triplet $\langle \mathcal{T}, \mathcal{G}, \mathcal{F} \rangle$, where $\mathcal{T}$ is the global time-line of C, $\mathcal{G}$ is the set of granularities $\mathcal{G} = \{G_1, G_2, \ldots, G_n\}$ with G_1 being the coarsest and G_n the finest granularity, and $\mathcal{F}$ is the set of conversion functions between the granularities,*

$$\mathcal{F} = \{f | f^{G_k \to G_{k+1}} : \mathbb{N}^k \to \mathbb{N}, 1 \leq k \leq n-1\},$$

where variables $(i)_1^k \in \mathbb{N}^k$ represent the ordinals of the granularities from the coarsest to the finest up to kth granularity. The outcome of the function $f^{G_k \to G_{k+1}}$ is the number of units of granularity G_{k+1} contained by the coarser unit of granularity G_k.

Granularities are the basic units of time. For example, a Gregorian calendar has granularities

$$\mathcal{G} = \{G_{year}, G_{month}, G_{week}, G_{day}, \ldots\}.$$

A business calendar usually augments this with $G_{quarter}$, and another related to sports could have G_{season} which would differ from the 'four seasons', for example. One could conduct a conversion from a granularity to another by the virtue of calendar specific conversion functions, e.g.,

$$\begin{aligned} f_{\mathcal{C}}^{G_{year} \to G_{month}}(i_{year}) &= N_{G_{month}}, \\ f_{\mathcal{C}}^{G_{month} \to G_{day}}(i_{year}, i_{month}) &= N_{G_{day}}, \\ &\vdots \\ f_{\mathcal{C}}^{G_{second} \to G_n}(i_{year}, i_{month}, \ldots, i_{second}) &= N_{G_n}, \end{aligned}$$

where the result $N_{G_{month}}$ is the number of months in a year, $N_{G_{day}}$ is the number of days contained by the given month of the year, and the final outcome, N_{G_n} is the number of the units of *bottom granularity* of $\mathcal{T}$. The bottom granularity denotes the shortest temporal unit recognized by the

calendar and is the basis for all other granularities. It could be a day, a second, or a nanosecond. Naturally, the values N_{G_k} have a lower and upper bound depending on the values of $i_1, \ldots, i_n$. For example, $f_{\mathcal{C}}^{G_{month} \to G_{day}} \in \{28, 29, 30, 31\}$:

$$f_{\mathcal{C}}^{G_{month} \to G_{day}}(2002, 12) = 31, \quad f_{\mathcal{C}}^{G_{month} \to G_{day}}(2001, 11) = 30,$$
$$f_{\mathcal{C}}^{G_{month} \to G_{day}}(2000, 2) = 29, \quad f_{\mathcal{C}}^{G_{month} \to G_{day}}(1999, 2) = 28.$$

In the above, the year 2002 represents the 2002th calendric element [1] in the granularity G_{year}, and the month 12 represents the 12th calendric element in the granularity G_{month}, respectively.

The granularity G_{monday} contains every seventh element of G_{day}, and $G_{december}$ every twelfth element of G_{month}. Analogously, $G_{weekend}$ comprises all the Saturday-Sunday pairs.

6.1.3 Normalization

Ultimately, we represent the temporal expression by an interval on the time-line. We transform the natural language temporal expression into operators that shift the date of utterance back or forth and expand it into an interval of appropriate size on the time-line. We call this process *normalization*, since it eliminates the variety in surface forms and provides a formalized interpretation for a temporal expression.

Explicit temporal expressions need only minimal normalization. Usually, it suffices to convert monthnames like 'July' into calendric elements. Implicit temporal expressions, on the other hand, require the time of reference and the tense of the relevant verb to be normalized. Expressions like 'last Monday' or 'this week' denote different temporal intervals depending on the time they are uttered.

Definition 6.3 *The normalized form of a temporal expression is a pair* $\langle t_i, t_j \rangle, t_i, t_j \in \mathcal{T}$ *of points on the time-line that denotes an interval* $[t_i, t_j] \subset \mathcal{T}$ *if* $t_i < t_j$*, or an instant* $[t_i, t_i] \subset \mathcal{T}$ *if* $t_i = t_j$*.*

In the following, we content ourselves with day-level, i.e., the bottom granularity is day, $G_n = G_{day}$. References to mornings, afternoons, and evenings are all interpreted as instances of baseterm *day*. We use datestamps like 20020626 as proper names for points on the time-line.

[1] We simplify things slightly, as Goralwalla et al. [63] count the year elements from the decree of Pope Gregorian XIII that led to adoption of the Gregorian calendar in 1582.

Definition 6.4 *A function $\pi : \mathcal{G} \times \mathcal{T} \rightarrow \mathcal{T}$, $\pi(G, t_r) = t_i$ returns the start point of an element of the granularity G that is previous to t_r. Similarly, a function $\rho : \mathcal{G} \times \mathcal{T} \rightarrow \mathcal{T}$, $\rho(G, t_r) = t_i$ returns the start point of the element of the granularity G that is next to t_r.*

Assuming a reference time of $t_r = 20020626$, that is, Wednesday, June 26th 2002, then the preceding beginning of a week would be $\pi(G_{week}, t_r) = 20020624$, the preceding beginning of a month $\pi(G_{july}, t_r) = 20010701$, the succeeding beginning of a year $\rho(G_{year}, t_r) = 20030101$, and the succeeding Tuesday $\rho(G_{tuesday}, t_r) = 20020702$, respectively.

Definition 6.5 *A function $\delta_L : \mathcal{G} \times \mathbb{N} \times \mathcal{T} \rightarrow \mathcal{T}$, $\delta(G, n, t_r) = t_i$ shifts the point t_r n elements of granularity G to the left on the time-line. A function $\delta_R : \mathcal{G} \times \mathbb{N} \times \mathcal{T} \rightarrow \mathcal{T}$ performs a shift similarly to the right on the time-line.*

The size of the shift is converted into days, because the length of the granularity G_{month}, for example, varies considerably. For instance, *three months ago* with respect to $t_r = 20020626$ would result in left shift

$$\begin{aligned} \delta_L(G_{months}, 3, t_r) &= \delta_L(G_{day}, \textstyle\sum_{k=1}^{n} f_{\mathcal{C}}^{G_{month} \rightarrow G_{day}}(y(t_r), m(t_r) - k), t_r) \\ &= \delta_L(G_{day}, (31 + 30 + 31), t_r) \\ &= 20020326, \end{aligned}$$

where functions $y : \mathcal{T} \rightarrow \mathbb{N}$ and $m : \mathcal{T} \rightarrow \mathbb{N}$ return the year and the month of a given point on the time-line $\mathcal{T}$. The end point of a formalized temporal expression is obtained by shifting the start point to the right for the length of the baseterm (minus one day) as shown in Table 6.2.

Sometimes the expressions contain prepositions such as *by*, *before*, *until* and *since* that convey vagueness. Although logically they leave one of the end points open, we interpret them as a closed interval. For instance,

Table 6.2: A list of formalized baseterms (here start point is in the past). Once the right shift is converted to days, the number of days should be decreased by one.

baseterm	start	end
year	$\pi(G_{year}, t_r)$	$\delta_R(G_{year}, 1, \pi(G_{year}, t_r))$
season	$\pi(G_{season_x}, t_r)$	$\delta_R(G_{season_x}, 1, \pi(G_{season_x}, t_r))$
quarter	$\pi(G_{Q_n}, t_r)$	$\delta_R(G_{Q_n}, 1, \pi(G_{Q_n}, t_r))$
month	$\pi(G_{month}, t_r)$	$\delta_R(G_{month}, 1, \pi(G_{month}, t_r))$
week	$\pi(G_{week}, t_r)$	$\delta_R(G_{week}, 1, \pi(G_{week}, t_r))$
day	t_r	t_r

the expression "before Thursday" is not interpreted as an interval from beginning of the time to the end of given Thursday. Instead, both the start and end points of the interval are assigned to the given Thursday. The interpretation of prepositions is listed in Table 6.3.

We interpret *internal* terms as about one third of the length of the baseterm. The expression *the end of August* would thus cover the time-line from the 20th to 31st of August, the expression *early this week* would cover a week from Sunday to Tuesday, and *mid-1990s* would denote an interval from beginning of 1993 to the end of 1996.

The process of formalization can be outlined as follows. First, we extract the baseterm of the expression. Second, we evaluate the start point of the interval denoted by the baseterm by examining the tense and the reference time. Third, we shift this start point to the left or to the right if need be. Finally, depending on the baseterm we determine duration of the denoted interval and shift end point accordingly.

Table 6.4 shows a tripartite division of temporal expressions. The prefix typically modifies the duration of the baseterm (*late*) or shifts the baseterm with respect to some coarser baseterm (*15th*, *second*). The postfix then shifts the baseterm to the left or to the right. Naturally, all expressions do not fall into this division (e.g., *October 11*). In practice, we employ finite-state automata with a window (previous-current-next) to be able to cope with the variety of expressions and to be able to disambiguate terms.

6.2 Using temporal information in TDT

We now present three approaches to integrating time to document similarity. The first two modify the term-term correlations, and the third compares the temporal intervals of the documents directly.

Table 6.3: The interpretation of non-baseterms.

term	*span*
after, from, since	$[t_i, t_i]$
before, by, until	$[t_i, t_i]$
during, in, throughout	$[t_i, \delta_R(G, 1, t_i)]$
on	$[t_i, t_i]$
beginning,early	$[t_i, \delta_R(G, \frac{1}{3}, t_i]$
mid-, middle	$[\delta_R(G, \frac{1}{3}, t_i), \delta_R(G, \frac{2}{3}, t_i]$
late, end	$[\delta_R(G, \frac{2}{3}, t_i), \delta_R(G, 1, t_i]$

Table 6.4: Examples of expressions formalized with respect to June 26, 2002.

prefix	*baseterm*	*postfix*	*start*	*end*
on the 21st	(day)	of September last year	20010921	20010921
in late	May		20020520	20020531
during the second	week	of June 2001	20010611	20010617
before the end of	October	this year	20020626	20021031
after	Friday		20020628	–
six	years	ago	19960626	19960626

6.2.1 Simple time-decay

Many TDT papers note that the new stories relating to a news event tend to be published in bursts. The initial story may be succeeded by some follow-up reports, but ultimately and usually in a relatively short time the topic is no longer reported. Of course, there are some deviations from this burstiness, where the media's interest does not fade away in a week or two.

The most straight-forward way of exploiting temporal information in topic detection and tracking is to gradually penalize the document similarity for difference in publication dates. This *time-decay* was adopted in TDT early on [11]. Previous work suggests that time-decay does not work well in FSD [13, 29].

We adopted an exponential time-decay function similar to one used by Nallapati et al. [107] which seemed to work well on topic clustering. It models the rates of decay per topic by incorporating the time span of the topic into the decay function:

$$decay(t_d, t_c) = \exp\left(-\frac{\alpha|t_d - t_c|}{T_c}\right), \tag{6.1}$$

where t_d is the publication date of the document d, t_c the publication date of the latest document in topic centroid c , $|t_d - t_c|$ the difference in days between the publication of the document and the topic, α a time decay factor, and T_c the number of days between the first and last document in the topic.

Given semantic vectors $\mathbf{d}_i$ and $\mathbf{q}_i$ of semantic class S_i, the time-decay modified similarity coefficient is a function

$$sim_i^{decay}(\mathbf{d}_i, \mathbf{q}_i) = decay(t_d, t_q)\ sim_i(\mathbf{d}_i, \mathbf{q}_i). \tag{6.2}$$

6.2.2 Davidsonian indexing

So far, we have viewed temporal expressions as properties of documents by which they are anchored onto the time-line. However, temporal expressions seem to really anchor only sentences or paragraphs in which they occur. We briefly discussed Davidson's analysis of action sentences in Section 4.1.2. Assuming a sentence expresses an action of some sort, we associate the temporal expressions occurring in the sentence with the hidden variables, i.e., events, of the logical form of the sentence. A news story can be seen to comprise several action sentences, the hidden variables of which refer to real world events. Temporal expressions would thus be properties of these hidden variables.

Let us take a look at some excerpts from TDT-2 corpus. They all contain the location term 'Cambodia' but refer to distinct real world events: (1) the era of Khmer Rouge regime, (2) the U.N. sponsored elections of 1993, (3) the coup of July 5-6, 1997, and (4) the scheduled elections for July 26, 1998.

> (1) At least a million Cambodians died during the years the Pol Pot -led Khmer Rouge was in power *in Cambodia*, **from 1975 to 1979**.
>
> (2) After the coalition government was installed *in Cambodia* following the successful election **in 1993**, Thailand joined other countries to recognize the Cambodian government run by the dual prime ministers of Prince Norodom Ranariddh and Hun Sen as the legitimate representatives.
>
> (3) The two sides have been locked in a conflict *in northern Cambodia* **since the July** coup that ousted Prince Ranariddh as first Prime Minister, and left Hun Sen firmly in control.
>
> (4) Ranariddh has been in exile since the coup and wants to return *to Cambodia* to participate in elections scheduled for **July 26**.

In a vector-space model, these fragments would be found similar to one another, because they share some very informative terms. The inner-product – be it based on a binary or non-binary term-term correlation or not – would treat all occurrences of 'Cambodia' in the same way, which works well for typical full-text information retrieval. With a news event -based retrieval focus, however, this approach is not as useful, because it fails to distinguish that the excerpts really discuss different events.

Motivated by Davidson's analysis, we propose recording the temporal context for each term occurrence and using this temporal context in document similarity. The *temporal context* of a term is the temporal interval denoted by the temporal expression in the sub-clause or the sentence in

which the term occurs. If the sentence is void of temporal expressions, the temporal context is defined by temporal interval of the previous temporal expression. If the document has no temporal expressions, the default temporal context is the publication date as the news media tends to live in a constant present. Thus, every occurrence of every term is associated with some interval on the time-line. Figure 6.2 depicts the temporal association of terms for examples 1 and 3.

When comparing two documents, we would like to reward two documents having common terms occurring in the same or similar temporal contexts. Suppose we have a new example 5 (published in 1998), and we want to find the most similar previous example.

(5) Hun Sen deposed Prince Ranariddh in a coup **last July**.

When we compare this new example to the excerpts using their temporal contexts, we would find example 3 the closest, because the matching terms 'coup', 'Hun Sen', and 'Ranariddh' are used in the most similar temporal context, i.e., that of July 1997.

The physical document representation need not change all that much. A simplified example is illustrated in Figure 6.3. A document contains a list of temporal intervals denoted in the text plus the publication date. Then, each occurrence of each term is recorded with a corresponding pointer, i.e., a pointer to temporal context in which the term occurred in the text. This can be seen analogous to retrieval systems that use proximity searches, i.e., searches that require two terms occur near each other in the text (see, e.g., [99]). In order to enable proximity determination between the occurrences of terms, an offset needs to be stored for each occurrence of each term. Here, we are doing the same, but our offset is not words or characters from the beginning of the document, but a temporal offset on a time-line.

Much of the previous work on temporal expressions has had to do with artificial intelligence, where the objective is to infer, whether something occurs before, during, or after something else. In contrast, we pursue a similarity score, and a strict overlap-based approach would disregard consecutive days, or dates that are a few days apart, for instance.

A simple solution is to measure the average distance between two intervals. We use a function

$$dist(A,B) = \log(1+\frac{1}{4}(|A_s-B_s|+|A_e-B_e|+|A_s-B_e|+|A_e-B_s|)), \quad (6.3)$$

where A_s is the start point of interval A and A_e is the end point of interval A (similarly for B). It measures on one hand how far the the intervals are from each other and on the other hand how long a period they cover. Yet,

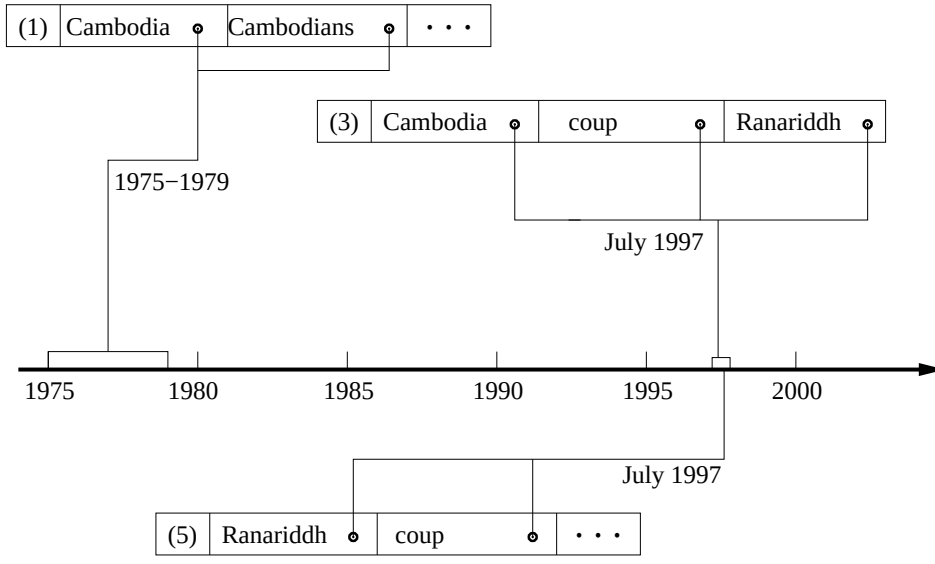

Figure 6.2: An example of Davidsonian indexing. Each occurrence of each term in the term vector is anchored onto to the time-line. The examples 1 and 3 both contain 'Cambodia' but in disparate and distant temporal contexts, so they are not likely to discuss the same topic. On the other hand, both example 3 and example 5 contain terms 'coup' and 'Ranariddh' with occurrences anchored onto July 1997. Thus, they are likely to discuss the same event.

it does not really matter if the intervals are eight or nine years apart, so we use the logarithm to curb the linear growth. When $dist(A, B)$ yields zero, the intervals A and B refer to the same day, i.e., the start dates and end dates for the intervals are the same.

We ran distance comparisons on TDT-2 training set to find out, if on-topic distances would be any different from off-topic ones. As a term can occur multiple times in a document in different temporal contexts, the comparison goes through all pairs of temporal contexts of the terms and uses the best match, that is, pair of intervals with minimal distance. Table 6.5 shows the results of minimum *dist* for each term comparison for 'all-terms' class, that is, best temporal match when two terms were compared in the inner product. The *dist* values are divided into value ranges indicated by the first column.

About 20% of the on-topic term comparisons have *dist* value of 0, which means the best matching temporal expressions denote the same day. Only

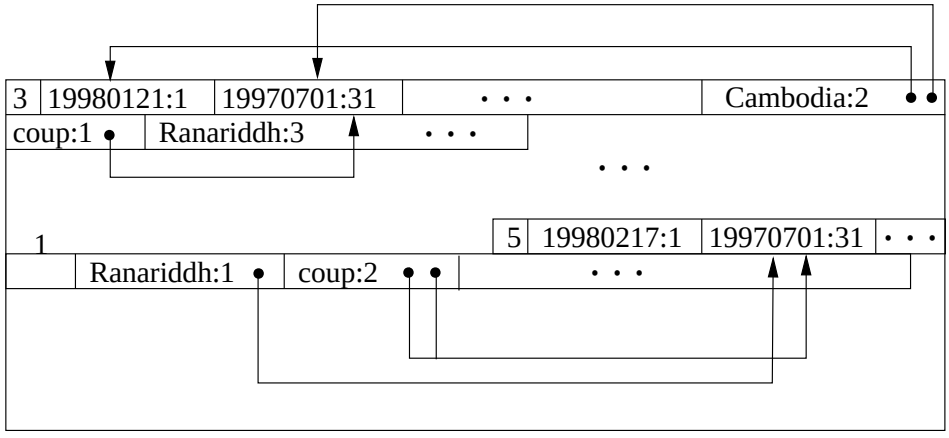

Figure 6.3: A representation of physical documents in a file. Each document contains a list of temporal intervals denoted in the text, and each term in the term vectors point to these temporal intervals.

4% of off-topic term comparisons have the same day as their best matching temporal context, but the absolute count is overwhelming. The *dist* values up to 2 correspond roughly to temporal intervals of the same week, e.g., two days within the same week or a week vs. a weekday. Values up to 4 correspond roughly to intervals in the same month, and values up to 6 in the same year, respectively. The on-topic documents discuss mostly events taking place in the same week, while off-topic documents have a broader temporal span.

To coin the results of Table 6.5 into a term-term correlation, we use a heuristic function that yields 1.0, if the temporal contexts of the terms

Table 6.5: The minimum $dist(A, B)$ for on-topic and off-topic term comparisons in the 'all-terms' class.

	on-topic			*off-topic*		
$x = \min dist(A, B)$	*count*	*%*	*cumulat.*	*count*	*%*	*cumulat.*
$x = 0$	14,340	0.191	0.191	576,852	0.041	0.041
$0 < x \leq 2$	35,819	0.476	0.667	4,157,178	0.297	0.338
$2 < x \leq 4$	15,141	0.201	0.869	5,117,876	0.366	0.704
$4 < x \leq 6$	6,417	0.085	0.954	2,015,097	0.144	0.848
$6 < x \leq 8$	3,323	0.044	0.998	1,935,189	0.138	0.986
$8 < x \leq 10$	149	0.002	1.000	189,827	0.014	1.000
$10 < x \leq 12$	0	0.000	1.000	892	0.000	1.000

coincide on the week-level, and gradually degrades for distances greater than that. Let d be a term occurring f_d times, and q be a term occurring f_q times. The term-term correlation for semantic class i using temporal contexts is a function

$$cor_i^{davidson}(d,q) = (1 - \beta_i \min_{jk} (dist(d_j^t, q_k^t))), \tag{6.4}$$

where β_i is a decay factor, $1 \leq j \leq f_d$, $1 \leq k \leq f_q$, d_j^t and q_k^t are the jth and kth temporal context of terms d and q, respectively. We trained the decay factor β_i separately for each semantic class i; the values ranged between 0.0 and 0.1. If the minimum distance $dist$ for d and q is 0, then $cor_i^{davidson}(d,q)$ yields 1. In Table 6.5 the values of $dist(A,B)$ of Equation 6.3 range from 0 to 12. The further the closest intervals are from each other, the closer to 0 the correlation $cor_i^{davidson}$ decreases.

Typically, a document has only a handful of temporal expressions and a substantially greater number of terms. The computation of document similarity using $cor^{davidson}$ can benefit from a matrix, where the distances $dist(d_j^t, q_k^t)$ are stored before iterating through the term vectors.

A similarity coefficient using Davidsonian term-term correlation lends itself to the modified inner-product of Equation 5.2 that replaced identity relation between terms with an ontology-based correlation with which non-identical terms could be found relevant. In itself, the Davidsonian term-term correlation does not examine or use relations between non-identical terms, but only the associated temporal contexts of the terms. It discounts the correlation over time, i.e., over the difference in temporal contexts. We use Davidsonian indexing together with cosine similarity coefficient (Equation 3.6),

$$sim_i^{cos/david}(\mathbf{d}_i, \mathbf{q}_i) = \frac{\sum_{j=1}^{n} cor_i^{david}(d_{ij}, q_{ij}) d_{ij} q_{ij}}{||\mathbf{d}_i||\,||\mathbf{q}_i||}, \tag{6.5}$$

where $\mathbf{d}_i$ and $\mathbf{q}_i$ are n-vectors of semantic class S_i and $||\mathbf{d}_i||$ and $||\mathbf{q}_i||$ their lengths. The Davidsonian term-term correlation $cor_i^{david}(d_{ij}, q_{ij})$ modifies multiplication of term-weights of term t_{ij}, when both vectors $\mathbf{d}_i$ and $\mathbf{q}_i$ have a non-zero value for term t_{ij}. Thus, two documents composed of exactly the same terms but disparate temporal contexts would yield a cosine score less than 1.

6.2.3 Temporal expressions as a semantic class

Temporal expressions can be treated as a semantic class, with normalized expressions as terms. We use the distance function from the earlier section

as a term-term correlation:

$$cor_i^{temp}(A, B) = 1 - \beta_i \; dist(A, B)), \tag{6.6}$$

where β_i is a decay factor of Equation 6.4 and $dist(A, B)$ is the average distance between two intervals defined in Equation 6.3. The similarity coefficient of TEMPORAL vectors is simply the highest correlation between the temporal intervals vectors,

$$sim_i^{temp}(\mathbf{d}_i, \mathbf{q}_i) = \max_{jk} \; cor_i^{temp}(d_{ij}, q_{ik}), \tag{6.7}$$

where $1 \leq j \leq |\mathbf{d}_i|$ and $1 \leq k \leq |\mathbf{q}_i|$. The similarity coefficient of TEMPORAL semantic class represents the best match of temporal intervals between two documents. As such, it is likely to yield high scores for documents published the same day.

6.3 Experiments

6.3.1 Preprocessing

We annotated manually 1,192 sentences from TDT-2 training data. The set was from randomly picked native English newswire documents. A temporal expression was considered to be something that can be reasonably well mapped onto the time-line. Thus, we focus on terms listed in Table 6.1. We did not include events (e.g., Easter, Christmas, elections), vague expressions ("some time ago", "after the elections", "until next time"), or references to time of day (e.g., "2 pm", "at 5 o'clock").

These restrictions left us with a total of 266 temporal expressions, 140 of which we considered *simple*, that is, containing only a preposition and an indexical (e.g., "today", "on Friday", "yesterday morning") or a numeral referring to a year (e.g., "in 1998"). The remaining 126 expressions contained more complex syntactic or functional relations and were thus regarded as *composite expressions* (e.g., "early last week", "six years ago", "the following week"). The division is not crucial in any way, it merely gives an idea what the temporal expressions were like.

Recognition

For temporal expression recognition, we used automata as described in Section 6.1.1. The disambiguation between monthnames ('May', 'March') and verbs ('may', 'march'), for instance, were resolved with syntactic information provided by Connexor parser.

The temporal expression recognition ran quite well on our test set. Only two expressions were missed or incompletely recognized resulting in recall of 99% and a precision of 100%. The good results are partly due to limited scope in what we acknowledged as a temporal expression. Also, the test set was of modest size. However, given the small size of relevant terms (Table 6.1), the shallow morphology of English language, and standard use of English in the newswire data, we think that the recognition of temporal expressions is not particularly difficult.

Normalization

We set the bottom granularity for normalization to 'day'. Temporal expressions referring to granularities shorter than that (e.g., "morning", "afternoon") were considered instances of 'day'.

There were 15 normalization errors in the 266 temporal expressions, mostly arising from incorrectly extracted verb tense leading to wrong direction with the shift operations when mapping the expressions onto the timeline. This translates to accuracy of 94%. In most cases, Connexor parser provided us with functional dependency tree from which we could find the relevant verb, but there were incomplete sentences and sub-clauses for which no parse tree was available.

6.3.2 Topic tracking

The temporal topic tracking system was the same as the semantic tracking system in Section 5.5.3. The document representations and the posting lists have been augmented with temporal information.

Σ-signatures

The Davidsonian indexing does not provide a strong term-term correlation on its own. The distance function *dist* yields 1.0 for terms that have been used in the context of the same day, and therefore we use it to modify other term-term correlations. Following the notation of Section 5.5.3 we denote the Davidsonian experiments as

$$\begin{aligned}\Sigma_{david} \quad = \quad & \Sigma_{all}^{cos/david} \cup \Sigma_{ter}^{cos/david} \cup \Sigma_{org}^{cos/david} \cup \Sigma_{pla}^{cos/david} \cup \Sigma_{per}^{cos/david} \\ & \cup \Sigma_{loc}^{cos/david} \cup \{f\}.\end{aligned}$$

This means that all semantic classes utilize cosine-TFIDF for similarity coefficient, but the term-wise comparisons are modified by the Davidsonian term-term correlation as shown in Equation 6.5.

Likewise, the time-decay was used as a modifier to cosine-TFIDF similarity. The further apart the documents are in terms of publication time, the more their term-term similarity is decreased. We denote this kind of run with

$$\begin{aligned}\Sigma_{decay} &= \Sigma_{all}^{cos/decay} \cup \Sigma_{ter}^{cos/decay} \cup \Sigma_{org}^{cos/decay} \cup \Sigma_{pla}^{cos/decay} \cup \Sigma_{per}^{cos/decay} \\ &\quad \cup \Sigma_{loc}^{cos/decay} \cup \{f\}.\end{aligned}$$

The third experiment with temporal information treated formalized temporal expressions as a distinct semantic class with temporal intervals as terms and Equation 6.7 as the similarity coefficient. Thus, we have seven semantic classes,

$$\Sigma_{temp} = \Sigma_{all}^{cos} \cup \Sigma_{ter}^{cos} \cup \Sigma_{org}^{cos} \cup \Sigma_{pla}^{cos} \cup \Sigma_{per}^{cos} \cup \Sigma_{loc}^{cos} \cup \Sigma_{time}^{temp} \cup \{f\}.$$

The score models were based on SVM like described in Section 5.5.3.

Results

The results from temporal topic tracking are shown in Table 6.6. What is obvious from the table is the consistently poor performance of time-decay. It is slightly improved when the topics are defined with four sample stories, but still the tracking costs hovers at or above 0.50, which is very poor. The miss rates are high, which would imply that the time-decay decreases the similarity scores of on-topic comparisons too much. In other words, documents discussing the same topic often occur further apart in time for any advantage from a decay function. Despite the short life-span of many topics, there are topics that evolve in a slower pace, and in topic-weighted evaluation their incompatibility with the time-decay weighs heavily on the effectiveness.

Using the temporal information as a semantic class does not improve the effectiveness either. Since it simply augments the Σ_{basic} signature and yet performs significantly worse, it is a dead end. With this we can put some of our previous work on temporal vectors to rest.

The Davidsonian indexing, however, shows some improvement in the minimum detection costs barring TDT-2 $N_t = 4$ and TDT-3 $N_t = 1$ that were difficult to improve the performance with in previous chapter, too. Although the difference between the temporal tracking and the best semantic tracking (Σ_{resnik}) in terms of the minimum detection cost is narrow, the DET curve of Figure 6.4 suggests a consistent improvement over both the baseline and the best performing semantic tracking. On the other hand,

Table 6.6: The temporal topic tracking results. The column 'kernel' specifies SVM kernel used ('all' stands for all-subsets kernel of Equation 5.17 and 'lin' for linear kernel of Equation 5.12) and the corresponding cost-factor ('l' for low, 'm' for medium). The column θ_{elim} refers to threshold for eliminative score models. If it is non-zero, an eliminative score model was used; a plain SVM otherwise. The column p gives the p-value by Student's t test as compared to the corresponding best baseline value in Table 4.5.

				Topic-Weighted at θ			Minimum Topic-Weighted			
signature	*kernel*	θ_{elim}	N_t	$p(\overline{r}\|\omega)$	$p(r\|\overline{\omega})$	$(C_{det})_{norm}$	$p(\overline{r}\|\omega)$	$p(r\|\overline{\omega})$	$(C_{det})_{norm}$	p
TDT2										
Σ_{david}	all-l	0.20	1	0.0766	0.0078	**0.1149**	0.0694	0.0078	**0.1078**	0.040
Σ_{decay}	all-l	0.08	1	0.6377	0.0004	0.6398	0.4727	0.0082	0.5129	
Σ_{temp}	all-m	0.14	1	0.1870	0.0039	0.2062	0.0888	0.0135	0.1551	
Σ_{david}	lin-m	0.00	4	0.1561	0.0069	**0.1222**	0.0630	0.0137	**0.1303**	0.114
Σ_{decay}	all-l	0.04	4	0.5544	0.0032	0.5700	0.4911	0.0078	0.5294	
Σ_{temp}	all-l	0.20	4	0.1044	0.0166	0.1860	0.0870	0.0152	0.1616	
TDT3										
Σ_{david}	all-l	0.16	1	0.0896	0.0147	**0.1617**	0.0848	0.0137	**0.1522**	0.992
Σ_{decay}	all-l	0.08	1	0.7199	0.0006	0.7229	0.5569	0.0116	0.6137	
Σ_{temp}	all-l	0.12	1	0.5506	0.0006	0.5535	0.2305	0.0142	0.2305	
Σ_{david}	all-m	0.06	4	0.0845	0.0152	**0.1590**	0.0849	0.0107	**0.1376**	0.017
Σ_{decay}	all-l	0.04	4	0.5542	0.0027	0.5675	0.4986	0.0093	0.5440	
Σ_{temp}	all-l	0.10	4	0.0461	0.0348	0.2168	0.0854	0.0159	0.1633	
TDT5										
Σ_{david}	all-l	0.00	1	0.1329	0.0063	**0.1635**	0.0687	0.0112	**0.1296**	0.008
Σ_{decay}	all-l	0.08	4	0.4949	0.0099	0.5455	0.3897	0.0117	0.4470	
Σ_{temp}	all-l	0.08	4	0.3261	0.0179	0.4140	0.2014	0.0134	0.2679	
Σ_{david}	all-l	0.06	4	0.0578	0.0184	**0.1478**	0.0803	0.0091	**0.1250**	0.001
Σ_{decay}	all-l	0.08	4	0.4095	0.0420	0.6197	0.4471	0.0098	0.4954	
Σ_{temp}	all-l	0.08	4	0.1984	0.0209	0.3009	0.2295	0.0101	0.2786	

the curve for temporal tracking would overlap with the curve of signature Σ_{basic} in Figure 5.6.

Figure 6.5 portrays the best tracking runs with TDT-5. As above, we omitted the poorly performing Σ_{decay} and Σ_{temp} from the plot. The temporal topic tracking Σ_{david} appears inferior to semantic Σ_{basic} run except for the mid-section of the curve. The graph of Davidsonian indexing pushes past the semantic run right in the area where we have both lower miss rate and and lower false-alarm rate. The minimum detection cost resides below both 10% miss-rate and 1% false-alarm rate.

In all, there appear to be benefits from integrating temporal contexts into document similarity, but the results are suggestive rather than conclusive. In the light of results of Table 6.5, the gains in the on-topic similarity are constrained by the overwhelming mass of off-topic data. Then again, this is merely the first step as there are many open questions in the Davidsonian indexing for further investigation. There are many ways in which

to compare temporal intervals, and we have shown only one that resulted in some gains in topic tracking.

6.3.3 First-story detection

The first-story detection experiments were conducted in the same way as in Section 5.5.4. The signatures evaluated are the same as in the tracking experiments of the previous section.

Results

The temporal first-story detection results are listed in Table 6.7. Again, the time-decay turns out to perform poorly. The suitability of using TEMPORAL semantic class in the signature is even worse. The temporal information of signature Σ_{david}, however, shows some improvement in the minimum detection costs for TDT-2 and TDT-5.

Davidsonian indexing fails to improve the results from baseline on TDT-3. The improvement on TDT-2 is only at 90% confidence and it is not statistically significant. The gain on TDT-5 data is statistically significant,

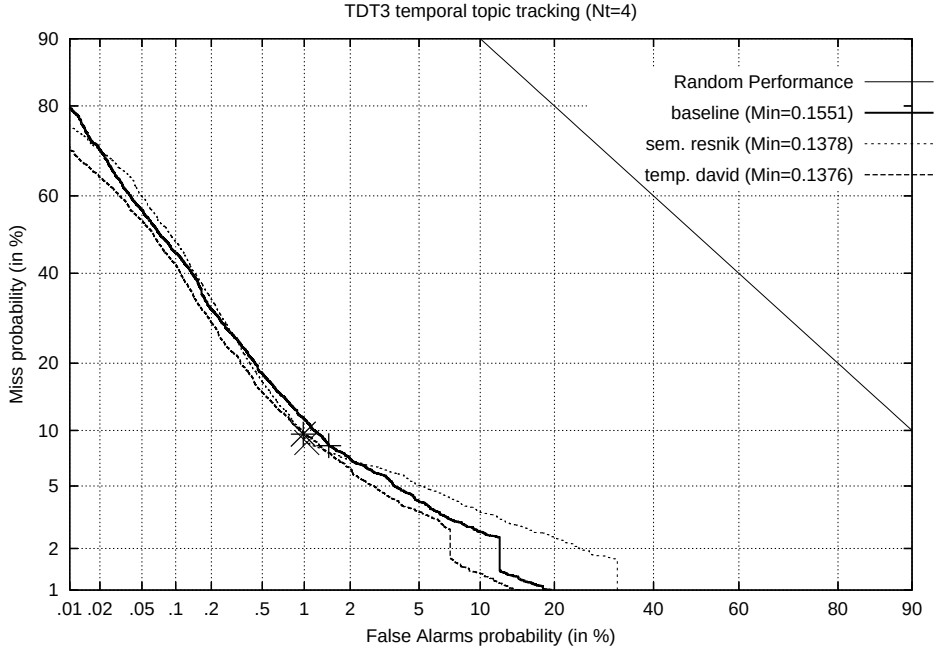

Figure 6.4: The DET curves of TDT3 temporal topic tracking ($N_t = 4$).

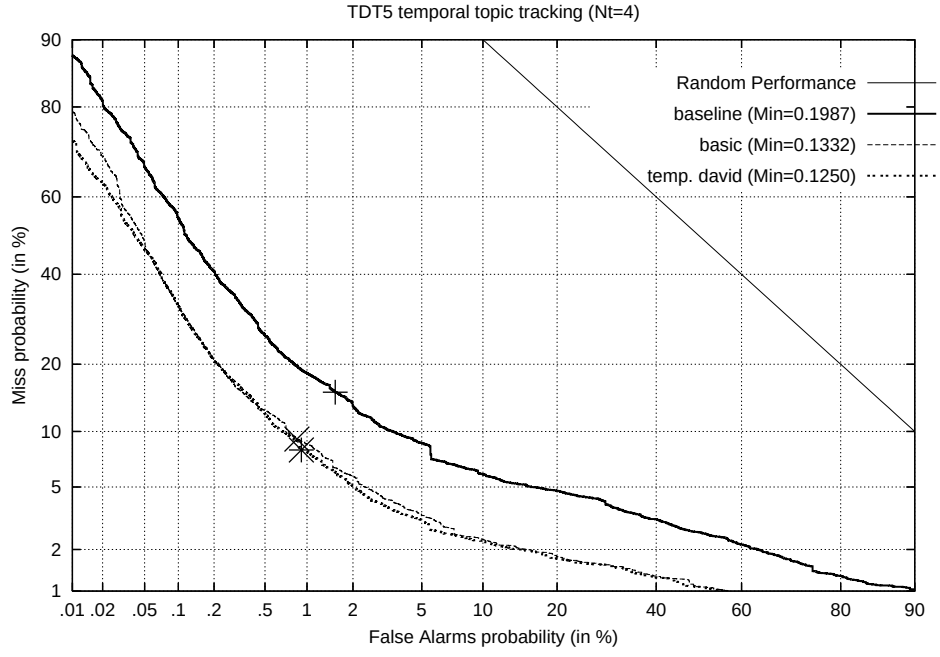

Figure 6.5: The DET curves of TDT5 temporal topic tracking ($N_t = 4$).

however, and signature Σ_{david} performs better than signature Σ_{asym} in Table 5.8.

Figure 6.6 represents the TDT3 DET curves for the baseline, the semantic Σ_{asym} signature, and the temporal Σ_{david} signature. Despite similar minimum detection costs, the baseline provides lower error rates than semantic and temporal methods throughout the space of decision scores. The Davidsonian indexing triumphs over the best semantic run implying there is some benefit from temporal information. The same trend is obvious from Figure 6.7 that illustrates the FSD DET curves with TDT-5 corpus. Although the temporal run has a lower minimum detection cost, it hovers above the baseline FSD DET curve for the most part.

In Section 5.5.4 we noted that the current semantic class system is somewhat geared towards topic tracking, and the same seems to apply here. The improved tracking effectiveness does not translate to increase in first-story detection performance.

In future, to combine semantic classes and temporal information with first-story detection we would steer away from the topic tracking approach somewhat and try to address FSD more directly. There is a need for rec-

Table 6.7: The temporal first-story detection results. The column 'kernel' specifies SVM kernel used ('all' stands for all-subsets kernel of Equation 5.17, 'rbf' for radial basis function of Equation 5.13, 'lin' for linear kernel of Equation 5.12) the corresponding cost-factor ('l' for low, 'm' for medium, 'h' for high). The column θ_{elim} refers to threshold for eliminative score models. If it is set to non-zero, an eliminative score model was used; a plain SVM otherwise. The column p gives the p-value by Student's t test as compared to the corresponding best baseline value in Table 4.9.

			Topic-Weighted at θ			Minimum Topic-Weighted			
signature	*kernel*	θ_{elim}	$p(\overline{r}\|\omega)$	$p(r\|\overline{\omega})$	$(C_{det})_{norm}$	$p(\overline{r}\|\omega)$	$p(r\|\overline{\omega})$	$(C_{det})_{norm}$	p
			TDT2						
Σ_{david}	all-l	0.10	0.4286	0.0159	0.4663	0.2679	0.0319	**0.3775**	0.094
Σ_{decay}	rbf-l	0.08	0.8929	0.0077	0.9024	0.7679	0.0117	0.8084	
Σ_{temp}	rbf-l	0.04	0.8036	0.0782	1.1824	0.9286	0.0081	0.9447	
			TDT3						
Σ_{david}	all-l	0.10	0.3810	0.0483	0.6086	0.4571	0.0253	**0.5810**	-
Σ_{decay}	rbf-l	0.08	0.9810	0.0037	1.0017	0.8762	0.0096	0.9280	
Σ_{temp}	rbf-l	0.04	0.3619	0.2413	1.5739	0.9905	0.0024	1.0026	
			TDT5						
Σ_{david}	lin-l	0.10	0.6905	0.0191	0.9553	0.7063	0.0165	**0.8082**	0.038
Σ_{decay}	rbf-l	0.08	0.6190	0.1190	1.2196	0.9762	0.0019	0.9822	
Σ_{temp}	rbf-l	0.04	1.0000	0.0004	1.0021	0.9921	0.0007	0.9942	

ognizing 'new' on the basis of the 'old', but it need not be based on topic tracking per se. For instance, keeping track of encountered named entities with their temporal contexts would help to recognize, when the entity is in a new situation. Thus, given the excerpts of Section 6.2.2, encountering 'Ranariddh' and 'Cambodia' in the temporal context of March 2003 would probably be a token of a new event. This kind of evidence would be collected for all terms and combined into a decision score.

6.4 Conclusions

This chapter presented a way to incorporate time into document similarity. In order to make use of temporal information, we used finite-state automata for recognition and normalization of temporal expressions occurring in the text. As a result, expressions were interpreted as intervals on a global timeline. The effectiveness of this preprocessing turned out to be quite good with accuracy above 90%.

We proposed three methods for introducing temporal information into the framework. The first was simple time-decay based on the difference of publication dates. So, the further the documents are apart, the more the

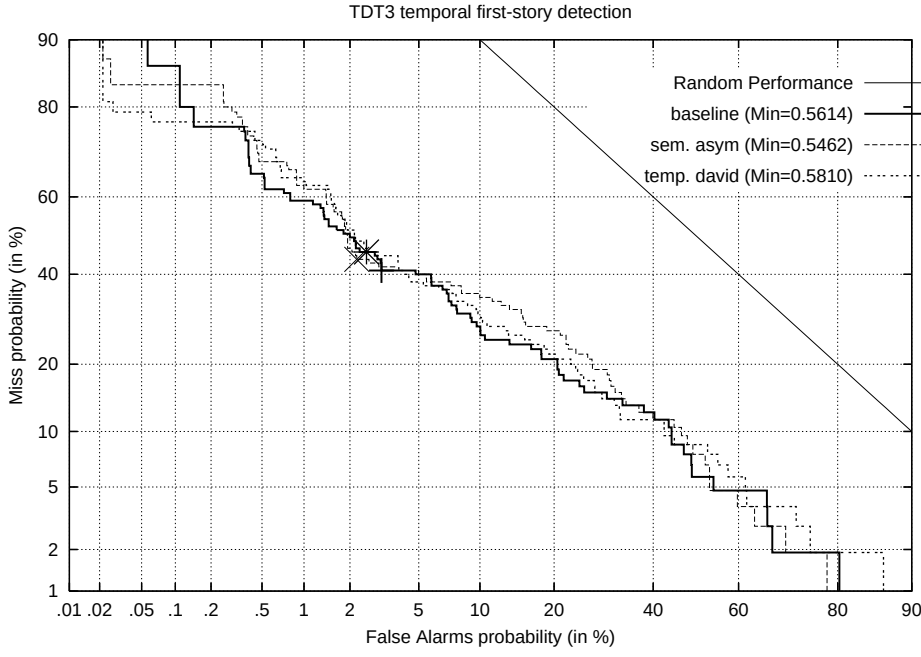

Figure 6.6: The DET curves of TDT3 temporal first-story detection.

document similarity is penalized. In the experiments, this worked neither in topic tracking nor first-story detection. This suggests that despite the burstiness of topics, i.e., that they have a short life-span in the media, is not easily exploited by time-decay.

Another method placed the formalized expressions into a designated semantic class using temporal intervals as terms. The method did not increase the effectiveness in topic tracking, much less in FSD.

The third approach was a novel way of recording the temporal context of each occurrence of a term in the document representation. Thus, when determining term-term correlations, we can account for terms occurring in a temporally proximate contexts. The approach had a positive impact on both topic tracking and first-story detection effectiveness, but the results are not quite conclusive in favor of Davidsonian indexing.

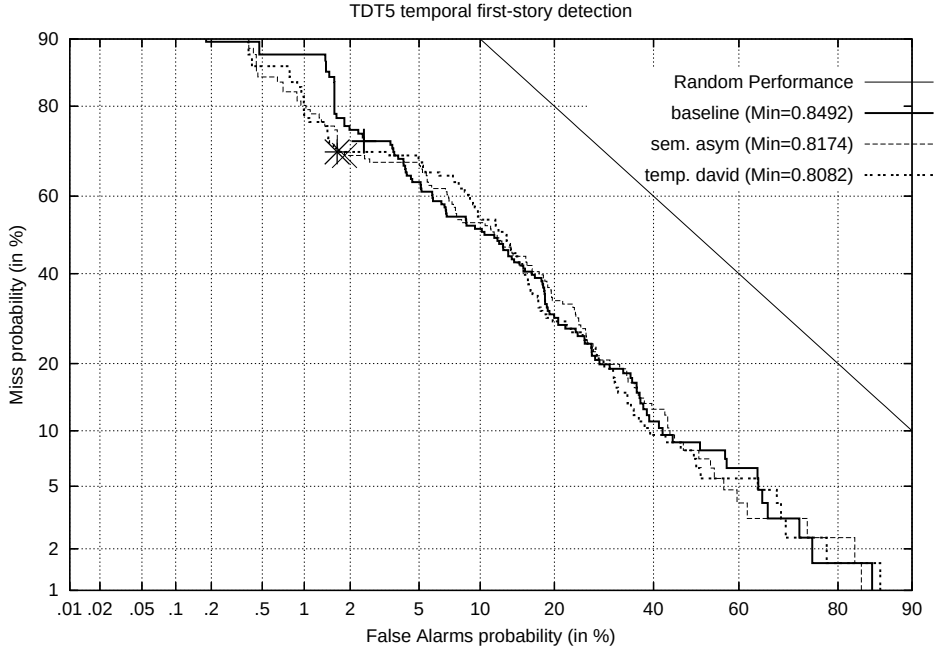

Figure 6.7: The DET curves of TDT5 temporal first-story detection.

Chapter 7

Adaptiveness in topic tracking

In Section 4.1.2, we discussed how real world events are transformed into news stories. An event is something happening somewhere, but the TDT system perceives only its natural language descriptions. We defined a topic as a set of related news events. The topic evolution, sometimes erroneously called event evolution, results from accumulation of new relevant stories. In this chapter, we will experiment simple adaptive models in order to address the changes in the topic. Section 7.1 describes the characteristics of adaptive systems and the mechanisms by which the adaptation ultimately changes the system behavior. In Section 7.2, we present our experimental results. Section 7.3 is a conclusion.

7.1 Adaptiveness

Adaptive systems change their behavior over time based on changes in the data. The change in behavior could be only a minor adjustment in the term weights or a complete re-modeling of topic representations. The incremental global document frequencies used in Sections 4.3.2 and 4.4.1 are a form of adaptation: the accumulating observations of terms affect the term-weights and document lengths. Usually, adaptive topic tracking refers to a system that modifies the topic model on encountering documents the system deems on-topic.

There are two kinds of adaptive systems: *supervised*, where the input or observation is labeled and verified by a supervisor, usually a human, and *unsupervised*, where the system makes changes autonomously and independently of human interference. In 2004 TDT benchmark evaluation there were experiments with supervised adaptive topic tracking: if the system made a 'YES' judgment, it was shown the true label of the document. The

benefits from supervised feedback were clear in nearly all the participated systems [6, 40, 159].

We examine unsupervised adaptive topic tracking. The basic idea is simple: if the similarity coefficient is "similar enough", the topic model is updated. This decision typically boils down to setting another threshold θ_{adapt} above the decision threshold, i.e., $\theta \leq \theta_{adapt}$, and if the decision score of a new document exceeds this second threshold, adaptation occurs.

We are using score models that produce signed distances from the decision hyperplane, and each kernel and parameter configuration would require gauging for a sound adaptation threshold. Instead of anchoring the adaptation threshold to the output of the score models, we use the ALL-TERMS similarity coefficient, i.e., the x_0 in the score vector $\mathbf{x}$ of class-wise similarities.

We now present two ways of adapting the topic models. The former compiles from all the relevant news stories while the latter focuses only on the latest ones.

7.1.1 Top-n terms

Suppose news stories pour into the system, and the topic they discuss evolves, i.e., there is new information. The *top-n* adaptation simply selects the best terms from the pool of on-topic documents. The best here means the terms with the highest term weights.

So, when adaptation occurs, the topic model is recomputed. All the terms in the on-topic documents are compiled into respective semantic class vectors. In case the same term occurs in numerous documents, the term frequencies are added. Then, the topic representation is built from semantic class vectors with just the n best terms per vector included; all the rest are omitted.

This approach emulates the narrativeness of news in that it accommodates the possibility of new, topic-wise important terms arising at some point. On the other hand, selecting terms from the pool of all on-topic documents sustains stability through core terms, like the ones we tracked down in Section 4.3.3.

7.1.2 Last-m documents

Much like top-n terms approach, last-m emulates the narrativeness, but assumes that the most recent documents are more useful for modeling the topic with respect to unseen stories. Upon re-computation of the model, *last-m documents* draws the candidate terms only from m latest documents.

With the topic model composed of latest (and sufficiently relevant) documents, the method prefers conforming to changes in the topic vocabulary to stability of the topic representation.

7.2 Experiments

7.2.1 Semantic tracking

The basic semantic system Σ_{basic} from Section 5.5.3 employs six semantic classes, cosine coefficient and InQuery term weights. Its score model uses all-subsets kernel with $\theta_{elim} = 0.08$ and low cost-factor. After a brief test run, we settled for $m = 4$ for last-m approach, and $n = 100$ for top-n approach. All topics are trained initially with four sample documents. Table 7.1 lists the adaptation effectiveness. Despite small improvement over no-adaptation, none of the methods outperform best static semantic or temporal runs of Chapters 5 and 6.

The top-100 adaptation decreases the detection costs compared to static runs consistently with all corpora. In signature Σ_{basic} the ALLTERMS class has cosine as the similarity coefficient. Since our adaptation threshold is anchored to x_0 of the score vector, the adaptation hinges on cosine. The adaptation threshold of 0.5 is fairly conservative as there are typically only few values that high. The majority of cosine scores range at a much lower level. The conservativeness is reflected in the results.

With TDT-2 the miss rate at θ is decreased slightly, while the false-alarm rate remains steady. This is something one would expect: the topic representation is augmented with new terms, which intuitively leads to catching the on-topic documents that were previously missed due low scores they yield. However, with TDT-3 and TDT-5 the effect is the opposite: the miss rate increases while the false-alarm rate drops. The more aggressively the topic model is re-computated, the lower the false-alarm rate sinks. This is something we did not anticipate. Apparently, since the decision threshold θ remains the same and the false-alarm rate drops, the adaptation leads to lower decision scores. Since the overall effectiveness is improving, the decrease has more impact on the off-topic than the on-topic documents. However, the minimum detection costs attain their decrease from decrease in miss-rate.

The last-4 adaptation is more aggressive in rebuilding the topic model, and in most cases this leads to worse effectiveness than without adaptation. With TDT-3, however, the optimized detection costs are marginally better than those of top-100. The last-4 run ($\theta_{adapt} = 0.3$) is illustrated in Figure 7.1. For the most part it runs right on top of the noadapt curve.

Table 7.1: The adaptive semantic ($N_t = 4$, Σ_{basic}, all-l, $\theta_{elim} = 0.08$) topic tracking topic-weighted results. The column 'adapt type' specifies type of adaptation and θ_{adapt} the adaptation threshold.

		Topic-Weighted at θ			Minimum Topic-Weighted		
adapt type	θ_{adapt}	*P(miss)*	*P(fa)*	$(C_{det})_{norm}$	*P(miss)*	*P(fa)*	$(C_{det})_{norm}$
TDT2							
noadapt	-	0.1295	0.0054	0.1558	0.0901	0.0078	0.1284
top-100	0.5	0.1104	0.0057	**0.1383**	0.0798	0.0083	**0.1204**
top-100	0.3	0.1221	0.0049	0.1459	0.0856	0.0080	0.1249
top-100	0.1	0.2505	0.0016	0.2586	0.0801	0.0106	0.1335
last-4	0.5	0.1844	0.0026	0.1974	0.1071	0.0070	0.1435
last-4	0.3	0.1765	0.0027	0.1899	0.0932	0.0088	0.1378
last-4	0.1	0.2428	0.0019	0.2522	0.0739	0.0120	0.1349
TDT3							
noadapt	-	0.1069	0.0121	**0.1660**	0.0879	0.0111	0.1422
top-100	0.5	0.1633	0.0055	0.1902	0.0761	0.0135	0.1411
top-100	0.3	0.1771	0.0047	0.2003	0.0823	0.0122	0.1414
top-100	0.1	0.2334	0.0031	0.2488	0.0977	0.0120	0.1558
last-4	0.5	0.1634	0.0055	0.1902	0.0761	0.0135	**0.1410**
last-4	0.3	0.1753	0.0047	0.1985	0.0805	0.0126	0.1412
last-4	0.1	0.2149	0.0033	0.2309	0.0844	0.0126	0.1459
TDT5							
noadapt	-	0.1112	0.0069	0.1451	0.0915	0.0085	0.1332
top-100	0.5	0.1147	0.0067	0.1476	0.0826	0.0106	0.1347
top-100	0.3	0.1235	0.0050	0.1479	0.0867	0.0088	0.1300
top-100	0.1	0.1713	0.0035	0.1886	0.0904	0.0089	0.1342
last-4	0.5	0.0903	0.0077	**0.1280**	0.0870	0.0081	**0.1266**
last-4	0.3	0.2791	0.0119	0.3379	0.2886	0.0094	0.3352
last-4	0.1	0.6528	0.0112	0.7135	0.7274	0.0131	0.7914

Figure 7.2 illustrates the semantic adaptive tracking on TDT-5. The last-4 adaptation is inferior to no-adapt basic run except for the mid-range of the curve. The spot for minimum detection cost pushes into the rectangle bounded by 10% miss rate and 1% false-alarm rate. Whether it is the more temporally sparse topics, larger corpus, or lengthier documents, the TDT-5 performance of last-4 approach deteriorates quickly as θ_{adapt} descends below 0.5.

The top-100 is a more consistent approach for adaptivity than last-4, although there is no clear difference between the two. There is a place for further investigation of values forof m. Another interesting direction would be to maintain multiple centroids for a topic as if to address "topic threading" [107]. Last-m approach could be used to recognize and maintain related yet slightly divergent aspects of the same topic.

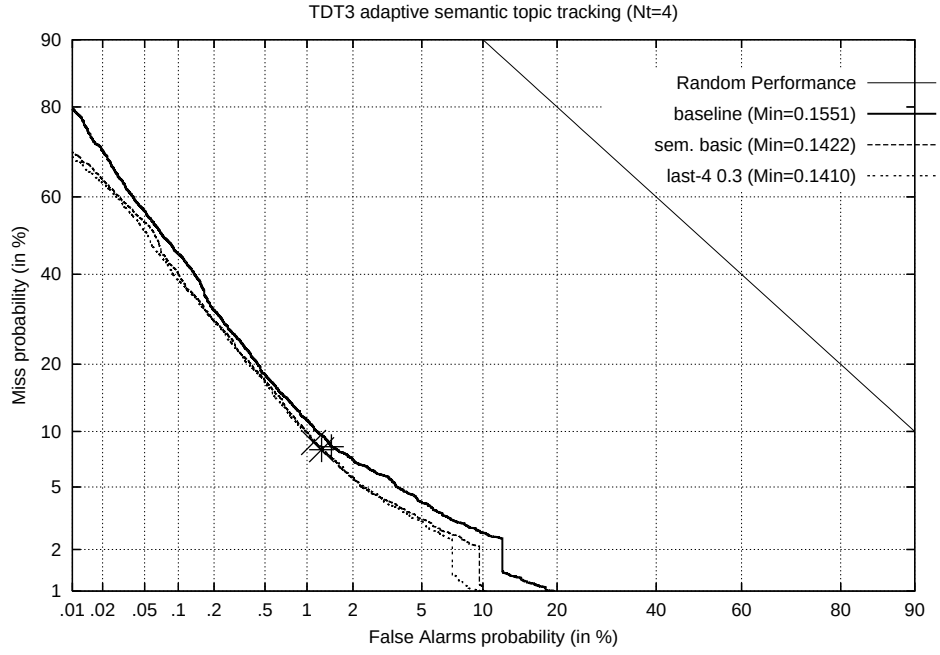

Figure 7.1: The topic-weighted DET curves of TDT3 adaptive semantic topic tracking.

7.2.2 Temporal tracking

The Davidsonian indexing system Σ_{david} from Section 6.2.2 employs six semantic classes, cosine coefficient combined with a temporal context similarity and InQuery term weights. Its score model uses all-subsets kernel with $\theta_{elim} = 0.06$ and low cost-factor. Adaptation parameters are the same as above: $N_t = 4$, $m = 4$ for last-m approach, and $n = 100$ for top-n approach.

The adaptation results for temporal topic tracking are listed in Table 7.2. Introducing temporal information into adaptive tracking does have an impact here and there, but it is not a consistent trend. In TDT-2 and TDT-5 the no-adapt method yields the best detection cost, and in TDT-3 no-adapt loses by a small margin.

In the poorly performing adaptation runs, it is the miss rate that increases, which means adaptation loses relevant documents that the static version is able to capture. Unsurprisingly, this is more a problem of last-4 approach: the adaptation is too greedy, and the topic representation con-

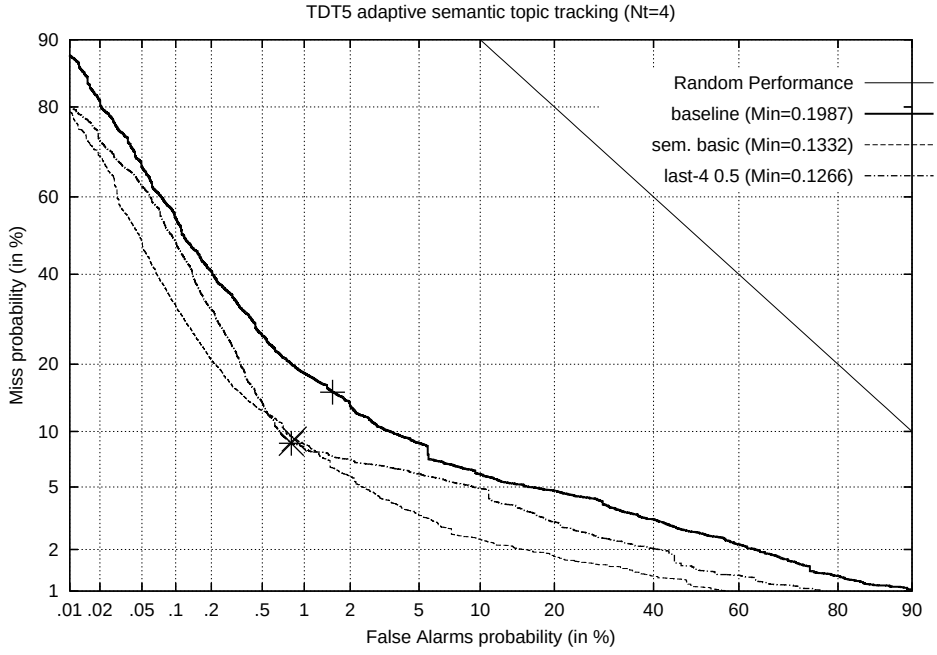

Figure 7.2: The topic-weighted DET curves of TDT5 adaptive semantic topic tracking.

forms to the new information. As the adaptation threshold is lowered, the topic representation is affected by similar but non-relevant documents leading to more non-relevant documents found similar. It increases false-alarm rate, but also miss-rate: the centroid is taken over by non-relevant documents, so the relevant are not found similar anymore.

The last-4 DET curves of Figures 7.3 and 7.4 conform to the same shape as the curve in Figure 7.2. The performance starts erode quickly once the miss rate reaches down to 7-8%. This L-shape of DET curves may have something to do with eliminative score models. Moving down the curve means the miss rate decreases and recall improves. This is achieved by lowering the decision threshold for the ultimate scores. But with eliminative model, the scoring function starts to work quite differently once the x_0 scores go below the elimination threshold θ_{elim} which in these runs was set to 0.06. There is a great volume of documents with $x_0 \in [0, 6)$, none of which are scored with semantic classes present (Equation 5.18). Possibly the outcome is erosion in the effectiveness.

Table 7.2: The adaptive temporal ($N_t = 4$, Σ_{david}, all-l, 0.06) topic tracking topic-weighted results. The column 'adapt type' specifies type of adaptation and θ_{adapt} the adaptation threshold.

		Topic-Weighted at θ			Minimum Topic-Weighted		
adapt type	θ_{adapt}	*P(miss)*	*P(fa)*	$(C_{det})_{norm}$	*P(miss)*	*P(fa)*	$(C_{det})_{norm}$
TDT2							
noadapt	-	0.813	0.0122	**0.1409**	0.0630	0.0137	**0.1303**
top-100	0.5	0.1174	0.0063	0.1500	0.0889	0.0102	0.1391
top-100	0.3	0.1357	0.0059	0.1661	0.0958	0.0073	0.1317
top-100	0.1	0.1515	0.0041	0.1716	0.0795	0.0107	0.1336
last-4	0.5	0.1138	0.0136	0.1803	0.1164	0.0064	0.1495
last-4	0.3	0.2069	0.0026	0.1937	0.1376	0.0062	0.1695
last-4	0.1	0.7272	0.0225	0.8351	0.5906	0.0110	0.6460
TDT3							
noadapt	-	0.0842	0.0113	**0.1398**	0.0849	0.0107	0.1376
top-100	0.5	0.0863	0.0145	0.1575	0.0863	0.0105	0.1373
top-100	0.3	0.1084	0.0109	0.1618	0.1012	0.0087	0.1432
top-100	0.1	0.1376	0.0123	0.1978	0.1834	0.0058	0.2119
last-4	0.5	0.0878	0.0148	0.1604	0.0870	0.0104	**0.1372**
last-4	0.3	0.1546	0.0088	0.1976	0.1049	0.0108	0.1572
last-4	0.1	0.5404	0.0451	0.7613	0.6840	0.0152	0.7584
TDT5							
noadapt	-	0.1069	0.0121	0.1660	0.0865	0.0087	**0.1250**
top-100	0.5	0.0923	0.0077	0.1299	0.0913	0.0077	0.1293
top-100	0.3	0.1084	0.0057	**0.1364**	0.0782	0.0106	0.1301
top-100	0.1	0.1713	0.0035	0.1886	0.1543	0.0046	0.1769
last-4	0.5	0.1138	0.0136	0.1803	0.0693	0.0122	0.1290
last-4	0.3	0.2069	0.0026	0.1937	0.1770	0.0134	0.2428
last-4	0.1	0.7272	0.0225	0.8351	0.5906	0.0110	0.6460

7.3 Conclusions

In this chapter we explored adaptive topic tracking. Adaptiveness is reacting to changes in the data. In case of topic tracking, it boils down to re-building the topic centroid vector upon encountering a document that yields sufficiently high similarity score. For re-building the centroids, we employed two approaches. One chooses the n best terms from the pool of terms of the centroid's documents, and the other focuses the term selection on last m documents and discards the older ones.

We ran experiments for basic semantic signature as well as for Davidsonian indexing signature. Overall, adaptiveness seems to enhance the effectiveness somewhat, but not significantly. The basic semantic run seems to accommodate for adaptivity slightly better than the temporal counterpart. In addition, being more conservative, the top-n approach tends to yield better results that last-m approach, especially with temporal information present.

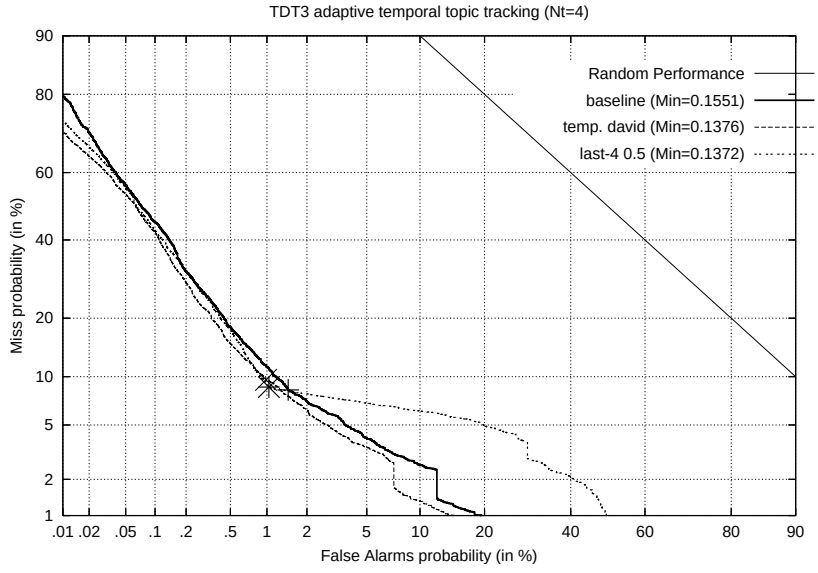

Figure 7.3: The topic-weighted DET curves of TDT3 adaptive temporal topic tracking.

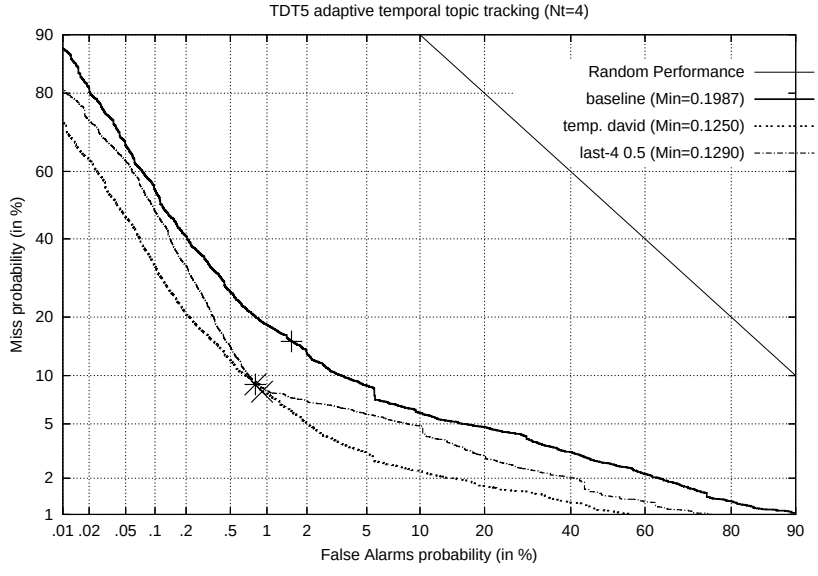

Figure 7.4: The topic-weighted DET curves of TDT5 adaptive temporal topic tracking.

Chapter 8

Conclusion

Topic detection and tracking is a body of information retrieval research that focuses on news events. It develops methods and evaluation metrics for the automatic organization of news data. This thesis has explored techniques for topic tracking and first-story detection that make use of semantic classes, i.e., groups of words with similar meaning, and temporal information. The aim has been to build topic representations that are capable of identifying event-based topics and making the distinction between documents discussing similar and same events.

In the introduction, we declared four contributions. The first contribution was carried out in the conceptual analysis of events and topics in Chapter 4. The purpose was to straighten out the misleading and partly erroneous definitions.

We then presented a simple baseline system using vector-space models. The baseline experiments outlined some of the problems characteristic to TDT: large data volume and small number of on-topic documents, problems arising from automatic translation and automatic speech recognition, and the difficulty in making the distinction between similar events and same events.

Our second contribution was the document similarity framework using semantic classes in Chapter 5. The idea was to detach the ideas about document similarity from direct comparison of intersecting terms, and instead combine external information into the comparison process. We split the term-space into groups of terms and identified term-term correlations to be used in comparing terms in each class. We presented similarity coefficients to be used in comparing semantic class vectors. In the similarity of hands framework, the documents are compared semantic class-wise with respective similarity coefficients, and the result is a vector of scores. This score vector is translated into a single decision score with a score model. For this

purpose, we employed support vector machines. We employed Resnik term-term correlation in comparison of locations with some success. Comparing named entities with asymmetric overlap was found useful in first-story detection. In topic tracking the gains in effectiveness were significant. The improvement in first-story detection was more modest.

Our third contribution was the integration of temporal information into the document similarity. In Chapter 6, we presented an approach to recognition of temporal expressions using finite-state automata. We formalized the recognized expressions by the virtue of calendar algebra. The formalized expressions were ultimately intervals on a global timeline. We then introduced Davidsonian indexing that records the temporal context for each occurrence of each term in each document. The temporal context of a term is defined by the temporal expression in the same sub-clause or sentence, or, in case there is none, of the previous sentences, or, in case there are still none, by publication date. The document similarity using this kind of indexing compares temporal contexts of matching terms. The further the temporal contexts of the terms are apart, the more the term-term correlation is penalized for those terms. The temporal approach improved topic tracking effectiveness over the semantic class approach in almost all experiments. The benefits were less obvious in first-story detection, although the temporal approach yielded the best result with the largest evaluation corpus.

Finally, our fourth contribution dealt with adaptive topic tracking. In Chapter 7, we outlined the basic ideas of adaptive systems, and presented two strategies for the re-building of topic centroids. The first selects n best terms from the pool of all documents compiled into the centroid. The second strategy focused on m latest documents drawing the terms for topic centroid from only those. We then experimented adaptive tracking with semantic and temporal systems. Overall, adaptiveness increased the effectiveness somewhat, but there was no break-through. The top-n strategy is more conservative and does yield better overall performance than the more aggressive last-m strategy.

When comparing our topic tracking results with the previous work (Table 4.8), we are humbled by the spectacular performance of CMU's Rocchio system. The use of negative evidence seem to make all the difference. It is something that could be integrated to similarity of hands, too. A Rocchio-like score model would simply combine the different scores. As to results, we think we have presented a proof-of-concept for both the similarity of hands and the Davidsonian indexing.

In first-story detection the results are less cheerful. The list of results of

Table 4.10 places our systems in the bottom half. Curiously, somewhat similar approaches using different kinds of evidence and combining them with a classifier have shown considerably better effectiveness than we achieved. It may be that our FSD system is geared towards topic tracking and as such does not work well in high recall retrieval like FSD.

Our work presented here could be characterized as exploratory, and our exploration covered but a modest area with many questions remaining open. There are many places for improvement. Since the semantic classes can be virtually anything, there are virtually infinite possibilities in combining ontologies, coefficients and score models. And the efforts need not concentrate on topic detection and tracking. The similarity of hands and the temporal contexts can be applied to other information retrieval problems relating to web search, digital libraries or question answering, for instance.

References

[1] *The Topic Detection and Tracking Phase 2 Evaluation Plan*, 1998. National Institute of Standards and Technology (NIST).

[2] TDT-1999 benchmark evaluation official results. National Institute of Standards and Technology FTP site, 1999. ftp://jaguar.ncsl.nist.gov/tdt/tdt3, last visited 2008-12-15.

[3] TDT-2000 benchmark evaluation official results. National Institute of Standards and Technology FTP site, 2000. ftp://jaguar.ncsl.nist.gov/tdt/tdt2000, last visited 2008-12-15.

[4] TDT-2001 benchmark evaluation official results. National Institute of Standards and Technology FTP site, 2001. ftp://jaguar.ncsl.nist.gov/tdt/tdt2001, last visited 2008-12-15.

[5] *The 2004 Topic Detection and Tracking Task Definition and Evaluation Plan*, 2004. National Institute of Standards and Technology (NIST).

[6] TDT-2004 benchmark evaluation official results. National Institute of Standards and Technology FTP site, 2004. ftp://jaguar.ncsl.nist.gov/tdt/tdt2004, last visited 2008-12-15.

[7] TDT5 Topics and Annotations, 2006. http://www.ldc.upenn.edu/Catalog/docs/LDC2006T19 /README, last visited 2009-02-27.

[8] J. Allan. Detection as Multi-Topic Tracking. *Information Retrieval*, 5(2–3):139–157, 2002.

[9] J. Allan. Introduction to topic detection and tracking. In J. Allan, editor, *Topic Detection and Tracking – Event-based Information Organization*, pages 1–16. Kluwer Academic Publisher, Norvell, MA, USA, 2002.

[10] J. Allan, editor. *Topic Detection and Tracking: Event-based Information Organization.* Kluwer Academic Publishers, Norvell, MA, USA, 2002.

[11] J. Allan, J. Carbonell, G. Doddington, J. Yamron, and Y. Yang. Topic detection and tracking pilot study: Final report. In *Proceedings of Broadcast News Transcription and Understanding Workshop*, pages 194–218, Lansdowne, VA, 1998. NIST.

[12] J. Allan, R. Gupta, and V. Khandelal. Temporal summaries of news topics. In *Proceedings of the 24rd annual international ACM SIGIR conference on Research and Development in Information Retrieval*, pages 10–18, New York, NY, USA, 2001. ACM Press.

[13] J. Allan, H. Jin, M. Rajman, C. Wayne, D. Gildea, V. Lavrenko, R. Hoberman, and D. Caputo. Topic-based novelty detection. Technical report, Center for Language and Speech Processing, Johns Hopkins University, 1999. Summer Workshop Final Report.

[14] J. Allan, V. Lavrenko, D. Frey, and V. Khandelval. UMass at TDT 2000. In *Proceedings of Topic Detection and Tracking Workshop (TDT-2000)*, Gaithersburg, MD, USA, 2000.

[15] J. Allan, V. Lavrenko, and H. Jin. Comparing effectiveness in TDT and IR. Technical Report IR-197, Department of Computer Science, University of Massachusetts, 2000.

[16] J. Allan, V. Lavrenko, and H. Jin. First story detection in TDT is hard. In *Proceedings of the 9th international Conference on Information and Knowledge Management (CIKM)*, pages 374–381, New York, NY, USA, 2000. ACM Press.

[17] J. Allan, V. Lavrenko, D. Malin, and R. Swan. Detections, bounds, and timelines: UMass and TDT-3. In *Proceedings of Topic Detection and Tracking Workshop*, pages 167–174, 2000.

[18] J. Allan, V. Lavrenko, and R. Nallapati. UMass at TDT 2002. In *Proceedings of Topic Detection and Tracking Workshop (TDT-2002)*, Gaithersburg, MD, USA, 2002.

[19] J. Allan, V. Lavrenko, and R. Papka. Event tracking. Technical Report IR – 128, Department of Computer Science, University of Massachusetts, 1998.

[20] J. Allan, V. Lavrenko, and R. Swan. Explorations within topic tracking and detection. In J. Allan, editor, *Topic Detection and Tracking – Event-based Information Organization*, pages 197–224. Kluwer Academic Publisher, Norvell, MA, USA, 2002.

[21] J. Allan, R. Papka, and V. Lavrenko. On-line new event detection and tracking. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 37–45, New York, NY, USA, 1998. ACM Press.

[22] O. Alonso and M. Gertz. Clustering of search results using temporal attributes. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 597–598, New York, NY, USA, 2006. ACM.

[23] O. Alonso, M. Gertz, and R. Baeza-Yates. Search results using timeline visualizations. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 908–908, New York, NY, USA, 2007. ACM.

[24] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press, New York, NY, USA, 1999.

[25] J. Barnes. *Early Greek Philosophy*. Penguin Books, London, UK, 1987.

[26] K. G. Barnhurst and D. Mutz. American journalism and the decline in event-centered reporting. *Journal of Communication*, 47(4):27–53, 1997.

[27] N. J. Belkin and W. B. Croft. Information filtering and information retrieval: two sides of the same coin? *Communications of ACM*, 35(12):29–38, 1992.

[28] A. Bell. News stories as narratives. In A. Jaworski and N. Coupland, editors, *The Discourse Reader*, pages 231–251. Routledge, London, UK, 1999.

[29] T. Brants, F. Chen, and A. Farahat. A system for new event detection. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 330–337, New York, NY, USA, 2003. ACM Press.

[30] R. K. Braun and R. Kaneshiro. Exploiting Topic Pragmatics for New Event Detection in TDT-2004. In *TDT-2003, the Sixth Topic Detection and Tracking Workshop*. NIST, 2003.

[31] R. K. Braun and R. Kaneshiro. Exploiting topic pragmatics for new event detection in TDT-2004. In *TDT-2004, the Seventh Topic Detection and Tracking Workshop*. NIST, 2004.

[32] J. P. Callan, W. B. Croft, and J. Broglio. TREC and TIPSTER experiments with INQUERY. *Information Processing and Management*, 31(3):327–343, 1995.

[33] J. Carbonell, Y. Yang, J. Lafferty, R. D. Brown, T. Pierce, and X. Liu. CMU report on TDT-2: Segmentation, detection and tracking. In *Proceedings of the DARPA Broadcast News Workshop*, pages 117–120. NIST, 1999.

[34] D. Carr. Narrative and the real world: An argument for continuity. *History & Theory*, 25(2):117–131, 1986.

[35] F. Chen, A. Farahat, and T. Brants. Story link detection and new event detection are asymmetric. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 13–15, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

[36] H.-H. Chen and L.-W. Ku. A NLP & IR approach to topic detection. In J. Allan, editor, *Topic Detection and Tracking – Event-based Information Organization*, pages 243–264. Kluwer Academic Publisher, Norvell, MA, USA, 2002.

[37] C. Cieri. Multiple annotations of reusable data resources: Corpora for topic detection and tracking. In *Actes 5ième Journées Internationales d'Analyse Statistique des Données Textuelles (JADT)*, 2000.

[38] C. Cieri, S. Strassel, D. Graff, N. Martey, K. Rennert, and M. Liberman. Corpora for topic detection and tracking. In J. Allan, editor, *Topic Detection and Tracking – Event-based Information Organization*, pages 33–66. Kluwer Academic Publisher, Norvell, MA, USA, 2002.

[39] M. Connell, S. Cronen-Townsend, A. Feng, F. Feng, G. Kumaran, and H. Raghavan. UMass TDT 2003 research summary. Technical

Report CIIR Technical Report, Department of Computer Science, University of Massachusetts, 2003.

[40] M. Connell, A. Feng, G. Kumaran, H. Raghavan, C. Shah, and J. Allan. UMass at TDT-2004. In *TDT-2004, the Seventh Topic Detection and Tracking Workshop*. NIST, 2004.

[41] S. Cottle. The production of news formats: determinants of mediated public contestation. *Media, Culture & Society*, 17(2):275–291, 1995.

[42] J. Cowie and W. Lehnert. Information extraction. *Commun. ACM*, 39(1):80–91, 1996.

[43] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, Cambridge, UK, 2000.

[44] W. B. Croft, S. Cronen-Townsend, and V. Lavrenko. Relevance feedback and personalization: A language modeling perspective. In *Proceedings of the DELOS-NSF Workshop on Personalization and Recommender Systems in Digital Libraries*, pages 49–54, 2001.

[45] D. Cutting, D. Karger, J. Pedersen, and J. Tukey. 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*, pages 318–329, Copenhague, Denmark, 1992. ACM Press.

[46] D. Davidson. The logical form of action sentences. In N. Rescher, editor, *The Logic of Decision and Action*, pages 81–95. University of Pittsburgh Press, 1967. Reprinted in D. Davidson, *Essays on Actions and Events*, Oxford University Press, New York, NY, USA, 1980, pp. 105–122.

[47] D. Davidson. The individuation of events. In N. Rescher and D. Reidel, editors, *Essays in Honor of Carl G. Hempel*, pages 216–234. D. Reidel Publishing, Co., 1969. Reprinted in D. Davidson, *Essays on Actions and Events*, Oxford University Press, New York, NY, USA, 1980, pp. 163–180.

[48] D. Davidson. Events as particulars. *Noûs*, 4(1):25–32, 1970. Reprinted in D. Davidson, *Essays on Actions and Events*, Oxford University Press, New York, NY, USA, 1980, pp. 181–187.

[49] D. Eichmann, M. Ruiz, P. Srinivasan, N. Street, C. Culy, and F. Menczer. Cluster-based approach to tracking, detection and segmentation of broadcast news. In *Proceedings of DARPA Broadcast News Workshop*, pages 69–75. Morgan Kaufmann, 1999.

[50] D. Eichmann and P. Srinivasan. A cluster-based approach to broadcast news. In J. Allan, editor, *Topic Detection and Tracking – Event-based Information Organization*, pages 149–174. Kluwer Academic Publisher, Norvell, MA, USA, 2002.

[51] T. Elsayed, D. W. Oard, and D. Doermann. TDT-2004: Adaptive Topic Tracking at Maryland. In *TDT-2004, the Seventh Topic Detection and Tracking Workshop*. NIST, 2004.

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

[53] J. R. Finkel, T. Grenager, and C. Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, pages 363–370, 2005.

[54] J. Fiscus, G. Doddington, and G. Kuhn. Topic detection and tracking evaluation overview. In J. Allan, editor, *Topic Detection and Tracking – Event-based Information Organization*, pages 17–31. Kluwer Academic Publisher, Norvell, MA, USA, 2002.

[55] R. Fowler. *Language in the News: Discourse and Ideology in the Press*. Routledge, London, UK, 1991.

[56] M. Franz, T. Ward, J. S. McCarley, and W.-J. Zhu. Unsupervised and supervised clustering for topic tracking. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 310–317, New York, NY, USA, 2001. ACM Press.

[57] G. Frege. On sense and reference. In P. Geach and M. Black, editors, *Translations from the Philosophical writings of Gottlob Frege*, pages 59–78. Basil Blackwell, Oxford, UK, 2nd edition, 1960. The original Über Sinn und Bedeutung was published in *Zeitschrift für Philosophie und philosophische Kritik*, Vol. 100, 1892, pp. 25–50. Translation by M. Black.

[58] F. Fukumoto and Y. Suzuki. Event tracking based on domain dependency. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 57–64, New York, NY, USA, 2000. ACM Press.

[59] P. Ganesan, H. Garcia-Molina, and J. Widom. Exploiting hierarchical domain structure to compute similarity. *ACM Transactions on Information Systems*, 21(1):64–93, 2003.

[60] H. J. Gans. *Deciding What's News*. Random House, New York, NY, USA, 1979.

[61] D. J. Gerner, P. A. Schrodt, R. A. Francisco, and J. L. Weddle. Machine coding of event data using regional and international sources. *International Studies Quarterly*, 38(1):91–119, 1994.

[62] Glasgow University Media Group. *Bad News*, volume 1. Routledge & Kegan Paul, London, UK, 1976.

[63] I. A. Goralwalla, Y. Leontiev, M. T. Özsu, D. Szafron, and C. Combi. Temporal granularity: Completing the puzzle. *Journal of Intelligent Information Systems*, 16(1):41–63, 2001.

[64] I. Hacking. *Historical Ontology*. Harvard University Press, Cambridge, Massachusetts, 2002.

[65] T. Harcup and D. O'Neill. What is news? Galtung and Ruge revisited. *Journalism Studies*, 2(2):261–280, 2001.

[66] V. Hatzivassiloglou, L. Gravano, and A. Maganti. An investigation of linguistic features and clustering algorithms for topical document clustering. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 224–231, New York, NY, USA, 2000. ACM Press.

[67] D. He, H. R. Park, G. C. Murray, M. Subotin, and D. W. Oard. TDT-2002 Topic Tracking at Maryland: First Experiments with the Lemur Toolkit. Technical Report LAMP-TR-099,CS-TR-4454,UMIACS-TR-2003-24, University of Maryland, College Park, February 2003.

[68] T. Järvinen and P. Tapanainen. A Dependency Parser for English. Technical Report TR–1, Department of General Linguistics, University of Helsinki, 1997.

[69] H. Jin, R. Schwartz, S. Sista, and F. Walls. Topic tracking for radio, tv broadcast, and newswire. In *Proceedings of DARPA Broadcast News Workshop*. Morgan Kaufmann, 1999.

[70] Y. Jin, S. H. Myaeng, and Y. Jung. Use of place information for improved event tracking. *Information Processing and Management*, 43(2):365–378, 2007.

[71] T. Joachims. *Learning to Classify Text Using Support Vector Machines*. Kluwer Academic Publishers, Boston, 2002.

[72] D. Jurafsky and J. H. Martin. *Speech and Language Processing: An Introduction to Natural Language*. Prentice Hall, Upper Saddle River, NJ, USA, 2000.

[73] A. Kent, M. M. Berry, F. U. Leuhrs, and J. W. Perry. Machine literature searching VIII: Operational criteria for designing information retrieval systems. *American Documentation*, 6(2):93–101, 1955.

[74] P. Kim and S. H. Myaeng. Usefulness of temporal information automatically extracted from news articles for topic tracking. *ACM Transactions on Asian Language Information Processing (TALIP)*, 3(4):227–242, 2004.

[75] D. B. Koen and W. Bender. Time frames: Temporal augmentation of the news. *IBM systems journal*, 39(3&4):597–616, 2000.

[76] G. Kumaran and J. Allan. Text classification and named entities for new event detection. In *Proceedings of the 27th annual international conference on Research and development in information retrieval*, pages 297–304, New York, NY, USA, 2004. ACM Press.

[77] G. Kumaran and J. Allan. Using names and topics for new event detection. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 121–128, Morristown, NJ, USA, 2005. Association for Computational Linguistics.

[78] W. Lam, H. M. L. Meng, K. L. Wong, and J. C. H. Yen. Using contextual analysis for news event detection. *International Journal of Intelligent Systems*, 16(4):525–546, 2001.

[79] L. S. Larkey, F. Feng, M. Connell, and V. Lavrenko. Language-specific models in multilingual topic tracking. In *SIGIR '04: Proceedings of*

the 27th annual international ACM SIGIR conference on Research and development in information retrieval, pages 402–409, New York, NY, USA, 2004. ACM.

[80] V. Lavrenko, J. Allan, E. DeGuzman, D. LaFlamme, V. Pollard, and S. Thomas. Relevance Models for Topic Detection and Tracking. In *Proceedings of Human Language Technology Conference (HLT)*, pages 104–110, 2002.

[81] V. Lavrenko and W. B. Croft. Relevance based language models. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 120–127, New York, NY, USA, 2001. ACM.

[82] L. Lee. On the effectiveness of the skew divergence for statistical language analysis. In *Artificial Intelligence and Statistics*, pages 65–72, 2001.

[83] T. Leek, R. Schwartz, and S. Sista. Probabilistic Approaches to Topic Detection and Tracking. In J. Allan, editor, *Topic Detection and Tracking – Event-based Information Organization*, pages 67–84. Kluwer Academic Publisher, Norvell, MA, USA, 2002.

[84] N. Lester and H. E. Williams. TDT2002 Topic Tracking at RMIT University. In *The Topic Detection and Tracking (TDT) Workshop*, 2002.

[85] B. Li, W. Li, and Q. Lu. Topic tracking with time granularity reasoning. *ACM Transactions on Asian Language Information Processing (TALIP)*, 5(4):388–412, 2006.

[86] X. Liu and W. B. Croft. Statistical Language Modeling for Information Retrieval. *The Annual Review of Information Science and Technology*, 39:3–31, 2004.

[87] Y.-Y. Lo and J.-L. Gauvain. The LIMSI Topic Tracking System for TDT-2001. In *Topic Detection and Tracking Evaluation workshop (TDT 2001)*, Gaithersburg, MD, USA, 2001. NIST.

[88] Y.-Y. Lo and J.-L. Gauvain. The LIMSI Topic Tracking System for TDT 2002. In *Topic Detection and Tracking Evaluation workshop (TDT 2002)*, Gaithersburg, MD, USA, 2002. NIST.

[89] Y.-Y. Lo and J.-L. Gauvain. Tracking topics in broadcast news data. In *ISCA Workshop on Multilingual Spoken Document Retrieval*, pages 43–48, Hong Kong, 2003.

[90] S. A. Lowe. The Beta-Binomial Mixture Model and its Application to TDT Tracking and Detection. In *Proceedings of DARPA Broadcast News Workshop*, pages 127–131. Morgan Kaufmann, 1999.

[91] J. Makkonen. Investigations on event evolution in TDT. In *Proceedings of Student Workshop of Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, pages 43–48, Edmonton, Canada, 2003. ACL.

[92] J. Makkonen and H. Ahonen-Myka. Utilizing temporal information in topic detection and tracking. In T. Koch and I. T. Solvberg, editors, *Proceedings of 7th European Conference on Digital Libraries (ECDL 2003)*, pages 393–404, Trondheim, Norway, 2003. Springer-Verlag.

[93] J. Makkonen, H. Ahonen-Myka, and M. Salmenkivi. Applying semantic classes in event detection and tracking. In R. Sangal and S. M. Bendre, editors, *Proceedings of International Conference on Natural Language Processing (ICON 2002)*, pages 175–183, Mumbai, India, 2002.

[94] J. Makkonen, H. Ahonen-Myka, and M. Salmenkivi. Topic detection and tracking with spatio-temporal evidence. In F. Sebastiani, editor, *Proceedings of 25th European Conference on Information Retrieval Research (ECIR 2003)*, pages 251–265, Pisa, Italy, 2003. Springer-Verlag.

[95] J. Makkonen, H. Ahonen-Myka, and M. Salmenkivi. Simple semantics in topic detection and tracking. *Information Retrieval*, 7(3–4):347–368, 2004.

[96] J. Mandel. *The Statistical Analysis of Experimental Data*. John Wiley & Sons, New York, 1964. Reprinted by Dover Publications, New York, 1984.

[97] I. Mani and G. Wilson. Robust temporal processing of news. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL 2000)*, pages 69–76, Hong Kong, 2000. ACL.

[98] R. Manmatha, A. Feng, and J. Allan. A critical examination of TDT's cost function. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 403–404, New York, NY, USA, 2002. ACM Press.

[99] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

[100] C. D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, USA, 1999.

[101] R. K. Manoff. Writing the news (by telling a story). In R. K. Manoff and M. Schudson, editors, *Reading the News*, pages 197–230. Pantheon Books, New York, NY, USA, 1986.

[102] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki. The DET curve in assessment of detection task performance. In G. Kokkinakis, N. Fakotakis, and E. Dermatas, editors, *Proceedings of EuroSpeech'97, the 5th European Conference on Speech Communication and Technology*, volume 4, pages 1895–1898. ESCA, 1997.

[103] B. Martins, M. J. Silva, and L. Andrade. Indexing and ranking in Geo-IR systems. In *GIR '05: Proceedings of the 2005 workshop on Geographic information retrieval*, pages 31–34, New York, NY, USA, 2005. ACM.

[104] G. A. Miller. WordNet: A Lexical Database for English. *Communications of ACM*, 38(11):39–41, 1995.

[105] R. Nallapati. Semantic language models for topic detection and tracking. In *Proceedings of HLT-NAACL Student Workshop*, pages 1–6, 2003.

[106] R. Nallapati and J. Allan. Capturing term dependencies using a language model based on sentence trees. In *Proceedings of the eleventh international conference on Information and knowledge management*, pages 383–390. ACM Press, 2002.

[107] R. Nallapati, A. Feng, F. Peng, and J. Allan. Event threading within news topics. In *CIKM '04: Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 446–453, New York, NY, USA, 2004. ACM.

[108] S. Nunes, C. Ribeiro, and G. David. Use of temporal expressions in web search. In *Proceedings of ECIR-2008, 30th European Conference on IR Research*, pages 580–584, Glasgow, UK, 2008.

[109] H. M. Pachter. Defining an event. *Social Research*, 41(3):439–466, 1974.

[110] R. Papka. *On-line New Event Detection, Clustering and Tracking*. PhD thesis, Department of Computer Science, University of Massachusetts, 1999.

[111] R. Papka, J. Allan, and V. Lavrenko. UMASS Approaches to Detection and Tracking. In *Proceedings of the DARPA Broadcast News Workshop*, pages 111–116. Morgan Kaufmann, 1999.

[112] F. Pianesi and A. C. Varzi. Events and event talk: An introduction. In J. Higginbotham, F. Pianesi, and A. C. Varzi, editors, *Speaking of Events*, pages 3–47. Oxford University Press, New York, NY, USA, 2000.

[113] A. Pons-Porrata, R. Berlanga-Llavori, and J. Ruiz-Shulcloper. Detecting events and topics by using temporal references. In *Proceedings of IBERAMIA-2002, 8th Ibero-American Conference on Artificial Intelligence*, pages 11–20. Springer, 2002.

[114] A. Pons-Porrata, R. Berlanga-Llavori, and J. Ruiz-Shulcloper. Temporal-semantic clustering of newspaper articles for event detection. In *Proceedings of PRIS-2002, 2nd International Workshop on Pattern Recognition in Information Systems*, pages 104–113, 2002.

[115] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 275–281. ACM Press, 1998.

[116] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[117] H. Reichenbach. *Elements of Symbolic Logic*. The Free Press, New York, NY, USA, 1966.

[118] P. Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 448–453. Morgan Kauffman, 1995.

[119] S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford. Okapi at TREC-3. In *Proceedings of the third Text REtrieval Conference (TREC-3)*, pages 109–126, 1995.

[120] J. J. Rocchio. Relevance Feedback in Information Retrieval. In G. Salton, editor, *The SMART Retrieval System: Experiments in Automatic Document Processing*, pages 313–323. Prentice-Hall, Englewood Cliffs, NJ, 1971.

[121] S. Russell and P. Norvig. *Artificial Intelligence – A Modern Approach*. Prentice Hall, Upper Saddle River, NJ, USA, 1995.

[122] G. Salton. *Automatic Information Organization and Retrieval*. McGraw-Hill, New York, NY, USA, 1968.

[123] G. Salton. *Automatic Text Processing – The Transformation, Analysis, and Retrieval of Information by Computer*. Addison–Wesley, Reading (MA), 1989.

[124] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Communications of ACM*, 18(11):613–620, 1975.

[125] T. Saracevic. Relevance: A Review of and a framework for the Thinking on the Notion in Information Science. *Journal of the American Society for Information Science*, 26:321–343, 1975. Reprinted in K. Spark Jones and P. Willett (eds.), *Readings in information retrieval*. Morgan Kaufmann Publishers, San Francisco, CA, 1997, 143–165.

[126] F. Schilder and C. Habel. From temporal expressions to temporal information: Semantic tagging of news messages. In *Proceedings of ACL-2001 Workshop on Temporal and Spatial Information Processing*, pages 65–72, Toulouse, France, 2001. ACL.

[127] M. Schudson. Deadlines, datelines, and history. In R. K. Manoff and M. Schudson, editors, *Reading the News*, pages 79–108. Pantheon Books, New York, NY, USA, 1986.

[128] J. M. Schultz and M. Y. Liberman. Topic detection and tracking using idf-weighted cosine coefficient. In *Proceedings of DARPA Broadcast News Workshop*, pages 189–192. Morgan Kaufmann, 1999.

[129] J. M. Schultz and M. Y. Liberman. Towards a Universal Dictionary for Multi-language IR Applications. In J. Allan, editor, *Topic Detection and Tracking – Event-based Information Organization*, pages 225–242. Kluwer Academic Publisher, Norvell, MA, USA, 2002.

[130] A. Setzer. *Temporal Information in Newswire Articles: An Annotation Scheme and Corpus Study.* PhD thesis, University of Sheffield, UK, 2001.

[131] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis.* Cambridge University Press, Cambridge, UK, 2004.

[132] L. V. Sigal. Sources of news. In R. K. Manoff and M. Schudson, editors, *Reading the News*, pages 9–37. Pantheon Books, New York, NY, USA, 1986.

[133] D. A. Smith. Detecting events with date and place information in unstructured text. In *Proceedings of JCDL-02, the 2nd Joint Conference on Digital Libraries*, pages 191–196, 2002.

[134] M. D. Smucker, J. Allan, and B. Carterette. A comparison of statistical significance tests for information retrieval evaluation. In *CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 623–632, New York, NY, USA, 2007. ACM.

[135] R. R. Sokal and P. H. A. Sneath. *Principles of Numerical Taxonomy.* W. H. Freeman, San Francisco, CA, USA, 1963.

[136] K. Spark Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21, 1972.

[137] M. Spitters and W. Kraaij. A language modeling approach to tracking news events. In *Topic Detection and Tracking Evaluation workshop (TDT 2000)*, Gaithersburg, MD, USA, 2000. NIST.

[138] N. Stokes. *Applications of Lexical Cohesion Analysis in the Topic Detection and Tracking Domain.* PhD thesis, Department of Computer Science, National University of Ireland, Dublin, 2004.

[139] N. Stokes and J. Carthy. First story detection using a composite document representation. In *Proceedings of Human Language Technology Conference (HLT-2001)*, pages 134–141, 2001.

[140] R. Swan and J. Allan. Automatic generation of overview timelines. In *SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 49–56, New York, NY, USA, 2000. ACM.

[141] J. A. Swets. Information retrieval systems. *Science*, 141(3577):245–250, 1963.

[142] J. A. Swets. Effectiveness of information retrieval methods. *American Documentation*, 20(1):72–89, 1969. Reprinted in John A. Swets, *Signal Detection Theory and ROC in Psychology and Diagnostics: Collected Papers*. Lawrence Erlbaum, Mahwah, NJ, USA, 1996, 205–233.

[143] M. J. Toolan. *Narrative: A Critical Linguistic Introduction*. Routledge, New York, NY, USA, 1988.

[144] D. Trieschnigg and W. Kraaij. TNO hierarchical topic detection report at TDT 2004. In *TDT-2004, the Seventh Topic Detection and Tracking Workshop*. NIST, 2004.

[145] A. Tversky. Features of similarity. *Psychological Review*, 84(4):327–352, 1977.

[146] T. A. van Dijk. *News as Discourse*. Lawrence Erlbaum Associates, Hillsdale, NJ, USA, 1988.

[147] P. van Mulbregt, I. Carp, L. Gillick, S. Lowe, and J. Yamron. Text segmentation and topic tracking on broadcast news via a hidden markov model approach. In *Proceedings of 5th Intl. Conference on Spoken Language Processing (ICSLP-98)*, pages 2519–2522, 1998.

[148] C. J. van Rijsbergen. *Information Retrieval*. Butterworths, London, UK, 2nd edition, 1979.

[149] E. M. Voorhees. The philosophy of information retrieval evaluation. In C. Peters, M. Braschler, J. Gonzalo, and M. Kluck, editors, *Proceedings of the Second Workshop of the Cross-Language Evaluation Forum (CLEF-2001)*, pages 355–370. Springer, 2001.

[150] H. White. The value of narrativity in the representation of reality. In W. J. T. Mitchell, editor, *On Narrative*, pages 1–23. The University of Chicago Press, Chicago, IL, USA, 1981.

[151] I. H. Witten, A. Moffat, and T. C. Bell. *Managing Gigabytes: Compressing and Indexing Documents and Images*. Morgan Kaufmann Publishers, San Francisco, CA, USA, second edition, 1999.

[152] L. Wittgenstein. *Filosofisia tutkimuksia*. WSOY, Juva, 1996. Translation from German *Philosophische Untersuchungen* (1953) by Heikki Nyman (1970,1983).

[153] J. Yamron, S. Knecht, and P. van Mulbregt. Dragon's Tracking and Detection Systems for the TDT2000 evaluation. In *Proceedings of Topic Detection and Tracking Workshop (TDT-2000)*, pages 75–79. NIST, Gaithersburg, MD, USA, 2000.

[154] J. P. Yamron, I. Carp, L. Gillick, S. Lowe, and P. V. Mulbregt. Topic tracking in a news stream. In *Proceedings of DARPA Broadcast News Workshop*, pages 133–136. Morgan Kaufmann, 1999.

[155] J. P. Yamron, L. Gillick, P. van Mulbregt, and S. Knecht. Statistical models of topical content. In J. Allan, editor, *Topic Detection and Tracking – Event-based Information Organization*, pages 115–134. Kluwer Academic Publisher, Norvell, MA, USA, 2002.

[156] Y. Yang, J. Carbonell, R. Brown, J. Lafferty, T. Pierce, and T. Ault. Multi-strategy learning for TDT. In J. Allan, editor, *Topic Detection and Tracking – Event-based Information Organization*, pages 85–114. Kluwer Academic Publisher, Norvell, MA, USA, 2002.

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

[158] Y. Yang, T. Pierce, and J. Carbonell. A study on retrospective and on-line event detection. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 28–36, New York, NY, USA, 1998. ACM Press.

[159] Y. Yang, S. Yoo, J. Zhang, and B. Kisiel. Robustness of adaptive filtering methods in a cross-benchmark evaluation. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 98–105, New York, NY, USA, 2005. ACM.

[160] Y. Yang, J. Zhang, J. Carbonell, and C. Jin. Topic-conditioned novelty detection. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 688–693, New York, NY, USA, 2002. ACM Press.

[161] J. Zhang, Z. Ghahramani, and Y. Yang. A probabilistic model for online document clustering with application to novelty detection. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *NIPS '04: Proceedings*

of Eighteenth Annual Conference on Advances in Neural Information Processing Systems (NIPS-2004), pages 1617–1624, Cambridge, MA, 2005. MIT Press.

[162] J. Zhang and Y. Yang. CMU at TDT-2004 – novelty detection. Workshop Presentation, 2004. http://www.nist.gov/speech/tests/tdt/2004/papers/CMU-NED-TDT2004.ppt, last visited 2009-02-28.

[163] K. Zhang, J. Li, and G. Wu. New event detection based on indexing-tree and named entity. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 215–222, New York, NY, USA, 2007. ACM.