TOPIC DETECTION AND TRACKING

Event-based Information Organization

edited by
James Allan

SPRINGER SCIENCE+BUSINESS MEDIA, LLC

TOPIC DETECTION AND TRACKING

Event-based Information Organization

THE KLUWER INTERNATIONAL SERIES ON INFORMATION RETRIEVAL

Series Editor

W. Bruce Croft
University of Massachusetts, Amherst

Also in the Series:

MULTIMEDIA INFORMATION RETRIEVAL: ***Content-Based Information Retrieval from Large Text and Audio Databases,*** *by Peter Schäuble;* ISBN: 0-7923-9899-8

INFORMATION RETRIEVAL SYSTEMS: ***Theory and Implementation,*** *by Gerald Kowalski;* ISBN: 0-7923-9926-9

CROSS-LANGUAGE INFORMATION RETRIEVAL, *edited by Gregory Grefenstette;* ISBN: 0-7923-8122-X

TEXT RETRIEVAL AND FILTERING: ***Analytic Models of Performance,*** *by Robert M. Losee;* ISBN: 0-7923-8177-7

INFORMATION RETRIEVAL: UNCERTAINTY AND LOGICS: ***Advanced Models for the Representation and Retrieval of Information,*** *by Fabio Crestani, Mounia Lalmas, and Cornelis Joost van Rijsbergen;* ISBN: 0-7923-8302-8

DOCUMENT COMPUTING: ***Technologies for Managing Electronic Document Collections,*** *by Ross Wilkinson, Timothy Arnold-Moore, Michael Fuller, Ron Sacks-Davis, James Thom, and Justin Zobel;* ISBN: 0-7923-8357-5

AUTOMATIC INDEXING AND ABSTRACTING OF DOCUMENT TEXTS, *by Marie-Francine Moens;* ISBN 0-7923-7793-1

ADVANCES IN INFORMATIONAL RETRIEVAL: ***Recent Research from the Center for Intelligent Information Retrieval,*** *by W. Bruce Croft;* ISBN 0-7923-7812-1

INFORMATION RETRIEVAL SYSTEMS: ***Theory and Implementation,*** **Second Edition,** *by Gerald J. Kowalski and Mark T. Maybury;* ISBN: 0-7923-7924-1

PERSPECTIVES ON CONTENT-BASED MULTIMEDIA SYSTEMS, *by Jian Kang Wu; Mohan S. Kankanhalli;Joo-Hwee Lim;Dezhong Hong;* ISBN: 0-7923-7944-6

MINING THE WORLD WIDE WEB: ***An Information Search Approach,*** *by George Chang, Marcus J. Healey, James A. M. McHugh, Jason T. L. Wang;* ISBN: 0-7923-7349-9

INTEGRATED REGION-BASED IMAGE RETRIEVAL, *by James Z. Wang;* ISBN: 0-7923-7350-2

TOPIC DETECTION AND TRACKING

Event-based Information Organization

edited by

James Allan
University of Massachusetts at Amherst

SPRINGER SCIENCE+BUSINESS MEDIA, LLC

ISBN 978-1-4613-5311-9 ISBN 978-1-4615-0933-2 (eBook)
DOI 10.1007/978-1-4615-0933-2

Library of Congress Cataloging-in-Publication Data

Originally published by Kluwer Academic Publishers in 2002
Softcover reprint of the hardcover 1st edition 2002

Printed on acid-free paper.

Contents

Preface ix

1
Introduction to Topic Detection and Tracking 1
James Allan
1 Introduction 1
2 TDT tasks 3
3 History of TDT 7
4 TDT 1999 and TDT 2000 10
5 The Future of TDT 13

2
Topic Detection and Tracking Evaluation Overview 17
Jonathan G. Fiscus and George R. Doddington
1 Introduction 17
2 TDT Definitions: Stories, Events, and Topics 18
3 TDT Corpora 19
4 Evaluation Methodology 20
5 Task Definitions 25
6 Summary 30

3
Corpora for Topic Detection and Tracking 33
Christopher Cieri, Stephanie Strassel, David Graff, Nii Martey, Kara Rennert, and Mark Liberman
1 Introduction 33
2 Overview of TDT Corpus Development 35
3 Collection of Raw Data 36
4 Transcription 38
5 Story Segmentation 39
6 Topic Definition 42
7 Topic Annotation 45
8 Corpus Formats 54
9 Some Properties of the Corpus 61
10 Conclusion 64

4
Probabilistic Approaches to Topic Detection and Tracking 67
Tim Leek, Richard Schwartz, and Srinivasa Sista
1 Introduction 67
2 Core TDT Technologies 68
3 Corpus Processing 75
4 Tracking 75
5 Detection 77
6 Crosslingual TDT 80
7 Conclusions and Future Work 81
8 Acknowledgments 82

5
Multi-strategy Learning for TDT 85
Yiming Yang, Jaime Carbonell, Ralf Brown, John Lafferty, Thomas Pierce, and Thomas Ault
1 Introduction 85
2 Segmentation 87
3 Topic and Event Tracking 88
4 Topic Detection 96
5 First Story Detection 99
6 Story Link Detection 101
7 Multilingual TDT 107
8 Concluding Remarks 111

6
Statistical Models of Topical Content 115
J.P. Yamron, L. Gillick, P. van Mulbregt, and S. Knecht
1 Introduction 115
2 Models of Story Generation 117
3 Tracking Systems 120
4 Detection System 128
5 Summary 132

7
Segmentation and Detection at IBM 135
S. Dharanipragada, M. Franz, J.S. McCarley, T. Ward, and W.-J. Zhu
1 Story Segmentation 135
2 Topic Detection 142
3 Acknowledgements 147

8
A Cluster-Based Approach to Broadcast News 149
David Eichmann and Padmini Srinivasan
1 Introduction 149
2 Segmentation 152
3 Detection 154
4 Tracking 163
5 Acknowledgements 173

9
Signal Boosting for Translingual Topic Tracking 175
Gina-Anne Levow and Douglas W. Oard
1 Introduction 176
2 The Signal-to-Noise Perspective 177
3 Topic Tracking System Architecture 178
4 Contrastive Conditions 184
5 Conclusions and Future Work 191
6 Acknowledgments 194

10
Explorations Within Topic Tracking and Detection 197
James Allan, Victor Lavrenko, and Russell Swan
1 Introduction 197
2 Basic System 198
3 Tracking 203
4 Cluster Detection 205
5 First Story Detection 208
6 Link Detection 208
7 Bounds on Effectiveness 216
8 Automatic Timeline Generation 219
9 Conclusions 222

11
Towards a "Universal Dictionary" for Multi-Language IR Applications 225
J. Michael Schultz and Mark Y. Liberman
1 Introduction 225
2 Our TDT tracking algorithm 229
3 The "Universal Dictionary" experiment 236
4 Conclusions and Directions for Future Work 239

12
An NLP & IR Approach to Topic Detection 243
Hsin-Hsi Chen and Lun-Wei Ku
1 Introduction 243
2 General System Framework 245
3 Representation of News Stories and Topics 246
4 Similarity and Interpretation of a Two-threshold Method 248
5 Multilingual Topic Detection 250
6 Development Experiments 256
7 Evaluation 259
8 Discussion 261
9 Concluding Remarks and Future Works 262

Index 265

Preface

The purpose of this book is to provide a record of the state of the art in Topic Detection and Tracking (TDT) in a single place. Research in TDT has been going on for about five years, and publications related to it are scattered all over the place as technical reports, unpublished manuscripts, or in numerous conference proceedings. The third and fourth in a series of on-going TDT evaluations marked a turning point in the research. As such, it provides an excellent time to pause, review the state of the art, gather lessons learned, and describe the open challenges.

This book is a collection of technical papers. As such, its primary audience is researchers interested in the the current state of TDT research, researchers who hope to leverage that work so that their own efforts can avoid pointless duplication and false starts. It might also point them in the direction of interesting unsolved problems within the area. The book is also of interest to practitioners in fields that are related to TDT—e.g., Information Retrieval, Automatic Speech Recognition, Machine Learning, Information Extraction, and so on. In those cases, TDT may provide a rich application domain for their own research, or it might address similar enough problems that some lessons learned can be tweaked slightly to answer—perhaps partially—questions in their own field.

The first three chapters of this book provide an overview of the research program and its evaluation paradigm. Chapter 1, "Introduction to Topic Detection and Tracking" by Allan, provides an overview of the research program, including its history, motivations, and some general results from the four evaluation workshops. Chapter 2, "Topic Detection and Tracking Evaluation Overview" by Fiscus and Doddington, describes the evaluation paradigm in detail, and Chapter 3, "Corpora for Topic Detection and Tracking" by Cieri, Strassel, Graff, Martey, Rennert, and Liberman, lays out how the two major evaluation corpora (TDT-2 and TDT-3), including stories, topics, and relevance judgments, were created.

The rest of the chapters are technical discussions of research carried out in the TDT 1999 and TDT 2000 evaluation workshops, held in March and November of 2000, respectively.

- Chapter 4, "Probabilistic Approaches to Topic Detection and Tracking" by Leek, Schwartz, and Sista, recounts the research of BBN Technologies. This chapter presents and justifies the use of probabilistic statistical techniques for tackling the TDT tasks. The authors also discuss the cross-topic score normalization problem that is critical in TDT, and outline their approaches that can be easily adapted to different domains and languages.

- Chapter 5, "Multi-strategy Learning for Topic Detection and Tracking" by Yang, Carbonell, Brown, Lafferty, Pierce, and Ault, discusses the work done at Carnegie Mellon University's (CMU) Language Technology Institute. This chapter covers a range of research activities carried out at CMU, incorporating results from all five TDT evaluation tasks. One theme running throughout their work is that of using multiple methods for a particular task, and then combining those methods to achieve results better than any individual approach.

- Chapter 6, "Statistical Models of Topical Content" by Yamron, Gillick, van Mulbregt, and Knecht, describes the research carried out at Dragon Systems. In this chapter, the authors discuss their statistical unigram model and contrast it with one based on a beta-binomial distribution. The work was carried out on the tracking and cluster detection tasks.

- Chapter 7, "Segmentation and Detection at IBM" by Dharanipragada, Franz, McCarley, Ward, and Zhu, presents the work at IBM's T.J. Watson Research Center. This chapter shows how decision trees and maximum entropy models can address the Story Segmentation task. It also develops a microcluster approach to the Cluster Detection task and shows how a modestly hierarchical technique can be helpful.

- Chapter 8, "A Cluster-Based Approach to Broadcast News" by Eichmann and Srinivasan, discusses research from the University of Iowa's School of Library and Information Science. This chapter addresses three TDT tasks: segmentation, cluster detection, and tracking. The authors provide extensive analysis of the effect of different approaches on the detection and tracking tasks.

- Chapter 9, "Signal Boosting for Translingual Topic Tracking" by Levow and Oard, recounts the work done at the University of Maryland, College Park. In this chapter, the authors describe their cross-language research model based on the twin goals of decreasing noise from translation errors, and boosting the importance of correctly translated words. The chapter explores and constrasts several different techniques.

- Chapter 10, "Explorations Within Topic Tracking and Detection" by Allan, Lavrenko, and Swan, presents the research from the University of

Massachusetts at Amherst's Center for Intelligent Information Retrieval. This chapter describes the approaches used by the authors for the TDT evaluation. The authors also make the controversial claim that TDT techniques requires substantially different approaches than those being used to date, by arguing that effectiveness has reached an expected maximum. Finally, this chapter includes an overview of an automatic timeline generation system that was inspired by the TDT research program.

- Chapter 11, "Towards a 'Universal Dictionary' for Multi-Language Information Retrieval Applications" by Schultz and Liberman, describes the research carried out at the University of Pennsylvania. In this chapter, Shultz and Liberman tackle the problem of how many words are needed in a dictionary for effective cross-language work. They show through the use of monolingual experiments, that a very small proportion of the words are needed to achieve high quality results, suggesting that word translation dictionaries may not require broad coverage.

- Chapter 12, " An NLP & IR Approach to Topic Detection" by Chen and Ku, describes the work done at the National Taiwan University. This chapter focuses on the cross-language issues within the TDT tasks, and in particular on translating names. The authors also address the problem of modeling topic shifts over time by incorporating the removal of no-longer-mentioned items from the topic models.

JAMES ALLAN

Chapter 1

Introduction to Topic Detection and Tracking

James Allan
Center for Intelligent Information Retrieval
Department of Computer Science
University of Massachusetts
Amherst, MA 01003

Abstract The Topic Detection and Tracking (TDT) research program has been running for five years, starting with a pilot study and including yearly open and competitive evaluations since then. In this chapter we define the basic concepts of TDT and provide historical context for the concepts. In describing the various TDT evaluation tasks and workshops, we provide an overview of the technical approaches that have been used and that have succeeded.

1. Introduction

Topic Detection and Tracking (TDT) is a body of research and an evaluation paradigm that addresses event-based organization of broadcast news. TDT investigation has been carried out over five years by about a dozen academic and industrial research institutions, and explored in the context of four "cooperatively competitive" evaluations sponsored by the U.S. government.

TDT research begins with a constantly arriving stream of text from newswire and from automatic speech-to-text systems that are monitoring selected television, radio, and Web broadcast news shows. Roughly speaking, the goal of TDT is 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.

The initial motivation for research in TDT was to provide a core technology for an envisioned system that would monitor broadcast news and alert an analyst to new and interesting events happening in the world. Analysts are very interested in keeping abreast of new things that happen in the news, but there are currently no highly effective methods for coping with the huge volume of information that arrives daily. In particular, there are no systems that do that by

monitoring the news for reporting of events rather than for news on a particular and broad subject.

That lack raises one of the points that makes TDT interesting: although it is similar in spirit to information filtering and retrieval work done in the past, it is more tightly defined than broad notions of "aboutness" and as a result admits the possibility of applying deeper automated analysis of content to solve the problem. That is, it is interesting partly because there is hope that some language technologies will be fruitful in this domain even though they are not helpful for information retrieval in general.

Within TDT, a topic is defined to be a set of news stories that are strongly related by some seminal real-world event. When a bomb explodes in a building, that is the seminal event that triggers the topic. Any stories that discuss the explosion, or the rescue attempts, the search for perpetrators, or arrests, trials, and so on, are all part of the topic. Stories on another bomb that explodes on the same day somewhere else in the world are not likely to be on the same topic, nor are stories about a heat wave in the northeastern part of the United States. An intuitive definition of a topic comes out of how people think about news reporting. Suppose someone is *uninterested* in some issue in the news: then, loosely speaking, the topic includes everything arising from the seminal event that the person would *not* want to hear on the news.

Much of information organization research deals with topics that arise out of the broader notion of subject—e.g., what is a story about, rather than what event triggered a story. The distinction between the two is not immediately clear, perhaps because events are subjects themselves: any event can be written as a subject,[1] although subjects cannot necessarily be viewed as events. For example, *flowers that grow in shade* is definitely a subject that an information retrieval system could address; however, there is no corresponding event.

This notion of event-based topic is narrower than a subject-based topic; it is built upon its triggering event. The topic begins at a set time, and is probably no longer reported in the news at some point. Because it is a specific event, it happened not only at some particular time, but in a specific location, and usually with an identifiable set of participants. That is, the topic is fairly well delimited in scope and possibly in time.

The temporal nature of topics in TDT is another way in which it differs from subject-based organization, and another point that makes it an interesting research area. Stories about *flowers that grow in the shade* are about the subject forever: relevance to the subject-based topic is not affected by time. Event-based topics such as those in TDT, on the other hand, are temporally anchored: the seminal event occurs at some time. Further, the topics evolve over time

[1] *Oklahoma City bombing* is both an event and a subject.

to include clearly related events that do not appear to be on the same subject. For example, the *Oklahoma City bombing* topic begins with the (incorrect) hypothesis that the explosion was the result of Middle East terrorism come to the United States. Within a few days, that notion completely disappeared from the topic: it was no longer discussed and was, effectively, no longer relevant. Instead, the topic now included stories on the subject of right-wing militia groups: the notion of relevance had shifted over time.

TDT also incorporates cross-media and cross-language components that are typically separated out into different subproblems in other information processing research efforts. As mentioned above, the news stories of interest in TDT come from both newswire and audio sources. The audio track is made available to TDT evaluation participants, but most researchers choose to use the provided automatic speech recognition (ASR) output that is also made available. As a result, all TDT tasks operate within and across "clean" stories from newswire and "degraded" stories from the various audio sources. One advantage of using the provided ASR output is that differences in speech recognition systems do not affect comparison of different TDT approaches at different sites.

In addition, most TDT tasks are defined across languages, with stories arriving intermixed in both English and Mandarin across both text and audio sources. Because few sites have access to machine translation software, the Mandarin stories are also provided to sites after automatic translation by SYSTRAN.[2] Note that Mandarin audio stories are converted to speech by an ASR system and then translated into English by SYSTRAN.

2. TDT tasks

The research program of TDT focuses on five tasks. Each is viewed as a component technology whose solution will help address the broader problem of event-based news organization. The tasks are:

- *Story Segmentation* is the problem of dividing the transcript of a news show into individual stories.
- *First Story Detection* is the problem of recognizing the onset of a new topic in the stream of news stories.
- *Cluster Detection* is the problem of grouping all stories as they arrive, based on the topics they discuss. (For historical reasons, this task is sometimes referred to simply as "detection.")
- *Tracking* requires monitoring the stream of news stories to find additional stories on a topic that was identified using several sample stories.

[2] http://www.systransoft.com

- *Story Link Detection* is the problem of deciding whether two randomly selected stories discuss the same news topic.

All of the tasks are performed across multiple languages and multiple media types (although Segmentation is meaningless for newswire stories, where the news is already broken into stories). There are clear relationships between several of them [Allan et al., 2000], but they require different styles of evaluation and generally encourage different approaches to their solution.

2.1 Story Segmentation

Audio sources of news generally broadcast shows that include a number of stories. The shows themselves rarely provide an obvious break between the stories, although human listeners and viewers can readily distinguish one story from another. Advertisements or commercial breaks—seemingly natural indicators of story boundaries—do not occur between every story, and even occur in the midst of longer stories. Further, it is not even easy in general to distinguish a commercial from the news show, and there are some audio shows that do not have commercials.

The goal of the Story Segmentation task is to take a show of news and to detect the boundaries between stories automatically. The work might be done on the audio source directly [Stolcke et al., 1999], but almost all research has focused on how to do the segmentation using a text transcript of the show—either closed captions or speech recognizer output.

One technique for addressing this problem is to look for changes in the vocabulary that is used [van Mulbregt et al., 1999, Ponte and Croft, 1997]. These approaches tend to have difficulty distinguishing between stories in the same broad area (e.g., domestic politics) and are troubled by the minor problem of being unable to decide where filler information (e.g., "thank you for joining us") should be placed. Other researchers have looked for words, phrases, pauses, or other features that occur near story boundaries, to see if they can find sets of features that reliably distinguish the middle of a story from its beginning or end [Lafferty et al., 1999, Greiff et al., 2000, Dharanipragada et al., 1999], and clustering those segments to find larger story-like units [Eichmann et al., 1999].

Story Segmentation is a pre-task for the remaining four TDT tasks. That is, each of the other tasks is thought of at the story level, so it needs to have stories as its input. The TDT program has attempted to evaluate the impact of poor Story Segmentation on the other tasks, with a range of results. It appears that Segmentation has little effect on the Tracking task, but does more dramatically impact the various Detection tasks.

2.2 First Story Detection

The goal of First Story Detection (FSD) is to recognize when a news topic appears that had not been discussed earlier. A good FSD system would detect that first news story that reports a bomb's explosion, a volcano's eruption, or a brewing political scandal. Technology of this type is of particular interest to information, security, or stock analysts whose job is look for new events that are of significance in their area.

It is important to note that FSD cannot be solved just by looking for cue phrases such as "this just in." Though such phrases clearly indicate that the story has arrived early, a TDT system might have encountered the same topic on a different stream of news that "scooped" the other. Further, "first story" is defined relative to what has been seen. So at the time an FSD system is turned on, the first story might actually be well into the midst of a news topic.

FSD is typically approached by reducing stories to a set of features, either as a vector [Allan et al., 1999] or a probability distribution [Jin et al., 1999]. When a new story arrives, its feature set is compared to those of *all* past stories. If there is sufficient difference the story is marked as a first story; otherwise, not. There is suggestive evidence that this simple approach to FSD will not ever be very effective (because small error probabilities in that comparison compound once the stream has been around a long time) [Allan et al., 2000].

2.3 Cluster Detection

An obvious extension to the FSD task is the cluster detection task. Here the system's job is not just to identify the onset of a new topic in the news, but to cluster stories on the same topic into bins. When a new bin is needed—i.e., when a first story arrives—it must be created. The creation of bins is an unsupervised task: the system has no knowledge in advance of the number of expected bins, when they should be created, or their contents.

The relationship between FSD and clustering is very clear. The major difference in the context of TDT is how they are evaluated. Clustering tasks are penalized slightly for missing the onset of the topic: it matters primarily that they get the bulk of the topic grouped together. In contrast, the FSD task heavily punishes a system for failing to get the beginning of the topic, and does not care at all what happens in the middle of the topic, provided the system does not declare a new topic (i.e., a FSD false alarm).

The most difficult problem in cluster detection has been finding a suitable evaluation. The currently adopted evaluation requires that a system partition the news stories into bins: no story can reside in more than one bin, even though we know that some stories discuss multiple topics. The resulting clusters are compared to the known topic clusters by finding, for each true topic, a system-proposed cluster that matches it well (based on a known cost function). All

other clusters are ignored since the evaluation has no way of knowing whether they were done correctly.

As a result, if a story were predominantly about an unjudged topic and so was correctly placed with that topic, the system would be penalized. Why? Because as far as the evaluation software is concerned, the system *should* have placed the story with the known topic: it has no information about the unknown topic, so cannot possibly expect the story to appear there. Similarly, if a system splits a topic into two large clusters, it is guaranteed to do poorly on that topic. Further, there is no distinction drawn between a system's splitting a topic into two large clusters and the possibly worse situation of a large cluster along with many small clusters: both situations are evaluated based on that large cluster's relationship to the topic. Finally, there is the problem that "topic" is an elusive notion, even though it is carefully defined within TDT. If a system breaks stories about a criminal case into one cluster covering the crime and another cluster covering the trial, has it really done poorly?

These questions have been raised several times and are likely to cause changes in the cluster detection task in the near future. Despite them, progress has been made toward addressing the problem. Most researchers have used an approach similar to that outlined for FSD above. Stories are represented by a set of features. When a new story arrives it is compared to all past stories and assigned to the cluster of the most similar story from the past (i.e., one nearest neighbor). Variations include making the decision based on multiple neighbors, consolidating all stories deemed to be on a particular topic, etc.

2.4 Tracking

The tracking task is similar in spirit to information retrieval's filtering task. The system is provided with a small number of stories (typically from one to four) that are known to be on the same topic, and is then expected to find all other stories on that topic in the stream of arriving news. In its evaluation mode, the system is given no supervision, so it does not know whether it is correctly tracking those arriving stories.

This task has been the most popular task within TDT evaluations, with more sites participating in it than in any other task. Its attractiveness is probably the result of its being similar to an existing task, and possibly also because it is much easier to work with multiple on-topic stories than to try to guess the topic from a single story.

All topics are tracked independently, so the decision about a story on one of the topics cannot be used to affect the decision on another topic. This choice allows a story to be tracked by multiple topics, in contrast to the cluster detection task that explicitly does not allow that.

The approach normally used in tracking is to extract a set of features from the training stories that differentiate it from the much larger set of stories in the past—ones that are much less likely to be on that topic. (There is no guarantee in TDT that the past stories are not also on topic, but it is very unlikely that many stories are on that topic.) When a new story arrives, it is compared to the topic features and if it matches sufficiently, declared to be on topic. Variations on the k-nearest neighbor approach and combinations of approach have been very successful in this area [Yang et al., 2000]. Some work has also been done using decision trees [Carbonell et al., 1999].

Tracking effectiveness is fairly good and the technology is probably ready for deployment in some specific areas. The TDT researchers have set themselves a challenge of reducing the error rate in tracking by a factor of two over the next few years.

2.5 Story Link Detection

The last TDT task is story link detection (SLD). In this task, a system is handed two news stories and is asked to determine whether or not they discuss the same topic. Unlike the other tasks, the utility of this task in its own right is not clear. However, it is an important core technology for all of the other tasks. Clearly, a perfect SLD system could solve tracking, cluster detection, and FSD. (It is even possible to construct a decent segmentation system using SLD, though it would be incredibly inefficient.)

Most TDT approaches depend upon some similarity function that measures whether two stories talk about the same topic. At a minimum, the SLD task makes it possible to compare similarity functions to determine which ones do a better job of distinguishing on- and off-topic pairs.

Unfortunately, SLD has not been strongly adopted by the research community, presumably because its broader applicability is not obvious: it is not clear how a *person* would use this technology. A few sites have provided sample runs for comparison in the evaluation workshops, but little concerted effort to address that task has appeared.

3. History of TDT

Topic Detection and Tracking research has been through a pilot study and three open and competitive evaluations. The major focus of this early work has been extending techniques from related areas (e.g., information retrieval and speech recognition) into the TDT domain. Research on TDT is now pushing into news-specific and event-focused approaches that should ideally be able to improve system effectiveness substantially.

TDT began and has been supported by the U.S. Government's Defense Advanced Research Projects Agency (DARPA), original within its broadcast news

understanding program, and later through the TIDES program (Translingual Information Detection, Extraction, and Summarization). In addition to funding the pilot study and some of the research organizations in future evaluations, the U.S. Government funded the creation of the TDT-2 and TDT-3 evaluation corpora that were used in the TDT evaluations.

The rest of this section briefly discusses what happened in the TDT evaluations that have already occurred—i.e., TDT 1997 (the pilot study), TDT 1998, TDT 1999, and TDT 2000. Note that in some publications "TDT 1998" is referred to as "TDT-2" because that was the second TDT evaluation and because it used the TDT-2 corpus as its evaluation vehicle. Similarly, "TDT 1999" is sometimes listed as "TDT-3." The confusion was cleared up in TDT 2000 since it used the TDT-3 corpus again and not the forthcoming TDT-4 corpus.

3.1 TDT 1997, a pilot study

In 1996 and 1997, researchers from DARPA, Carnegie Mellon University, Dragon Systems, and the University of Massachusetts at Amherst set out to define "Topic Detection and Tracking" (after deciding that was an appropriate name for the technologies) and to develop preliminary technology to address the problems. The goal was to determine the extent to which subject-based technologies from information retrieval could address event-based organization issues.

The group defined four tasks to be part of TDT. The first two, segmentation and tracking, are as defined in Section 2 above. The other two were retrospective detection and on-line detection. The latter task is identical to what is referred to as "cluster detection" above. The former is a variation on cluster detection where the system is permitted to look at the entire set of news stories before beginning the clustering task. The reason for the two versions was to allow comparison between "zero knowledge" and "full knowledge" of the document stream, to see how having no future information degraded performance.

All tasks were conceived as detection tasks and evaluated using miss and false alarm error rates. A Detection Error Tradeoff (DET) plot [Martin et al., 1997] was the primary means of portraying the errors.

The research groups constructed a pilot corpus out of CNN stories taken from Primary Source Media and Reuters news stories acquired from the Linguistic Data Consortium. This corpus of 15,683 news stories covers the news from January through June of 1995. The stories are ordered as they were reported. The corpus also includes relevance judgments for 25 events within the corpus. The events were chosen to represent a range of events that seemed "interesting," to ensure that there would be a fair number of stories on each event in the corpus, and also to cover a range of event classes that are generally recognized as "events." Unusually for IR-related tasks, every story in the collection was

judged against every one of the 25 topics (almost 400,000 judgments). Some additional research was carried out using a small corpus of stories generated by a speech recognition system.

The researchers met regularly throughout that year, rotating between their respective sites: Arlington in October 1996, Amherst in January 1997, Pittsburgh in March, and Boston in May. Results of the pilot study were presented at Harpers Ferry in October of 1997, and again at the Broadcast News Transcription and Understanding Workshop in February 1998 [Allan et al., 1998].

The TDT pilot study was viewed as a success. The problem could be defined in a way that was tractable and that could be evaluated. Over the course of the next year, a full-scale TDT evaluation was begun.

3.2 TDT 1998

The TDT 1998 evaluation included eleven research sites: GTE Internetworking's BBN Technologies, Columbia University, Carnegie Mellon University, Dragon Systems, General Electric, IBM's T.J. Watson Laboratories, SRI International, University of Iowa, University of Massachusetts at Amherst, University of Maryland, and the University of Pennsylvania. The group of researchers met regularly throughout 1998 to define the formal evaluation, the nature of the evaluation corpus, and to discuss research results and progress.

The evaluation tasks were changed primarily by removing "retrospective detection," leaving just segmentation, tracking, and cluster detection. A cost measure that mixed miss and false alarm error rates with respect to a cost values was adopted as the primary evaluation criterion. This single-number measure allowed sites to train their systems to optimize performance for that cost function. The researchers investigated a range of additional measures, but no other measure was adopted broadly.

The TDT-2 corpus was created for this evaluation. It consists of approximately 57,000 stories from newswire, radio, and television. The non-newswire sources, 630 hours of audio, were converted into text using Dragon System's speech recognition system, modified to run at a sixth its previous speeds without any drop in error rate [Gillick et al., 1998]. The stories were bundled into files that contained a half hour of news (approximated for the newswire). The model of news delivery to the system was a file at a time, with decisions on a file required before the next file arrives. A variation on the tasks supported deferral, where TDT decisions on stories could be defined until a large number of files had arrived.

The pilot study created topics carefully but not rigorously. For TDT 1998, the Linguistic Data Consortium (LDC) was contracted to create the TDT-2 corpus [Cieri et al., 1999]. Topic definitions were much more carefully crafted, using rules of interpretation to make it clear which stories were on topic and which

were not. Topic were chosen by randomly sampling stories from the corpus and developing a topic from that random choice. In the end 100 topics were created for the TDT-2 corpus.

The researchers in TDT 1998 met frequently throughout the year, with the final reports being presented as part of the DARPA Broadcast News Workshop in March 1999. The overall conclusions were that tracking and segmentation effectiveness were reasonably good, and perhaps even ready for use in some applications. There was concern expressed about evaluation measures for segmentation, however, making those conclusions tentative. Substantial room for improvement remained for the other three tasks.

4. TDT 1999 and TDT 2000

This book is primarily a discussion of the TDT 1999 and TDT 2000 evaluation workshops. Both workshops used the TDT-3 corpus for research evaluation. The major characteristics of the corpus are:

1 The stories came from October through December, 1998. Training data came from the TDT-2 collection of stories that spanned January through June of the same year.

2 Two additional English sources were added, and Chinese newswire and audio sources were also added. The Chinese sources are included in TDT-3 in their original language (speech recognition output for audio sources) and a SYSTRAN translation of each story is also provided.

3 The evaluation topic set used for TDT 1999 was engineered to encourage stories of international scope. Those topics required a minimum of four stories in *each* of Chinese *and* English.

4 Another set of evaluation topics was used in TDT 2000, where the minimum story count restriction was lifted. The international scope of the topics was encouraged by seeding half of them from English and half from Chinese documents, but no longer requiring a minimum of four stories in each language. These topics were also annotated differently, using a search-based approach to find on-topic stories rather than by exhaustively considering every story-topic pair as was done before.

Because the research results of these two years are discussed in detail in this book, the remainder of this section just touches on the highlights of each meeting.

4.1 TDT 1999

The TDT 1999 evaluation saw the creation of the three-month TDT-3 corpus for evaluation and the use of the six-month TDT-2 corpus as training. The large

number of training stories helped with the research, but the addition of Chinese news and the change in the nature of the topics (to the international flavor) made some training difficult.

Eleven research groups participated in the evaluation: BBN Technologies, Carnegie Mellon University, Dragon Systems, GE, IBM's T.J. Watson Laboratories, MITRE Corporation, University of Iowa, University of Maryland, University of Massachusetts, University of Pennsylvania, and the National University of Taiwan.

The researchers met only a few times throughout the year. The evaluation had been well established in TDT 1998, and the researchers felt there was little value in meeting more than a couple of times. The final reports were presented at a two-day TDT-only meeting in Vienna, Virginia.

The researchers made progress on segmentation, finding that it was not substantially different in Chinese than it was in English. The largest issue in segmentation appeared to be the quest for useful features that indicate a story boundary. In tracking, the cross-language component provided the most interesting results. The researchers found that score normalization by source and by language was critical, that cross-language tracking caused a 20% drop in effectiveness compared to within-language tracking, that combining several methods was useful, and that there seemed to be little impact caused by using automatically segmented stories rather than manually segmented stories.

In the cluster detection task, the cross-language component failed almost completely: systems generally created clusters of English stories and separate clusters of Mandarin stories. Oddly, systems were not hurt too much if they did not bother to combine the two clusters. There was considerable discussion at the meeting about the nature of topic clusters in TDT and how they should really be evaluated. There were no changes decided upon in the evaluation, but the stage was set to revisit the problem in the future.

In First Story Detection, the few participating sites found the best performance by ignoring clusters that are implicitly created. Instead, the best way to decide if a story is on a new topic is to see if there is any other *single* story that is too similar. If so, the story is not new. Overall effectiveness was nonetheless poor, although one group showed some results that strongly suggested that the FSD performance is as good as can be expected given current approaches to solving the problem [Allan et al., 2000].

The Story Link Detection task received little attention, despite its status as a core technology. Evaluation problems surfaced in this first run of the task, a result of how the pairs of stories were sampled from the collection. Some special SLD-specific annotation was used, but it is not clear how valuable that was.

4.2 TDT 2000

TDT 2000 was the first TDT evaluation supported by the DARPA TIDES program (Translingual Information Detection, Extraction, and Summarization). This cycle of TDT started in March of 2000 and ended in November of the same year, to bring it in line with other TIDES-supported evaluation programs. Because of that short evaluation cycle and minimal funding available, the same TDT-3 evaluation corpus was used, but a new set of evaluation topics was created, this time with a less restrictive international focus. Participating sites were on their honor to use the TDT-2 corpus for training and not to use any information from TDT-3.

In order to make the largest possible strides, the research community chose to focus attention on two tasks. First, Story Link Detection as a core technology of TDT was emphasized, although interest in the task was weak and few sites participated. Second, the tracking task with a *single* positive training instance was highlighted. Most sites participated in that program. The other tasks were still available for evaluation.

The TDT 2000 workshop was held immediately following the TREC evaluation workshop in November 2000, in Gaithersburg, Maryland. The nine participating sites were Chinese University of Hong Kong, Dragon Systems, IBM's T.J. Watson Laboratories, MITRE Corporation, National Taiwan University, Texas A&M University, TNO TPD (Netherlands), University of Iowa, and University of Massachusetts.

For tracking, performance with a single training story was not much different than TDT 1999's results with *four* training stories. The researchers also found that automatic story boundaries had a noticeable impact on effectiveness, in contrast to past results that suggested this was not an issue. Score normalization across topics continued to be a major problem for the sites.

In the cluster detection task, sites found advantages to smoothing incoming stories with information from another corpus, presumably because it provided additional related vocabulary to describe the topic. The cross-language aspect of the problem caused high costs for large topics where it was easier for a system to accidentally split a topic into multiple pieces. Clustering evaluation continued to be a problem, resulting in the recommendation that the evaluation workshop search for a "better" application model.

The only site participating in Story Link Detection showed advantages of applying score normalization based on whether the stories are within or across languages. They also found that it was valuable to smooth each story with related stories from the corpus to increase vocabulary overlap.

5. The Future of TDT

The pilot study and three full-scale evaluations have shown that event-based organization via information retrieval approaches performs reasonably well. There are strong similarities between the notions of "event" and "subject," so it is not surprising that this happens. Much of the research so far on TDT has been tuning parameters to make those techniques slightly more appropriate for events.

The future of TDT evaluations requires moving beyond those approaches, leveraging the aspects of TDT that are different from information retrieval. For example, an event is something that has a specific time, location, and people associated with it. A topic within TDT is actually a collection of inter-related events. Whether an event is part of a topic is determined by a human-generated "rule of interpretation." Techniques that explicitly model any of those aspects of the problem seem likely to do better than techniques that ignore them.

Similarly, the time-based aspect of TDT is different from much other work in information retrieval, and specific handling of that is likely to be helpful. Old events and topics need to be substantially discounted, so there is a lower and lower chance that anything will match them. The focus of a topic can shift over topic, to the extent that there may be very little obvious connection between early and late stories.

TDT research is active and continuing. TDT 2001 is being held in November of that year, with about a dozen participating organization. Plans are already underway for a TDT 2002 evaluation workshop, that will include testing on a newly-created TDT-4 corpus that incorporates a larger number of English and Chinese news sources, with stories spanning the end of 2000 into the beginning of 2001.

TDT-related research also continues independently of the evaluation workshops. The corpora have been used in two summer research workshops at Johns Hopkins University's Center for Language and Speech Processing [Allan et al., 1999, Meng et al., 2001]. The TDT ideas were the inspiration for work on "temporal summarization," summarizing the changes in a news topic from day to day (as opposed to overall, as addressed by most summarization research) [Khandelwal et al., 2001, Allan et al., 2001], and for work on automatic timeline construction [Swan and Allan, 1999, Swan and Allan, 2000]. Also, some early work has been done trying to capture the distinction between topics and events [Fukumoto and Suzuki, 2000].

News and other information is constantly arriving and it is impossible for anyone to keep abreast of everything that is happening without some way to reduce what they have to consider. Technologies such as Topic Detection and Tracking, technologies that seek to impose some order on the inflow of information, technologies that provide a means for people to understand what is

happening and now it is changing—these technologies are a critical component of any solution to the problem of information overload. The research within the TDT evaluation program provides first steps toward addressing those needs.

References

[Allan et al., 1998] Allan, J., Carbonell, J., Doddington, G., Yamron, J., and Yang, Y. (1998). Topic detection and tracking pilot study: Final report. In *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, pages 194–218.

[Allan et al., 2001] Allan, J., Gupta, R., and Khandelwal, V. (2001). Temporal summaries of news topics. In *Proceedings of ACM SIGIR, Research and Development in Information Retrieval*, pages 10–18.

[Allan et al., 1999] Allan, J., Jin, H., Rajman, M., Wayne, C., Gildea, D., Lavrenko, V., Hoberman, R., and Caputo, D. (1999). Topic-based novelty detection: 1999 summer workshop at CLSP, final report. Available at http://www.clsp.jhu.edu/ws99/tdt.

[Allan et al., 2000] Allan, J., Lavrenko, V., and Jin, H. (2000). First story detection in TDT is hard. In *Ninth International Conference on Information Knowledge Management (CIKM'2000)*, Washington, D.C. ACM.

[Carbonell et al., 1999] Carbonell, J., Yang, Y., Lafferty, J., Brown, R. D., Pierce, T., and Liu, X. (1999). CMU report on TDT-2: Segmentation, detection and tracking. In *Proceedings of the DARPA Broadcast News Workshop*, pages 117–120. Morgan Kauffman Publishers.

[Cieri et al., 1999] Cieri, C., Graff, D., Liberman, M., Martey, M., and S. Strassel (1999). The TDT-2 Text and Speech Corpus. In *Proceedings of the DARPA Broadcast News Workshop*, pages 57–60. Morgan Kauffman Publishers.

[Dharanipragada et al., 1999] Dharanipragada, S., Franz, M., McCarley, J., Roukos, S., and Ward, T. (1999). Story segmentation and topic detection in the broadcast news domain. In *Proceedings of the DARPA Broadcast News Workshop*, pages 65–68. Morgan Kauffman Publishers.

[Eichmann et al., 1999] Eichmann, D., Ruiz, M., Srinivasan, P., Street, N., Culy, C., and Menczer, F. (1999). A cluster-based approach to tracking, detection and segmentation of broadcast news. In *Proceedings of the DARPA Broadcast News Workshop*, pages 69–75. Morgan Kauffman Publishers.

[Fukumoto and Suzuki, 2000] Fukumoto, F. and Suzuki, Y. (2000). Event tracking based on domain dependency. In *Proceedings of ACM SIGIR, Research and Development in Information Retrieval*, pages 57–64.

[Gillick et al., 1998] Gillick, L., Ito, Y., Manganaro, L., Newman, M., Scattone, F., Wegmann, S., Yamron, J. P., and Zhan, P. (1998). Dragon systems' automatic transcription of new TDT corpus. In *Proceedings of the Broadcast News Transcription and Understanding Workshop*, pages 219–222. Morgan Kauffman Publishers.

[Greiff et al., 2000] Greiff, W., Morgan, A., Fish, R., Richarsd, M., and Kundu, A. (2000). MITRE TDT-2000 segmentation system. In unpublished TDT 2000 proceedings. Also available at http://www.nist.gov/TDT.

[Jin et al., 1999] Jin, H., Schwartz, R., Sista, S., and Walls, F. (1999). Topic tracking for radio, TV broadcast, and newswire. In *Proceedings of the DARPA Broadcast News Workshop*, pages 199–204. Morgan Kauffman Publishers.

[Khandelwal et al., 2001] Khandelwal, V., Gupta, R., and Allan, J. (2001). An evaluation corpus for temporal summarization. In *Proceedings of the Human Language Technology Conference*, pages 102–106. Morgan Kauffman Publishers.

[Lafferty et al., 1999] Lafferty, J., Beeferman, D., and Berger, A. (1999). Statistical models for text segmentation. *Machine Learning*, 34(1-3):177–210.

[Martin et al., 1997] Martin, A., G. Doddington, T. K., Ordowski, M., and Przybocki, M. (1997). The DET curve in assessment of detection task performance. In *Proceedings of EuroSpeech*, volume 4, pages 1895–1898.

[Meng et al., 2001] Meng, H., Chen, B., Khudanpur, S., Levow, G.-A., Lo, W.-K., Oard, D., Schone, P., Tang, K., Wang, H.-M., and Want, J. (2001). Mandarin-english information (MEI): Investigating translingual speech retrieval. In *Proceedings of the Human Language Technology Conference*, pages 239–245. Morgan Kauffman Publishers.

[Ponte and Croft, 1997] Ponte, J. and Croft, W. (1997). Text segmentation by topic. In *Proceedings of the First European Conference on Research and Advanced Technology for Digital Libraries*, pages 113–125.

[Stolcke et al., 1999] Stolcke, A., Shriberg, E., Hakkani-Tür, D., Tür, G., Rivlin, Z., and Sönmez, K. (1999). Combining words and speech prosody for automatic topic segmentation. In *Proceedings of the DARPA Broadcast News Workshop*, pages 61–64. Morgan Kauffman Publishers.

[Swan and Allan, 1999] Swan, R. and Allan, J. (1999). Extracting significant time varying features from text. In *Eighth International Conference on Information Knowledge Management (CIKM)*, pages 38–45. ACM Press.

[Swan and Allan, 2000] Swan, R. and Allan, J. (2000). Automatic generation of overview timelines. In *Proceedings of ACM SIGIR, Research and Development in Information Retrieval*, pages 49–56.

[van Mulbregt et al., 1999] van Mulbregt, P., Carp, I., Gillick, L., Lowe, S., and Yamron, J. (1999). Segmentation of automatically transcribed broadcast news text. In *Proceedings of the DARPA Broadcast News Workshop*, pages 77–80. Morgan Kauffman Publishers.

[Yang et al., 2000] Yang, Y., Ault, T., Pierce, T., and Lattimer, C. W. (2000). Improving text categorization methods for event tracking. In *Proceedings of ACM SIGIR, Research and Development in Information Retrieval*, pages 65–72.

Chapter 2

Topic Detection and Tracking Evaluation Overview

Jonathan G. Fiscus and George R. Doddington
Information Access Division
Information Technology Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899

Abstract The objective of the Topic Detection and Tracking (TDT) program is to develop technologies that search, organize and structure multilingual, news oriented textual materials from a variety of broadcast news media. This research program uses controlled laboratory simulations of hypothetical systems to test the efficacy of potential technologies, to gauge research progress, and to provide a forum for the exchange of research information. This chapter introduces TDT's evaluation methodology including: the Linguistic Data Consortium's TDT corpora, evaluation metrics used in TDT and the five TDT research tasks: Topic Tracking, Link Detection, Topic Detection, First Story Detection, and Story Segmentation.

1. Introduction

The objective of the Topic Detection and Tracking (TDT) program is to develop technologies that search, organize and structure multilingual, news oriented textual materials from a variety of broadcast news media. This research program uses controlled laboratory simulations of hypothetical systems to test the efficacy of potential technologies, to gauge research progress, and to provide a forum for the exchange of research information.

The TDT program began in 1997 with a pilot study involving a small set of researchers who identified potential technologies for automatically organizing news texts. The community continued to meet and to design the primary components of the project, the experimental research tasks, and the data resources needed for research.

Perhaps the most important concept in TDT is that operational systems of the future will process data continuously as it is collected. Most previous research on text retrieval and information organization has been focused on static, retrospective text archives [1]. In contrast, TDT technologies operate on data collected in real time and from a variety of sources and potentially in a variety of languages.

The second concept fundamental to TDT is the notion of an event, or in TDT parlance a *topic*. In TDT, a topic is defined to be "a seminal event or activity along with all directly related events and activities." Since TDT focuses on processing news data, a natural way to organize news articles is by the reported events.

During the pilot study and intervening years, the community selected and defined five research tasks that simulate deployable TDT systems. The tasks were named, *Topic Tracking*, *Link Detection*, *Topic Detection*, *First Story Detection* and *Story Segmentation*.

The National Institute of Standards and Technology (NIST) has administered three open evaluations of the TDT tasks since 1998 [2,3,9]. The NIST TDT website [4] contains information about the evaluations as well as numerous papers and presentations given at the TDT workshops that NIST held after each evaluation.

The remainder of this chapter discusses details of the TDT program's evaluation infrastructure. There are four more sections in this chapter. First, TDT terminology is discussed; this includes the definition of a story and a topic. Second, the data used for research, the TDT corpora, are introduced. The third section is a brief introduction to detection task evaluations, the evaluation formalism used in TDT. The fourth section contains explanations of each of the evaluation tasks.

2. TDT Definitions: Stories, Events, and Topics

In the course of preparing corpora for the TDT program, the Linguistic Data Consortium (LDC) [5,6] transcribed hundreds of hours audio recordings collected from TV and radio news broadcasts. Since the unit of retrieval for the TDT program is stories, the LDC annotated the broadcasts with story boundaries. To aid the LDC's annotation of story boundaries, the community agreed that a story is "a topically cohesive segment of news that includes two or more declarative independent clauses about a single event." While the definition doesn't address stories that discuss multiple events, which happens frequently in the TDT corpora, the definition enabled the LDC to tag the data with story boundaries with adequate reliability.

The definition of topic has changed over the course of the program. In the TDT pilot study, the notion of a topic was limited to be an "event", meaning something that happens at some specific time and place along with all necessary preconditions and unavoidable consequences. Later, in the second year, the definition of a topic was broadened to include, in addition to the triggering event, other events and activities that are directly related to it. This definition has persisted for the ensuing years. Formally, the TDT definition of a topic is "a seminal event or activity, along with all directly related events and activities." A story is considered "on topic" when it discusses events and activities that are directly connected to that topic's seminal event. Therefore, for example, a story on the search for survivors of an airplane crash, or on the funeral of the crash victims, will be considered a story on the crash event. Obviously there must be limits to this inclusiveness. (For example, stories on FAA repair directives that derive from a crash investigation would not be considered stories on the crash event.) Since definition of a topic's extension to related events is not readily agreed upon, the LDC has created topic annotation guidelines to improve agreement and consistency of topic labelling. [5]

3. TDT Corpora

The LDC provided three corpora to support TDT research [5]: the TDT Pilot corpus, the TDT2 corpus and the TDT3 corpus. These corpora are collections of news, including both text and speech, from a number of sources in both English and Mandarin.

The TDT Pilot corpus contains 26K news stories from the Reuters newswire service and transcripts of CNN broadcasts. The corpus spans the period from July 1, 1994 to June 30, 1995. TDT researchers annotated the corpus with 25 events, (using the initial definition of events).

The TDT2 corpus spans the first six months of 1998 and contains 74K news stories from six English and three Mandarin newswire and broadcast news sources. Newswire data are rendered using the original electronic text, with the addition of consistent SGML tagging to minimize formatting differences among various sources. Radio and television material is rendered as digital audio, as human-generated transcripts, and as mechanically-generated transcripts produced by an Automatic Speech Recognition (ASR) system. In addition to these forms, the Mandarin data has been translated to English using the SYSTRAN translation system. The TDT2 corpus is annotated for 100 topics in English, 20 of which have also been annotated in Mandarin. The LDC also annotated 100 topics is support of the 1999 Johns Hopkins Summer Workshop [4].

The TDT3 corpus spans Oct-Dec 1998 contains 45K news stories from eight English newswire and broadcast news sources, and three Mandarin newswire and broadcast news sources, all of which are organized identically to the TDT2 corpus. There are 240 topics annotated in the TDT3 corpus: 120 topics were judged against the whole corpus (including both English and Mandarin), and 120 have been partially annotated against the English portion of the corpus.

Each story in the TDT2 and TDT3 corpora is tagged according to whether it discusses the defined topics. These story-topic tags are assigned a value of *YES*, if the story discusses the target topic, or *BRIEF* if that discussion comprises less than 10% of the story. Otherwise, the (default) tag value is *NO*.

There were two styles of complete annotation used for topic tagging. For the first style of annotation, the annotators were given a list of 20 topics to annotate at a time. The annotator would read a story and judge whether or not the story discussed any of their 20 topics. While this process was thought to be the best way to annotate TDT data, it was labor intensive. During 2000, the second and more efficient technique called *search-guided* annotation reduced the labor by using a search engine to limit the number of stories an annotator had to read. This protocol gave each annotator a single topic to work on at a time and a relevance-ranked list if stories which he/she read until they reached a point of diminishing return. Early investigations suggest that the latter technique produces better consistency presumably due to a reduced cognitive load.

4. Evaluation Methodology

TDT is a technology research and development program. At the core of the program is the "technology evaluation cycle" employed by DARPA-sponsored R&D programs in the speech field for many years, Figure 1. The cycle essentially has five phases: task definition, system design, system building, system testing, and system refinement. After the refinement, developers re-evaluate their systems in order to assess how the refinements have affected performance. Periodically, (every year for the TDT program,) there is a community-wide technology evaluation that culminates in a meeting to discuss recent research and progress, and possibly modify the task definition.

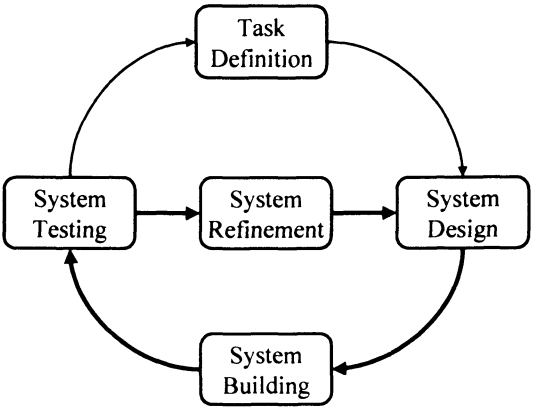


Figure 1. DARPA Evaluation Cycle

This evaluation cycle requires a considerable amount of infrastructure. Not only must the community agree on the evaluation tasks, but they must also agree on the corpora (for training, development and evaluation sets), evaluation metrics, data formats, and system I/O requirements. Consensus on these issues are reached through periodic meetings and codified in the TDT task specification [4]. As the program changes, the task specification codifies in detail the expectations of systems operation. The remainder of this section discusses key components of the task specification.

4.1 TDT Tasks Evaluated as Detection Tasks

All of the TDT tasks are cast as detection tasks. That is, a system is presented with input data and a hypothesis about the data, and the system's task is to decide whether the hypothesis about this data is true. This is called a trial. If the hypothesis is true, the trial is called a target; if not, the trial is called a non-target trial.

A target story can be correctly detected as a target, or a story can be missed in which case the error is called a *missed detection*. A non-target trial can be correctly determined to be a non-target, or it can be falsely detected in which case the error is called a *false alarm*. Table 1 summarizes the contingency matrix of detection system responses.

Along with the actual decisions, a detection system emits a score. The detection decision is based on this score, which indicates how strongly the evidence suggests that the trial is a target trial. While systems are at liberty to construct their own decision score space, the scores must be comparable across topics and corpus, i.e., a score of 1.0 "means" the same for two

different topics, across languages or across sources. Indeed this is a considerable challenge for TDT systems.

		Reference Annotation	
		Target	Non-Target
System Response	YES (a Target)	Correct	*False Alarm*
	NO (Not a Target)	*Missed Detection*	Correct

Table 1. Contingency table of detection system responses

There are two techniques for representing performance based on missed detections and false alarms; the detection cost function (C_{Det}) and the decision error tradeoff curve (DET) curve [7]. The former is a "single number" performance measure that estimates system performance at a particular operational point using the actual decisions (YES/NO), and the latter is a visualization of the tradeoff between missed detections and false alarms using the distribution of decision scores.

Since TDT evaluations use many topics, the global assessment of system performance is accomplished by averaging both the detection cost function and DET curves across topics. In TDT, we call these topic-weighted performance metrics. The major advantage to using the averages is that confidence intervals are trivially established for performance measurements as well as outliers are easily identified. Alternatively, global performance could be assessed using a trial-weighted detection cost function and DET curve. In TDT, this is called a story-weighted measure since the trials are typically decisions based on stories. The disadvantage of a story-weighted measure is that topics with disproportionately large numbers of trials can swamp smaller topics.

The remainder of this section discusses the detection cost function and the DET curve.

4.2 Normalized Detection Cost Function

Detection system performance is characterized in terms of the probabilities of missed detection and false alarm errors (P_{Miss} and P_{Fa}). These error probabilities are linearly combined into a single detection cost, C_{Det}, by assigning costs to missed detection and false alarm errors and specifying an *a priori* probability of a target.

The cost model provides a convenient framework for evaluating systems that exhibit a performance trade off between P_{Miss} and P_{Fa}. Intuitively, when a user employs a searching or filtering technology, they're doing so to reduce

their workload, e.g., you want to find all the stories that discuss an event while not reading millions of stories. For the user, there's a fixed cost for reading a story, and an increased cost for reading a non-target story, since the time spent was wasted. Thus, the detection cost function uses C_{Miss} and C_{Fa} as estimates of these costs. Linearly combining P_{Miss} and P_{Fa} using the assigned costs would be sufficient if the richness of targets and non-targets were identical. However, in TDT, and most other filtering applications, the difference is several orders of magnitude. Therefore, a term is needed to compensate for the difference in target richness, hence P_{Target}. The resulting formula for C_{Det} is

$$C_{Det} = (C_{Miss} * P_{Miss} * P_{Target} + C_{Fa} * P_{Fa} * (1 - P_{Target}))$$

$$P_{Miss} = \#Missed\ Detections\ /\ \#Targets$$
$$P_{Fa} = \#False\ Alarms\ /\ \#Non\text{-}Targets$$

Where

- C_{Miss} and C_{Fa} are the costs of a missed detection and a false alarm respectively, and are pre-specified for the application,
- P_{Miss} and P_{Fa} are the probabilities of a missed detection and a false alarm respectively and are determined by the evaluation results, and
- P_{Target} is the *a priori* probability of finding a target as specified by the application.

For each TDT task, the evaluation specification states C_{Miss} and C_{Fa}. Their values are set using previous experience with detection systems development. For most TDT evaluation tasks, they are set to 10 and 1 respectively. Note that these constants are arbitrarily chosen, and their value is less important than their consistent use. P_{Target} is based on corpus statistics and is a measure of the richness of on-topic stories in the training data. Again, any reasonable choice will suffice as long as the value is used consistently.

While C_{Det} is a convenient measure to assess performance, its dynamic range makes it difficult to interpret, e.g., good performance results in detection costs on the order of 0.001. Therefore, in TDT we use a *Normalized Detection Cost*, or $(C_{Det})_{Norm}$. The goal of normalization is to ground the performance to a more meaningful range. This is accomplished by expanding the dynamic range in the "good performance" range of the scale. To do so, we divide C_{Det} by the minimum expected cost achieved by either answering YES to all decisions or answering NO to all decisions. The resulting normalized cost still has a minimum of zero, but now a cost of 1.0

means a system is doing no better than consistently guessing YES or NO. The derivation of the normalized detection cost formula is as follows:

$$(C_{Det})_{Norm} = C_{Det} / MIN((C_{Miss} * 1.0 * P_{Target} + C_{Fa} * 0.0 * (1 - P_{Target})), (C_{Miss} * 0.0 * P_{Target} + C_{Fa} * 1.0 * (1 - P_{Target})))$$

$$(C_{Det})_{Norm} = C_{Det} / MIN(C_{Miss} * P_{Target}, C_{Fa} * (1 - P_{Target}))$$

4.3 Detection Error Tradeoff Curves

Detection Error Tradeoff (DET) curves are visualizations of the tradeoff between of missed detection (P_{Miss}) rate and the false alarm (P_{Fa}) rate. The curves are constructed by sweeping a threshold through the system's space of decision scores. At each point in the score space, P_{Miss} and P_{Fa} are estimated and plotted as a connected line.

Figure 2 is a DET curve from the 1999 tracking evaluation. The Y-axis is the probability of missed detection and the X-axis is the probability of false alarms. Since missed detections and false alarms are types of errors, improvements in performance will be shown by lines moving closer to the lower left hand corner. Note that the normal deviant scale (expressed as percentages) is used on both axes. The normal deviant scale has advantages over linear scales. It expands the "high performance" region, and resulting straight lines indicate normality of the underlying error distributions of P_{Miss} and P_{Fa}.

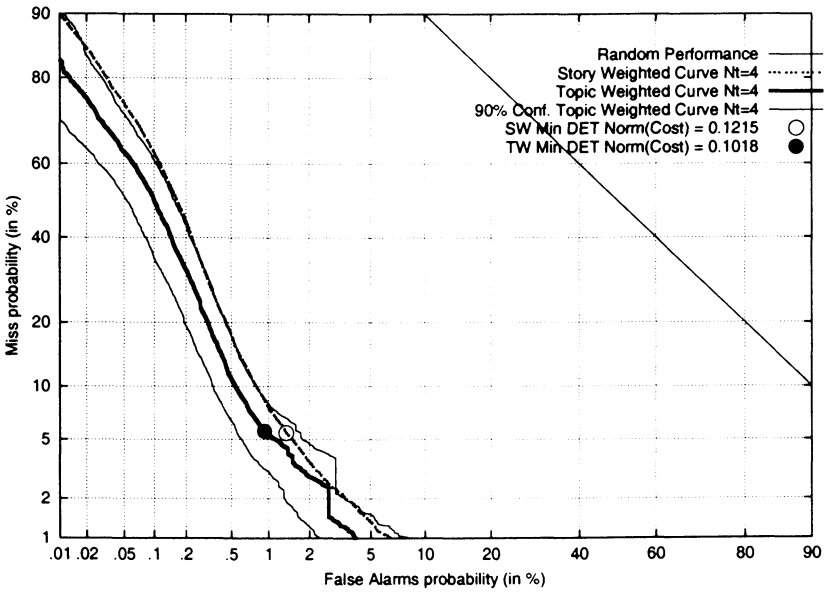


Figure 2. Example DET curve from the 2000 TDT Evaluation

The method described above generates a story-weighted DET curve. Story-weighted DET curves suffer from the same vulnerabilities as story-weighted measures discussed earlier, so TDT uses a topic-weighted DET curve to match the topic-weighted $(C_{Det})_{Norm}$. Topic weighted DET curves are made as follows: sort the stories in order of decision scores separately for each topic. Again, step through the score space, but rather than calculate global P_{Miss} and P_{Fa}, compute the average of P_{Miss} and P_{Fa} across topics. Since means are estimated, variances can also be computed which allows computation of confidence region.

Figure 2 contains both a story-weighted and topic-weighted DET curves. Also presented are the P_{Miss} and P_{Fa}, points corresponding to the minimum in the detection cost function for each curve type. Note the disparity in the story- versus topic-weighted curves. Either technique would be an appropriate means of assigning performance; the benefit to using topic-weighted DET curves is the ability to calculate 90% confidence DET curves and topic-weighted curves have lower variances that story-weighted curves.

5. Task Definitions

There are five evaluation tasks in the TDT program. The tasks can vary in focus and size from hypothetical applications to enabling technologies. In brief, the goal of each of the tasks is:

- Topic Tracking – detect stories that discuss a target topic,
- Link Detection – detect whether a pair of stories discuss the same topic,
- Topic Detection – detect clusters of stories that discuss the same topic,
- First Story Detection – detect the first story that discusses a topic, and
- Story Segmentation – detect story boundaries

5.1 Topic Tracking

TDT topic tracking systems detect stories that discuss a previously known topic. A topic is "known" by its association with stories that discuss it. Tracking systems are given these sets of on-topic stories and a portion of the evaluation corpus to train models on. The systems are tested by their ability to find on-topic stories within the remainder of the corpus.

Developers must adhere to three key system design issues.

First, tracking systems must train and test on each topic independently. Systems cannot make use of any other topic's definition, which would

presumably make the task easier. As a by-product of topic independence, the training epochs, the portion of the evaluation corpus used for training the systems models, differ from topic to topic. Since the number of evaluated stories differ from topic to topic, the topic-weighted detection cost function is the preferred system performance metric. Independence of topic has a major advantage. Since the evaluation protocol creates orthogonal topics, stories that discuss multiple topics are evaluated separately for each topic and thus are handled gracefully.

The second system design issue is decision score normalization across topics. Decision scores should "mean" the same thing across topics, so for instance a decision score of 15.0 for one story and one topic indicates the same amount of evidence supporting an on-topic decision for another story and another topic. Mathematically, not only do the means of the underlying target/non-target decision score distributions have to match, but also the variances. Note that this task would be much simpler if systems were allowed to make use of other evaluation topics for score normalization; however, formulating the task as such makes the systems deal with issues of evidence reliability to some extent.

The third system design issue requires tracking systems to be multilingual. Systems must track topics in all languages within the corpus regardless of all training/test language pairs. No doubt, this is a daunting task and requires considerable infrastructure. To make this task more accessible to small researchers, the evaluation corpus includes English translations for the Mandarin texts.

Tracking systems are evaluated using the topic-weighted normalized cost function and the topic-weighted DET Curve, both of which were described in section 4.

There are many experimental conditions identified in the evaluation plan, each enabling developers and NIST the opportunity to decompose system performance on factors that are thought to affect system performance. The TDT 2000 evaluation plan calls out the following conditions: the number of training stories, the number of negative example training stories, the language of the training stories, the form of the broadcast news data, and reference vs. automatic story boundaries.

5.2 Link Detection

TDT link detection systems detect when a pair of stories discuss the same topic (i.e., the stories are "linked" by a common topic). These systems answer the YES/NO question: "do these two stories discuss the same topic?" and output a decision score that the answer is YES. The actual decisions and

decision scores are used to calculate $(C_{Det})_{Norm}$ and DET curves respectively as described in section 4.

This task can be thought of as a core capability from which topic tracking and topic detection systems can be built. The link detection task is related to topic tracking with one training story, but rather than track the stories through time, the link detection task sub-samples the story space to be more efficient. Otherwise, a system would need to evaluate *N*(N-1)/2* story pairs.

There are advantages to the link detection paradigm. As defined, the task does not require annotator effort to define topics as in topic tracking or topic detection. Performance can be evaluated using human judgements on random story pairs as to whether or not they discuss the same topic without a formal statement of topic. Since the topic space does not need to be organized into orthogonal clusters of stories, handling of stories on multiple topics is a non-issue.

Another advantage to link detection is the ability to separate performance of monolingual and cross-lingual story pairs. Since system judgements on each story pair are made independently of each other, assessing performance based on any division is simply a matter of sub-sampling the story pairs.

The task is more flexible than the tracking task because there are provisions for systems to take advantage of deferral periods, (a specified amount of future data that can be processed before making decisions on the current story).

There are relatively few evaluation conditions defined by the evaluation plan. For the TDT 2000 evaluation, those conditions were the form of the broadcast news data and the deferral period.

5.3 Topic Detection

The topic detection task evaluates technologies that detect novel, previously unknown, topics. As in the tracking task, topics are defined by associating together stories that discuss the topic. However, topic detection systems are not given *a priori* knowledge of the topic. Therefore, systems must embody an understanding of what constitutes a topic, and this understanding must be independent of topic specifics. The task is multilingual and therefore systems must build clusters that span languages.

The systems detect clusters of stories that discuss the same topic. The concept of clustering is easily applied to news stories, but the assessment of performance is difficult because stories frequently discuss multiple topics. This phenomenon not only means the topic clusters are dependent on previously processed stories, but also that decomposition of performance into casual subsets is misleading.

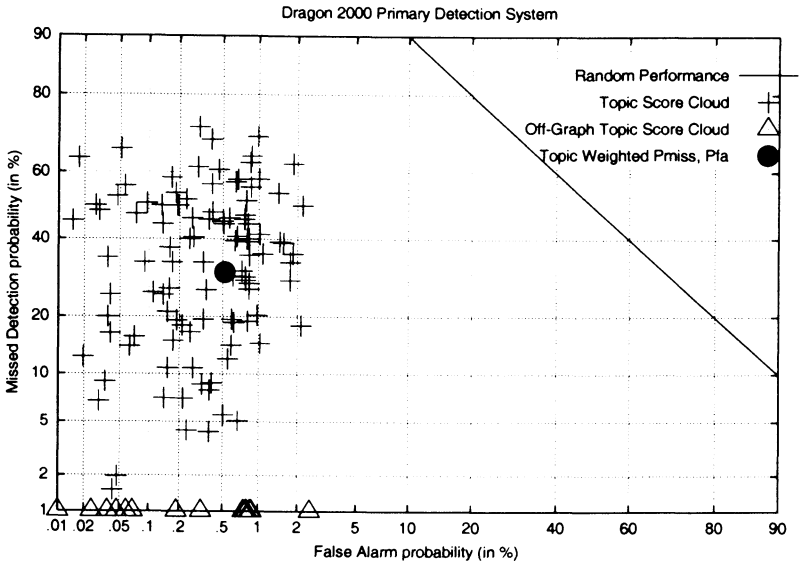


Figure 3. Example DET Cloud from Dragon's 2000 Primary Topic Detection System

The evaluation protocol must deal with the issue of topic independence. The multi-topic stories are declared unscorable even though the systems perform clustering on all test stories. Thus, multiple topic stories may influence a system, but they do not contribute to the error measures.

Performance assessment for topic detection uses the detection cost model but with two variations: P_{Miss} and P_{Fa} are calculated after mapping system-defined topics to reference topic clusters, and DET clouds are used rather than DET curves. P_{Miss} and P_{Fa} are calculated for each reference topic cluster using the system-defined cluster that has the lowest detection cost. This reference to system-defined cluster mapping permits system clusters to "map" to any number reference clusters. This mapping is the least cost mapping; therefore, the reported topic detection scores are the minimum score[1]. Second, DET curves are not used since decision scores are not meaningful in the context of detection systems. Instead, detection performance assessment makes use of DET clouds, i.e., a point for each topic's P_{Miss} and P_{Fa}, are plotted on a DET-scaled graph, see Figure 3. The DET cloud also includes the system's topic-weighted average P_{Miss} and P_{Fa}. As in the other evaluation tasks, stories marked as BRIEF are declared unscoreable and as such are left out of calculations of P_{Miss} and P_{Fa} for the topic.

[1] A globally optimised mapping that would enforce a 1:1 mapping would yield higher measured detection costs. While such an algorithm is straight forward, it is computationally expensive and it could degenerate to a very long search.

There are a number of evaluation conditions defined by the evaluation plan. For the TDT 2000 evaluation, those conditions were the source language (English, Mandarin and both English and Mandarin), the form of the broadcast news transcripts, reference vs. automatic boundaries, and the decision deferral period.

5.4 First Story Detection

The first story detection task (FSD) evaluates technologies that detect the first story to discuss topic. This special case of the topic detection task focuses on the specific aspect of topic detection associated with novel information detection, i.e., knowing when to start a new cluster. The task parameters are essentially the same as topic detection. The real difference is in what the system outputs.

FSD systems output an actual decision, either YES or NO, in response to the question: "does this story discuss a new topic?" and a decision score that the answer is YES. While there are relatively few first stories in a corpus, performance assessment for this task uses DET curves in addition to normalized first story detection cost using the same protocol as defined in the evaluation methodology section.

Like topic detection, the FSD evaluation assumes that first stories always discuss a single topic. The TDT annotations of topics disprove this assumption, so the evaluation ignores first stories that are ambiguous, i.e., stories known to discuss a previously seen topic.

Unlike other tasks, stories labelled as BRIEF mentions of a topic are considered as potential non-first stories. However, they are not used as first story candidates.

For the TDT 2000 evaluation, FSD was strictly an English task. The restriction was a pragmatic decision made by the community to streamline the evaluation. The task has the additional evaluation conditions involving the form of the broadcast news transcripts, reference vs. automatic story segmentation, and decision deferral periods.

5.5 Story Segmentation

The story segmentation task evaluates technologies that detect story changes. The systems segment streams of automatically transcribed text into TDT-style stories. In TDT, a story is a "topically cohesive segment of news that includes two or more declarative independent clauses about a single event." The notion of story explicitly excludes commercials from being stories, and therefore systems are not evaluated on boundaries between consecutive commercials.

In TDT, story segmentation is seen to be an enabling technology since all retrieval is story based. This implies that all automatically transcribed speech data will need to be segmented by stories. As previously discussed, TDT is multilingual; the segmentation task is not an exception. Rather than requiring segmenters to work on English translations of Mandarin texts, segmentation systems work on native orthographies.

Performance assessment of segmentation systems makes use of the detection cost model, but the derivation of the missed detection and false alarm probabilities is quite different compared to the other TDT tasks. System performance is judged by determining how well computed story boundaries agree with reference boundaries. This agreement will be judged with an evaluation interval, nominally 15 seconds, that is swept through the input data. The technique is a derivation of the method proposed by Beeferman, et al. [8] The evaluation interval is chosen to be long enough to include all computed boundaries that might reasonably be associated with a true reference boundary, but short enough to exclude unreasonable associations and multiple reference boundaries (i.e., whole stories).

Evaluation is performed by sweeping the evaluation interval through the input source stream and judging the correctness of the segmentation at each position of the interval:

1. If there is both a computed boundary and a reference boundary within the interval, then segmentation is judged correct.
2. Likewise, if there is neither a computed nor a reference boundary within the interval, then segmentation is judged correct.
3. However, a missed detection is declared if there is no computed boundary within an interval that contains a reference boundary,
4. Moreover, a false alarm is declared when a computed boundary exists within an interval that doesn't contain a reference boundary.

The evaluation conditions for the segmentation tasks are the language of the material, the form for the broadcast news data and the decision deferral period, measured in seconds. Note that the task ignores newswire texts since newswire services routinely include story segmentations.

6. Summary

In this chapter, the Topic Detection and Tracking evaluation methodology was introduced. The TDT evaluation methodology is codified by the TDT corpora and the TDT evaluation specification document. The evaluation specification covers three major topics; structure of the TDT corpora, the

TDT evaluation metrics, and the TDT research tasks: Topic Tracking, Link Detection, Topic Detection, First Story Detection, and Story Segmentation.

References

[1] Voorhees, E, Harman, D., "Overview of the Eighth Text REtrieval Conference (TREC-8)", NIST Special Publication 500-246: The Eighth Text REtrieval Conference (TREC 8)

[2] Fiscus, J., Doddington, G., Garofolo, J., and Martin, A., "NIST's 1998 Topic Detection and Tracking Evaluation (TDT)", Fifth European Conf. On Speech Comm. and Tech., Vol. 4, pp. 247-250.

[3] Fiscus, J., and Doddington, G., Results of the 1999 Topic Detection and Tracking Evaluation in Mandarin and English, 6th International Conference on Spoken Language Processing, October 2000, Beijing China, SS(06)-05, paper 320.

[4] TDT Homepage at the National Institute of Standards and Technology, http://www.nist.gov/TDT

[5] Cieri, C., Graff, D., Libermann, M., Martey, N., Strassel, S., "Large, Multilingual, Broadcast News Corpora for Cooperative Research in Topic Detection and Tracking: The TDT-2 and TDT-3 Corpus Efforts", Second International Conference on Language Resources and Evaluation, 31 May - 2 June, 2000, pp. 925-930.

[6] Linguistic Data Consortium TDT Corpora Homepage, http://www.ldc.upenn.edu/Projects/TDT

[7] Martin, A., Doddington, G., Kamm, T., Ordowski, M., Przybocki, M., "The DET Curve in Assessment of Detection Task Performance", Eurospeech 1997, Proceedings Volume #4 Pages 1895-1898

[8] Beeferman, D., Berger, A., and Lafferty, J., Text Segmentation Using Exponential Models, In Proceedings of the Second Conference on Empirical Methods in Natural Language Processing, pp. 35-46. 1997

[9] TDT 2000 Evaluation Website, (Includes presentations and papers) http://www.nist.gov/TDT/tdt2000

Chapter 3

Corpora for Topic Detection and Tracking*

Christopher Cieri, Stephanie Strassel, David Graff, Nii Martey,
Kara Rennert, and Mark Liberman
Linguistic Data Consortium
University of Pennsylvania
Philadelphia, PA 19104

Abstract The TDT corpora, developed to support the DARPA-sponsored program in Topic Detection and Tracking, combine data collected over a nine month period from 8 English and 3 Chinese sources. The published corpora contain audio, reference text including written news text and transcripts of the broadcast audio, boundary tables segmenting the broadcasts into stories and relevance tables resulting from millions of human judgments. Sections of the corpora have undergone topic-story, first story and story link annotation. Both the TDT-2 and TDT-3 text corpora and the accompanying broadcast audio are now available from the Linguistic Data Consortium. This paper described the raw material collected for the corpora, the annotation of that material to prepare it for research use and the formats in which it is distributed. Special attention is paid to the quality control measures developed for these data sets.

1. Introduction

The TDT Corpora were created to support the DARPA-sponsored common-task research program in Topic Detection and Tracking (TDT). The Topic Detection and Tracking program seeks to develop core technology for a news understanding system capable of processing streams of news in

* The Linguistic Data Consortium's work in building the TDT-2 and TDT-3 corpora was supported in part by grant IRI-9528587 from the Information and Intelligent Systems division of the National Science Foundation.

multiple languages and across multiple media. Processing includes dividing the news streams – newsfeeds or broadcast news programs – into individual stories and categorizing them according to the topic or topics they discuss. The languages could be as diverse as English and Chinese. The media might include broadcast television and radio, newswires, WWW sites, newsgroups, e-mail lists or some future innovation or combination. Specific TDT research tasks include:

- segmentation - divide news streams into individual stories
- topic detection - identify new topics in the news, group stories by topic
- topic tracking - identify all stories that discuss a selected topic
- first story detection - identify the first story to discuss each selected topic
- story link detection – determine whether two stories discuss the same topic

The TDT research program maintains a strong commitment to real world data. Researchers may work directly with the broadcast audio or with text intermediaries created by submitting the audio to speech recognition systems. Where the source language is not English, TDT researchers may work either with the original text or with English translations created by best-of-breed systems. In any case, TDT systems need to perform well despite the impediments of errorful transcription and translation. Research sites are also constrained to use only the linguistic content of the raw material. Formatting information such as the paragraph breaks, headers and slug lines that may appear in newswire are not available during TDT evaluations.

Although the TDT Corpora were created to accommodate the TDT program, they have also been designed with future projects in mind. The published corpora contain a human-readable reference version of each story, a tokenized version with all formatting and extra-linguistic material removed, the original audio of broadcasts and both original and English translations of non-English text. To date, a number of other research programs have used the TDT corpora. These include the TREC-8 Spoken Document Retrieval Evaluation (Garofalo, et. al. 1999), the DARPA sponsored programs in Automatic Content Extraction (NIST, 2000) and Broadcast News Transcription (NIST 1999) and the John's Hopkins CLSP 1999 Summer Workshop on Novel Information Detection (CLSP 1999).

Since 1997, the Topic Detection and Tracking program has sponsored the creation of three corpora that have been used in formal technology evaluations each year since. The TDT Pilot corpus was created by research participants in 1997 and used for the pilot evaluation that year. The TDT Pilot corpus has been used as training material in the 1998-2000 evaluations. The Linguistic Data Consortium created the second and third corpora, TDT-2 and TDT-3, in 1998 and 1999 respectively. TDT-2 provided training and evaluation material for the 1998 evaluation and training material for the 1999

and 2000 evaluations. TDT-3 provided evaluation material in 1999, 2000, and 2001.

This chapter will focus on TDT-3 and the annotations used in the two most recent evaluations. It will describe the processes for collecting, transcribing, segmenting and otherwise annotating that data with special attention paid to the definition of "topic" within TDT and the quality control procedures applied throughout corpus development.

2. Overview of TDT Corpus Development

Before dealing with the individual stages in the development of the TDT corpora, it will be helpful to summarize the process as a whole. LDC begins by collecting the raw material, newswires and other electronic text, broadcast radio and television. The next step is to produce text intermediaries for all audio material. The text intermediaries are used both in subsequent annotation tasks and in evaluation tasks. The text intermediaries are next segmented into individual stories that become the input to all subsequent annotation tasks. Before annotation can begin, LDC annotators select and define topics explicitly according to the procedure described below. With topics in hand, the team annotates the corpus in an attempt to identify all stories that discuss each of the selected topics even if only briefly. There are quality control efforts inserted at several places in the process. Once the corpus has been completely annotated, the text and the tables that encode annotators' judgments are formatted to accommodate the evaluation specification and released to the research community either directly or via the National Institute of Standards and Technology (NIST) who manage TDT evaluations. Figure 1 shows this process graphically.

Why doing annotation?

Much of the discussion in this chapter will focus on Quality Control. In general we will speak of four kinds of quality control. In *Precision QC*, one reviews cases where annotators have asserted that some relationship exists – a phrase begins a new story or a story describes a topic – to confirm the relation. Errors found during Precision QC are *false alarms*. In *Recall QC*, one reviews cases where no relationship was asserted in hopes of finding any *misses*. In *Dual Annotation*, two independent human annotators perform the same task independently. A third annotator reviews the results to measure inter-annotator agreement and remediate discrepancies. Finally in *Adjudication*, human annotators review cases where system outputs disagree with previous annotations to either confirm or overrule the previous judgment. There are obvious interactions among these processes. For example, adjudication may either supplement or replace Precision QC and Recall QC. However, for a project of the scale and scope of TDT, where the

annotation is relatively novel, all four forms of quality control play an important role.

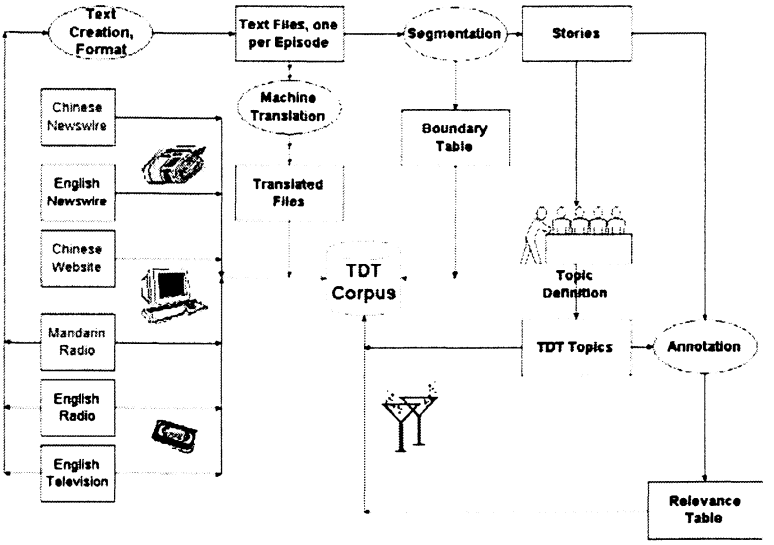


Figure 1. Stages in TDT corpus development

3. Collection of Raw Data

The TDT-2 and TDT-3 corpora each contain multiple sources of data in English and Chinese. To create TDT-2, LDC collected six English and three Chinese sources over the six-month period from January through June 1998. The English sources include daily samplings from two television, two radio and two newswire sources, specifically: ABC *World News Tonight*, CNN *Headline News*, Public Radio International *The World*, Voice of America English news radio and newswires from Associated Press and New York Times. The Chinese sources were also sampled daily over the same period and include Voice of America's Mandarin News program, Xinhua newswire and news stories downloaded from Zaobao's web site (www.zaobao.com, www.asiaone.com). LDC began collecting the TDT-2 English sources and Xinhua in January 1998 adding Zaobao news in February and VOA Mandarin in March. For TDT-3, LDC continued the sampling over the three-month period from October through December 1998 and added two English television sources, NBC *World News Tonight* and MSNBC *News with Brian Williams*.

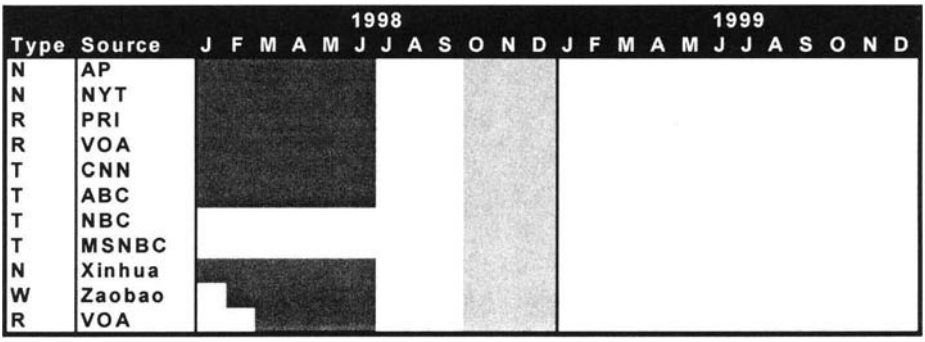


Figure 2. Data Sources collected for TDT-2 and TDT-3 Corpora.
Types: N=Newswire, R=Radio, T=Television, W=WWW Site.
Use: ■=used in TDT-2, =used in TDT-3, =not yet used

The radio and television sources are available as a single hour or half-hour broadcast daily. Voice of America broadcasts several hours of programming in both English and Mandarin. LDC recorded one or two hours of VOA broadcasts per day focusing on those time slots that contained more hard news and fewer features. CNN *Headline News* broadcasts throughout the day recycling news stories several times per hour. To maximize novel content, LDC recorded multiple half-hour segments spread across CNN's broadcast day. The newswire services deliver stories throughout the day via modem or once a day via FTP. In either case, LDC sampled the daily offering and retained on average 80 stories from the English sources and 60 from the Chinese. To avoid rendering the TDT evaluation's segmentation task artificial, LDC did not sample a fixed number of stories each day but rather collected a daily volume of data that corresponded on average to these targets.

The collection mechanism varied by source. LDC collected the broadcast television sources from cable TV using high quality, consumer grade video-cassette recorders attached to DAT recorders that connect to computer via a Townshend DAT-Link. At scheduled times each day, the VCRs began recording the broadcast video and audio on VHS tapes. A closed-caption decoder collected captions to disk when they were present. Simultaneously, a Sun Sparcstation controlling the DAT Links would instruct them to turn on the DAT recorders and begin collecting the audio portion of the television broadcast. The audio from the DAT recorders flowed through the DAT Links onto computer disk where it was stored for later processing. This process allowed LDC to retain the text, audio and video. The process for collecting broadcast radio, PRI, was simpler, replacing the cable-TV receiver, closed-caption decoder and VCRs with an AM/FM receiver. LDC acquired VOA transmissions via satellite where they were de-multiplexed, decoded and

saved to computer disk. The newswire services transmit their material either via dedicated modem or via FTP. LDC ran a daily web-harvesting program to collect the news stories from Zaobao. For each of the sources, LDC negotiated rights to use the data for purposes of research and technology development.

The TDT-2 corpus contains over 72,000 stories and 800 hours of recorded audio. TDT-3 contains more than 51,000 stories and more than 600 hours of audio. In preparation for subsequent generations of TDT research, LDC continued the collection through 1999. Figure 2 summarizes the collection schedule.

4. Transcription

While TDT newswire is collected as electronic text, the broadcast sources are collected as audio (and video where available). To support annotation and the TDT research, LDC produced text intermediaries of all of the audio. Because TDT project managers have a strong interest in systems that can handle real world data, no attempt was made to produce high quality transcripts of the kind used for speech recognition projects such as the DARPA sponsored HUB-4 and HUB-5 projects (NIST 1999, 2000). Text intermediaries come from a number of sources. LDC uses closed-captioning or commercial transcripts where they are available. Otherwise, LDC transcribes or contracts the transcription of the broadcast television and radio sources. Figure 3 summarizes the sources that provided input to TDT-2 and TDT-3 corpus building. To accommodate future use, both the audio and text intermediaries are available in the published corpus.

LDC contracted Federal Documents Clearing House (FDCH, now E-Media) to create the transcripts for approximately 180 hours of TDT-3 English audio. The data were sent to FDCH on CD-ROM in increments of 30 hours per week. FDCH produced nearly verbatim transcripts with initial speaker turns, story boundaries, standard punctuation and capitalization but lacking speaker identification, background noise, speaker hesitations, partial words and verification of the spelling of proper nouns. The TDT-3 Mandarin audio data consisted of 150 VOA program hours transcribed by Philadelphia Chinese News (PCN) a subsidiary of the Grace Communications Corporation. The data were distributed to PCN on audio tape, for use in their transcription machines. Like FDCH, PCN produced nearly-verbatim electronic transcripts of the audio files, including identification of speaker turns and story boundaries. Transcripts were encoded in GB (simplified Chinese); English speech was transcribed in English if possible and numbers were transcribed in Chinese characters. Punctuation was left to the discretion

of the transcriber. The transcripts rendered by both bureaus simulated a reasonable quality single pass commercial transcript similar to but less errorful than closed captioning. Completed transcripts were returned via FTP where they were converted to SGML format for subsequent use.

Type	Language	Source	Native Format	Source of Text
newswire	English	Associated Press	electronic text	--
newswire	English	New York Times	electronic text	--
radio	English	Voice of America	broadcast radio	CT
radio	English	Public Radio Inter.	broadcast radio	CT
television	English	ABC	broadcast TV	CC
television	English	CNN	broadcast TV	CC
television	English	NBC	broadcast TV	CC
television	English	MSNBC	broadcast TV	CC
newswire	Chinese	Xinhua	electronic text	--
web site	Chinese	Zaobao	electronic text	--
radio	Chinese	Voice of America	broadcast radio	CT

Figure 3. TDT-2 and TDT-3 sources showing native format and the source of text intermediaries. CC = "closed captioning" and CT = "contract transcription"

5. Story Segmentation

Segmentation refers to the identification of individual stories within a news broadcast by inserting boundaries where the topic of discussion changes. Newswire services deliver their material with story boundaries marked; however, the transcripts and closed-caption text from the seven broadcast sources required segmentation. TDT Segmentation is a two-pass procedure. During the first pass, annotators listen to the audio of the entire broadcast while viewing the corresponding waveform display and text intermediary and add, remove or re-position story boundaries. Annotators also classify each boundary as beginning a 1) news story 2) miscellaneous text section or 3) untranscribed section. *News stories* are topically contiguous segments of broadcast. *Miscellaneous text* segments include commercials, reporter chit-chat and the previews of upcoming news reports that are commonly found at the start of a broadcast. If an audio segment contains no text or inadequate text to determine its topic, that segment is classified as *untranscribed.* Transcripts that are incomplete but contain enough text to identify the main topic are classified as valid news stories. In TDT sources,

text intermediaries from closed-captioning contained more untranscribed sections than commercial transcripts. Annotators perform a second-pass to check the quality of the segmentation and to measure inter-annotator variation. Section 5.3 describes the second pass in further detail.

5.1 The "two clause" rule

The segmentation of the TDT-2 corpus employed the "two clause rule"; two independent declarative clauses on the same topic defined a story. Although this may seem straightforward, it in fact slowed progress and increased confusion as annotators counted clauses and argued over which were independent. Segments of potential interest, for example brief stock reports, that failed the "two clause" rule were not classified as news. The previews of upcoming reports at the start of a broadcast or immediately before a commercial break varied in their form sometimes containing two independent clauses, other times not. In TDT-3, annotators classified segments as news stories if they could easily render a judgment about the segment's relevance to a topic; story size became irrelevant. News previews were categorized as miscellaneous text and it was assumed these would be expanded later in the broadcast.

5.2 Annotator Training

TDT-3 annotators cut their teeth on files selected from TDT-2 corpus to provide a number of difficult situations including files containing large sections of under-transcribed text, files containing drop-outs in the audio signal and files that required close attention to the actual content of the reports in order to find the story boundaries. Consider the following example:

> *<t 165.00>Reporter: have you ever seen anything like this before?*
> *<t 168.00>I've seen serious storms. This is the worst since I've been here in 1972.*
> *<sr 172.118>Maintenance workers are trying to remove slabs of ice from atop buildings and bridges.*
> *<b 182.00>Power was restored to much of the city overnight, but 700,000 people outside the city are still in the dark.*

From a brief reading, one might conclude that there is one story in this example but there are in fact two different stories about two different ice storms, one in New York, and one in Montreal. Subtleties of this kind are not uncommon and emphasize the need to use all one's resources, text, audio and waveform, during segmentation. Annotators were required to demonstrate their ability to establish accurate story boundaries in difficult cases like these before they were permitted to work on actual segmentation files.

5.3 Quality Control

LDC adopted a number of measures to control the quality of segmentation, including second passing, spot-checking, dual segmentation, the review of rejected segments during topic annotation and the evaluation of the ratio of word tokens to time.

A complete second pass of all segmented data was the fundamental quality assurance measure. During the early stages of TDT-2 segmentation, the second pass was not an exhaustive review of the entire audio file. Instead, second-pass annotators simply re-examined the story boundaries placed by the first annotator and adjusted existing timestamps. Unfortunately, without listening to the entire broadcast, second-pass annotators could not find story boundaries that had been missed by the first annotator. In broadcast television sources where there may be no closed captioning for large sections of audio, this practice can result in story boundaries being missed. In TDT-3, LDC implemented an exhaustive second pass that increased segmentation accuracy. The implementation of two complete segmentation passes was costly in terms of human effort: segmentation accounted for 25% of all annotation effort for the TDT-3 corpus. However, the need for accurate and complete story boundaries warranted this expenditure. Over time, annotators tended to specialize in a particular broadcast source; this familiarity with the peculiarities of each source also lead to increases in efficiency and accuracy.

LDC senior annotators conducted spot-checks on approximately 1% of all segmented material. As they discovered mistakes or inconsistencies they remediated with the specific annotators involved but also discussed the inconsistencies with the entire annotation team via e-mail and during weekly annotator meetings. Spot-checks began during TDT-2, but became a regular part of quality control in TDT-3.

During the topic annotation stage that follows story segmentation, segments may be rejected if they actually contain two stories, are non-news or display data formatting problems. Broadcast stories that were rejected for containing more than one story, missing part of the story or containing reparable formatting problems were repaired, re-segmented and then returned to the pipeline for further annotation.

Dual segmentation served as both a quality control and a means to measure inter-annotator agreement. LDC selected 5% of all broadcast files with a balance across sources and dates and submitted them for segmentation by two independent annotators. These files received the complete segmentation treatment, both first and second pass. Senior annotators then compared the results and reconciled any discrepancies. The results of dual segmentation showed high rates of consistency among annotators. During

TDT-3, LDC dual-segmented files containing 1300 story boundaries. Of these, there were discrepancies involving 203 segments. These differences took three forms. Over half of the discrepancies were the result of stylistic differences. Segmentation staff received no instruction on how to handle reporter chit-chat; some included it at the end of the previous story, others in the beginning of the subsequent and still others segmented it as a separated miscellaneous text section. 15% of the differences resulted from other types of "judgment calls". Segmentation is not an exact science, and there is some ambiguity in the task. When reports of similar content are adjacent to one another in a news broadcast, it is often difficult to tell where one story ends and the next begins. Annotators were instructed to rely on audio cues (speaker changes, music, pauses) to inform their judgments, but some level of indeterminacy remains. 25% of the differences were attributed to a recognizable error on the part of one of the annotators (a missed story boundary or inaccurate timestamp). Human annotators compared favorably to system performance on the segmentation task when scored by NIST's evaluation software for both the TDT-2 and TDT-3 corpora.

The final quality control measure, introduced towards the end of TDT-2 and adopted as a regular check during TDT-3, was the measure of word tokens in the text intermediary of a segment per unit of time in the corresponding audio. Stories with an unusual ratio of text words to audio duration aroused suspicion of a segmentation error. These cases were reviewed, re-segmented and re-annotated as appropriate. Although this method does produce a number of false alarms involving news stories with long musical interludes, it did prove helpful for identifying missed boundaries in some audio sources.

6. Topic Definition

The notions of event and topic are crucial to TDT annotation. A TDT *event* is defined as a **specific thing that happens at a specific time and place along with all necessary preconditions and unavoidable consequences**. For instance, when an U.S. Marine jet sliced a funicular cable in Italy in February 1998, the cable car's crash to earth and the subsequent injuries were all unavoidable consequences and thus part of the same event. For the purposes of TDT, a *topic* is defined as **an event or activity, along with all directly related events and activities.** It is important to highlight the difference between a TDT topic and the notion of topic in normal discourse. While one might normally think of a topic as something broad like "accidents", a TDT topic is limited to a specific

collection of related events of the type accident, in this case a particular cable car crash.

To increase the consistency of judgments about what constitutes "related" events, annotators refer to a set of *rules of interpretation*. These rules state, for each type of *seminal event* -- crimes, natural disasters, scientific discoveries, scandals, etc. -- what other types of events should be considered related. This then informs the annotators' judgments about which stories are "on-topic". In the example above, stories about the investigation, the Marine pilot, the repercussions for his unit, the victim's families and their quest for justice were all on topic. In TDT, there are eleven topic types with corresponding rules of interpretation:

- Elections (e.g. new public officials, change in governments, voter scandals)
- Scandals/Hearings (e.g. Monica Lewinsky, Kenneth Starr's investigations)
- Legal/Criminal Cases (e.g. crimes, arrests, cases)
- Natural Disasters (e.g. tornado, snow/ice storm, flood, drought, mud-slide, volcano)
- Accidents (e.g. plane, car, train crash, bridge collapse, accidental shootings, boats sinking)
- Ongoing Violence or War (e.g. terrorism in Algeria, the Israeli/Palestinian conflict)
- Science and Discovery News (e.g. John Glenn's return to space)
- Finance (e.g. Asian economic crisis, major corporate mergers)
- New Laws (e.g. proposed amendments, new legislation passed)
- Sports News (e.g. Olympics, Super Bowl, Figure Skating Championships, Tournaments)
- Miscellaneous News (e.g. Dr. Spock's Death, Madeleine Albright's trip to Canada)

This particular conceptualization of topic is a critical component of TDT annotation, as it allowed annotators to potentially identify *all* the stories in the corpus that discussed some pre-defined topic. The topic definitions and rules of interpretation ensured that each annotator was working with the same understanding of the topic at hand and, at least in theory, that all annotators would identify the same stories as on-topic.

The format of the topic definition is fixed for each topic to enforce annotation consistency. The topic title is a brief phrase that is easy to remember and immediately evoked the topic. Each topic is accompanied by a topic icon, which provides the annotator with a visual reminder. The seminal event that contributed the topic is described by answering the questions what, when and where with regard to the event. The topic explication provides further details. Links to the relevant rule of

interpretation, topic research, Mandarin-specific topic information and links to sample on-topic stories are provided. Figure 4 contains an example of a TDT-3 topic definition.

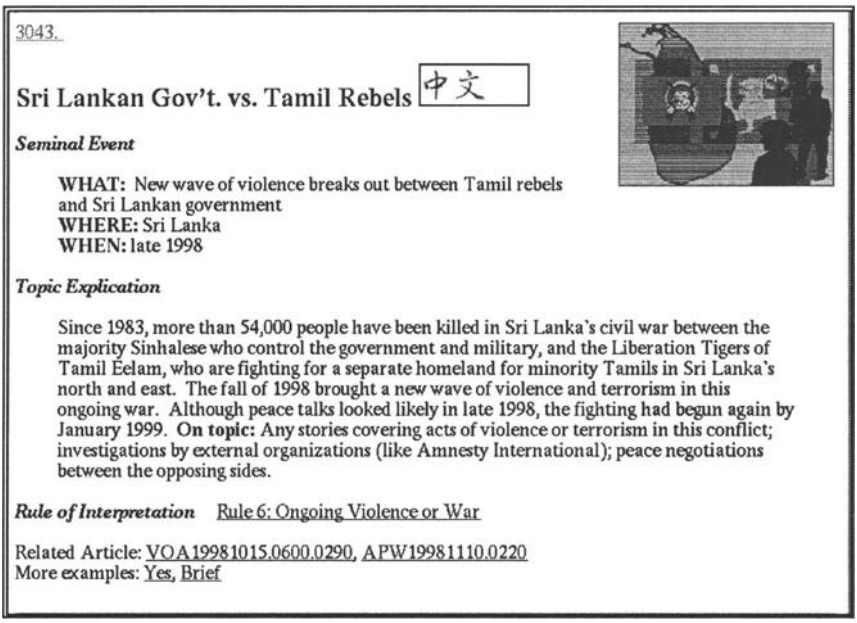


Figure 4. An example topic definition

6.1 Topic Research

One of the largest challenges to the annotators was the task of keeping abreast of developments for a particular topic. Although the topic definitions spelled out what sorts of stories might be considered on-topic, it was impossible to know in advance from having examined only one seed story how the topic might develop over time. In order to put the topics into a larger context, annotators conducted topic research, providing additional material like timelines, maps, keywords, named entities, and links to online resources, for each topic. Topic research was a valuable resource not only for initial topic annotation, but also at later stages of quality control, when it provided a framework to monitor topic development and curb "topic drift". Topic research was always accessible to annotators and was updated as the project and the topics evolved.

6.2 Topic Selection

In TDT-2, LDC defined 100 topics based upon a stratified, random sample of the six English news sources collected January through June of 1998. The sampling gave each month of data from each source an equal chance of contributing a topic. Within any month of data from a source, stories were selected at random. There was no requirement that any topic produce a minimum number of on-topic stories. For TDT-3, a central requirement was to ensure that for each topic there were at least four on-topic stories both in the English data and in the Mandarin data. This involved a closely coordinated effort by senior annotators to determine appropriate search strategies. In order to select the 60 required topics, over 1600 seed stories were considered. For the 2000 TDT Evaluation, LDC augmented TDT-3 with an additional 60 topics. The seed stories that generated these topics were equally divided between English and Mandarin. There was no guarantee that any topic would produce a minimum number of hits in either language.

7. Topic Annotation

LDC performed four types of topic annotation on the TDT corpora: topic-story, search-guided topic-story, first story and story-link annotation.

7.1 Topic-Story Annotation

The vast majority of annotator effort in both TDT-2 and TDT-3 was devoted to topic labeling. This annotation task alone comprised a third of all annotator effort during TDT-3. Using a custom-designed interface, annotation staff read the text of each story in each daily news file and decided whether it related to the selected topic. For each topic-story pair the annotator rendered a decision of *yes* if the story discussed the topic, *brief* if the story contained only a brief mention of the topic or *no* if the story contained no mention of the topic. Any mention of a topic warranted a label of at least *brief*. Stories that were primarily about something else but discussed the target topic in at least 10% of their volume were also labeled *yes*. This preserved the premise that news stories can discuss more than one topic. Annotators could also reject a story as non-news or as exhibiting some data formatting problem. A comment field allowed annotators to record their analysis of problem stories, or to note questions about a news report (e.g., "Is this one or two stories?"). The interface also allowed annotators to review their work and make changes.

For TDT-2, LDC English annotation staff made five passes over the data each time labeling a story with respect to 20 topics. Subsequently, 20 of these topics were also annotated in Mandarin. In 1999, LDC annotated TDT-3 for 60 topics jointly defined in English and Mandarin and requiring 3 complete passes over the data. Before beginning each session, annotators were required to study the relevant list of topic definitions and review the topic research documents, where available. The TDT labeling interface, in Figure 5, guided annotation staff through the stories, recorded each annotator's progress and logged their judgments into an Oracle database.

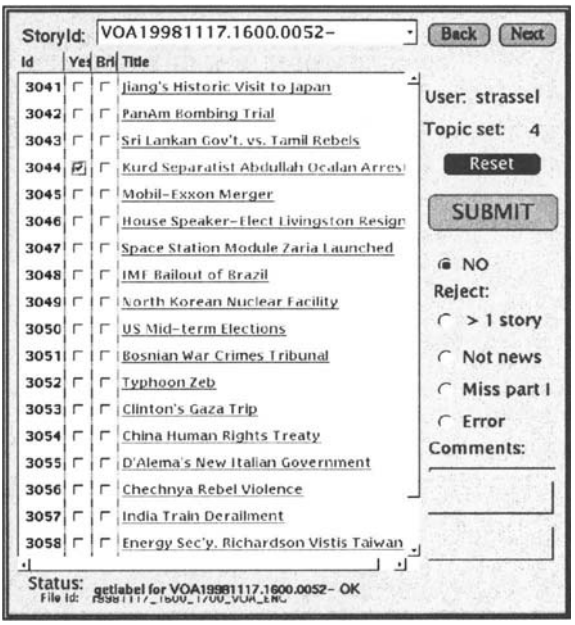


Figure 5. The TDT topic-story annotation interface

7.2 Topic-Story Quality Control

For both TDT-2 and TDT-3, between 5% and 8% of all news files received a complete second annotation by an independent annotator. During TDT-2, this process required project managers to hand-select files for dual annotation and reassign them manually to independent annotators. Although the staff was not told that the files had already been annotated, because of the timing and style of the assignment of these files, annotators often suspected that they were duplicating effort. The annotators did not know who had done the first round of annotation, and they did not have access to the original

annotators' judgments. However, from a procedural design perspective, dual annotation was single-blind at best.

For TDT-3, the topic labeling interface was modified to provide double-blind assignment of dual annotation files. No one, not even project managers, knew which files had been selected for dual annotation or to whom they had been assigned. The dual annotation was completely incorporated into the regular distribution of work. After topic labeling had been completed for a particular list, discrepancies between the two sets of topic labels were reviewed and inter-annotator consistency was measured. The topic labeling interface allowed team leaders to act as "fiat", a superuser who checked discrepancies and corrected errors. The kappa statistic was used to measure consistency of human annotation. Where a kappa of 0.6 indicates marginal consistency and 0.7 measures good consistency, kappa scores on TDT-2 were routinely in the range of 0.59 to 0.89. Scores for TDT-3 ranged from 0.72 to 0.86.

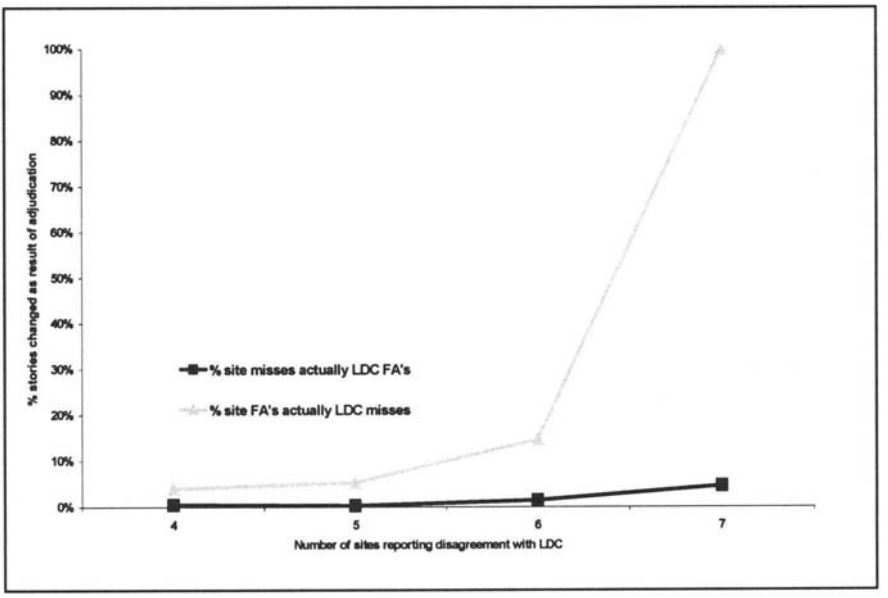


Figure 6. During adjudication of system results, the probability that the systems had uncovered an annotator error grew in proportion to the number of systems reporting disagreement with the annotation.

During Precision QC, senior annotators reviewed all stories labeled as *yes* or *brief* to identify possible false alarms, stories erroneously labeled as on-topic. Working with a modified version of the labeling interface and examining one topic at a time, senior annotators read each story and either

verified it as on-topic, or vetoed it. When possible, the precision check was performed by the same senior annotator who had conducted topic research for that topic. During the precision check, annotators kept a sharp eye out for cases of "topic drift", when the definition of the topic varied across annotators or over the course of topic labeling. By referring back to the topic explication and rules of interpretation, topic research documentation and annotator e-mail archives discussing the topic, senior annotators excluded stories outside the scope of the topic. Team leaders independently verified all changes resulting from the precision check. The results of Precision QC revealed a very low incidence of false alarms. For TDT-3, the precision check resulted in a veto of only 2.5% of the original annotator judgments (213 of 8570 on-topic stories).

During Recall QC, senior annotators used a search engine to generate a projection of the corpus relevance-ordered with respect to a single topic. Queries could be the seed story, a list of miscellaneous keywords, the topic explication itself or the union of all stories labeled as related to the target topic or a subset thereof. In TDT-2, LDC staff conducted an exhaustive recall check over all 100 topics. The search engine returned a list of 1000 relevance-ranked stories. Annotators were required to skim through the list of 1000 to find any potential misses. After reading at least 50 consecutive off-topic stories, annotators could evaluate whether there were likely to be any on-topic stories further down the relevance-ranked list. If the 50 consecutive off-topic stories were clearly irrelevant to the topic, annotators were permitted to move on to the next topic.

Because this labor intensive Recall QC yielded a relatively small return in TDT-2, TDT-3 underwent a more limited Recall QC. A search engine identified and an annotator checked potential misses prior to the (chronologically) first on-topic story. During the evaluation of the tracking task, systems are given four chronologically early on-topic stories as training data. Therefore, a miss during the training epoch was thought to have more severe consequences for the evaluation.

Recall QC for both TDT-2 and TDT-3 showed, not surprisingly, that the rate of misses is higher than the rate of false alarms for human annotators. False alarms were easily caught and corrected since the ratio of on-topic stories to total stories was relatively low. The total number of stories in each corpus, on the other hand, was very high (approximately 72,000 for TDT-2 and 51,000 for TDT-3). For some topics, the "hit rate" was extremely low. In TDT-2, some topics had no hits at all; in the TDT-3 1999 Evaluation topics, each topic was guaranteed to have at least 4 hits in each language but often had no more than that. For these small topics, labeling was akin to searching for a needle in a haystack, and some number of misses was inevitable.

In order to offset the potential for missed stories even after recall QC, the LDC performed one additional quality assurance task. NIST provided the LDC with the results of each research site's results for the topic tracking task. The sites' systems were scored against the LDC's human-produced topic relevance tables, with the annotators' judgments taken as ground truth. Each system miss corresponded to a potential LDC false alarm, and each system false alarm was a potential LDC miss. The LDC adjudicated the systems' results as a final QC measure.

It would not have been feasible to completely adjudicate all cases where LDC annotators differed from system performance. In the case of the TDT-3 1999 Evaluation topics, NIST delivered results containing approximately 1.5 million topic-story tuples from 7 research sites. The effort needed to adjudicate all the cases of discrepancy would have exceeded the original corpus creation effort. Instead, the LDC reviewed cases where a majority of systems (i.e. 4 or more) disagreed with the original annotation. The adjudication effort for TDT-2 was larger than that of TDT-3 but still did not entail a complete adjudication of all discrepancies.

For both corpora, the number of LDC false alarms uncovered through the adjudication process was very low – for TDT-3, less than 1% of putative system misses revealed LDC false alarms. This was not surprising, given the complete precision check over all on-topic stories that had already eliminated most LDC false alarms. The rate of LDC misses identified during adjudication was higher than that of false alarms for both TDT-2 and TDT-3. However, even for the TDT-3 corpus when no exhaustive recall check was performed, the rate of LDC misses was quite low: only 5% of stories reviewed during adjudication were actual misses. As expected, the probability that a putative system false-alarm was in fact an LDC miss was in direct proportion to the number of systems reporting disagreement with LDC annotations.

7.3 Search Guided Topic-Story Annotation and QC

In preparation for the 2000 TDT evaluation, LDC selected and annotated another 60 topics from the TDT-3 corpus. Half of these topics were selected from English sources and half from Chinese sources. All topics were annotated in both language but there was no a priori guarantee that there would be on-topic stories in both languages for all topics.

Although previous topic-story annotation had employed a brute-force approach in the sense that every story was read for each list of 20 topics, for the 2000 evaluation, it seemed plausible to implement a search guided annotation strategy. Previous experiments had shown that search guided annotation could produce results as good as brute force annotation while

reducing costs and the effect of fatigue. In search guided annotation, individual annotators work with one topic at a time. After using the procedures described in section 6.0 to perform topic selection, research and definition, one annotator would make multiple passes over the English data while another made multiple passes over the Chinese data in an attempt to find all stories discussing the topic at hand. During topic development, annotators were able to initially identify English seeds for 51 of the 60 topics and Chinese seeds for 34 of the topics.

The first pass involves submitting the concatenation of all on-topic stories as a query into the corpus. During this first stage, on-topic stories includes the seed story itself and any stories found during topic definition and research. The annotators then read through all the stories in the relevance-ranked list returned by the search engine until reaching the "off-topic threshold", defined as a 2:1 ratio of off-topic to on-topic stories provided that the last 10 stories read were off-topic. As before, each story is labeled yes, brief or no.

In the second stage, annotators iterated their searches using the concatenation of all on-topic stories as they continued to find them. Although the teams were required to perform only one query at this stage, they were encouraged to and in fact often did perform multiple iterations. For the most difficult topics, the teams performed as many as 15 additional English queries and 20 additional Chinese queries. Regardless of how many queries they issued, annotators were obligated to read enough stories to reach the "off-topic threshold" before moving to the third stage.

During the third stage, annotators issue new queries using the text of the topic research document and topic explications. As before they read the relevance-ranked list of returns to reach the "off-topic threshold" before progressing to the fourth stage. During the third stage annotators were required to issue at least one query and reach the "off-topic threshold" at least once. In typical fashion, some annotators issued as many as 9 English queries and 14 Chinese queries.

In the fourth stage, annotators were encouraged to think creatively. By this point they had worked on the topic for at least several hours and as many as a few days. They had become topic experts. Here are the instructions given to the teams:

> *You are encouraged to use your specialized knowledge (drawn from topic research and the known on-topic stories) to conduct additional manual searches through the corpus. These additional searches will be based on keywords, names, particular on-topic stories, etc. Think creatively! If you come up with a novel way to search for additional on-topic stories, let us know. If you find*

> *additional information (names, places, dates, events) about your topic, you should revise the topic research page for that topic.*

Annotators took this challenge seriously. Here is an example of the notes kept by one staffer working on the European Cold Wave topic:

> *In annotating this topic I had to go beyond the regular parameters. It was apparent that there were YES stories remaining beyond the "off-topic threshold". Many of the intervening NO stories were CNN weather reports that had nothing to do with the topic. So I did extra text searches and concentrated on stories within a particular timeframe to find additional hits.*

The quality control measures for this annotation method were similar to those used in brute-force topic-story annotation. Working through topics individually, annotators reviewed all stories marked yes. During Precision QC, they changed 18 English stories from 'yes' to 'no' and 47 from 'yes' to 'brief'; 5 Chinese stories were changed from 'yes' to 'no' and 9 from 'yes' to 'brief'. This yields an overall precision of 96%. Given the nature of the annotation task, Recall QC would have meant nearly repeating the original task. LDC did perform Dual Annotation on 10% of all topics (6 in each language). Given the nature of search-guided annotation, one cannot derive measures of inter-annotator consistency from dual annotation. The relevance-ranked lists returned by the search engine depend upon their input and prior annotation. Where annotation results differ, normal variation in annotation practice is as likely an explanation as annotator inconsistency. However, LDC did perform dual annotation purely as a form of quality control. Reviewing 6 English topics with a total of 142 on-topic stories and 6 Chinese topics with 107 on-topic stories, annotators found 31 discrepancies in English and 18 in Chinese. The 2000 evaluation recycled the 60 topics used in 1999 and added 60 more. After sites returned their results, LDC adjudicated those cases in the 2000 topics where the majority of sites, four or more, disagreed with LDC annotation as well as those cases in the already adjudicated 1999 topics where all sites disagreed with LDC.

7.4 First Story Annotation and Quality Control

In the First Story Detection task, systems must find the first story in the corpus discussing each topic in the evaluation. To support this task, LDC selected 180 topics in the TDT-3 corpus and found the chronologically first as well as several other on-topic stories. 60 of the 180 topics were those selected for topic-story annotation; 120 were new topics from seed stories selected at random but with each month and source of data having an equal probability of contributing a topic.

First story annotation involved three stages: the identification of the seminal event and definition topic; the identification of additional on-topic articles; and the search for the chronologically first on-topic story. Annotators were given lists of seed stories and instructed to develop a brief TDT-style topic definition where possible. All the topics were then reviewed to ensure that each seminal event was unique, although topics were permitted to partially overlap in their content. Senior annotators reviewed all topic descriptions. The annotators then used the seed story as a query into the corpus. At least fifty of the top-ranked stories were reviewed for each of the 180 topics. During the third phase annotators continued to issue searches based on the concatenation of on-topic stories. The interface selected the most highly ranked results of these queries, excluded stories already labeled on-topic and sorted the remainder in reverse chronological order. Annotators worked through this list refining their search and looking iteratively in chronologically earlier periods. This type of annotation requires the annotator to think creatively. Annotators were able to modify their searches by changing their queries but also by limiting the returns to a specific time period.

LDC conducted limited Precision and Recall QC on the first story annotation results. Precision QC involved the review of each identified first story to ensure that the story was indeed on the particular topic and was a logically reasonable first story. Given that a true Recall QC effort would have replicated the original task, Recall was performed only on those topics modified during Precision QC. Of the 120 additional topics, LDC found two whose first stories had been identified erroneously as on-topic.

7.5 Story Link Annotation

The story linking task was introduced into the 1999 Evaluation as a way of avoiding the difficulties and costs of explicitly defining a TDT topic. Unlike topic-story annotation, where one defines a topic explicitly and somewhat formally after identifying a seed story, in story-link annotation, one reads two stories and decides whether they discuss the same topic. The same meta-definitions of event and topic apply, but the annotator need not make them explicit to proceed. Eleven annotators, all undergraduates or recent University graduates and all familiar with the TDT concept of topic and trained in topic annotation, performed the story linking annotation.

In story linking, the annotator reviews a pair of stories, called the "seed" and "compare" story. Reading the seed story, the annotator naturally begins to construct a notion of topic. As the annotator reads the compare stories, the goal is to decide whether the two discuss the same topic even though the topic definition has not been formalized. There were 180 seeds for Story

Link annotation; 60 were the seeds used in 1999 TDT-3 topic story annotation and 120 were selected randomly but evenly across months and sources of data. Duplicate and near duplicate stories were avoided. For each seed, 120 compare stories were selected as follows. Each seed was submitted as a query into the corpus. The top 60 relevance-ranked returns comprised one half of the compare story set; the second half were randomly selected to provide a cluster centered chronologically around the time of the seed story.

研究方法探討 During annotation, LDC staff mentally established the topic for the seed article and determined for each compare story whether it discussed the same topic, labeling each story pair as 'yes' or 'no'. The story link annotators were given no specific methodology to determine that two stories discuss the same topic. This approach casts topic as a free variable subject to evolution and drift and probably better simulates a real world use of TDT technology, where users have not undergone training in topic definition. Once a compare list had been completed, the annotators were not allowed to modify their previous judgments. Although collaboration was encouraged and even required in other forms of TDT annotation, story-link annotators were isolated as much as possible from the other annotators and not permitted to speak to one another about topics or annotation approaches. They also received minimal input from senior annotators prior to the Precision QC stage. During story link annotation, 21,506 judgments were rendered and 94 stories were rejected as ill-formed. Annotators established links for 3760, or 17% of, the story pairs.

Precision QC involved reviewing and confirming or modifying all linked story-pairs. Although annotators were initially instructed to label story links simply as 'yes' or 'no', in order to increase consistency with the other TDT annotations, we introduced the 'brief' label during Precision QC. Of the 3760 story pairs labeled 'yes', 3068 were confirmed, 350 were changed to 'brief' and 342, or 9%, were changed to 'no'.

Annotators performed a limited Recall QC reviewing the three story pairs per topic which had the highest relevance according to the search engine that had nonetheless been labeled 'no' by humans. During Recall QC, annotators viewed stories in pairs where the identity of 'seed' and 'compare' were unknown. Three annotators worked on this task, each reviewing one of the three story pairs guaranteed not to be a pair they had annotated previously. As we saw elsewhere, for an annotation project of this size and density requiring intense human effort, problems in recall are greater than those in precision. Of the 521 story pairs reviewed, 72% retained their 'no' label, while 17.7 % changed to 'yes' and 10% changed to 'brief'. 關聯性有可能再次更改

Reviewing those topics that had experienced the greatest changes as a result of QC, we observed the trend that topics tended to become more limited and precise in scope during QC. In one example, a seed story

discussed a soccer match during the World Cup. The first annotator defined the topic as the World Cup, while the second annotator performing Precision QC felt that the topic should have been the individual game. Another seed story discussed a space shuttle launch. The original annotator defined the topic as the entire mission and found 39 story links. During Precision QC, the second annotator focused on the actual launch and reduced the number of links to 13 'yes' and 3 'brief'. It should be noted that no attention was paid to the ordering of story-pairs during this annotation. On any subsequent Story Link annotation we would propose to present story pairs in order of relevance both to echo the approach now commonplace in web searches and to try to guide annotators toward more conscious decisions about the granularity of their topic definition.

8. Corpus Formats

The published TDT corpora include the following components:

- original reference audio of the broadcast sources
- SGML-structured, segmented reference text used as background information and for annotation but excluded from the evaluation tasks
- automatic speech recognition output for all audio
- tokenized, unsegmented versions of the reference text and ASR output
- reference and tokenized versions of machine translation of original text and ASR output
- boundary tables indicating the position within tokenized files of story boundaries
- relevance tables indicating which stories are related to each topic or each other
- NIST's canonical, chronological ordering of stories

Each sampling unit has a unique file ID, which identifies the date and time that the sampling took place, and the source. Several sources are sampled multiple times per day. The file ID's are formatted so that when they are put in a default sorted order they will appear in chronological sequence. Some typical file ID's are:

- 19980106_1830_1900_ABC_WNT
- 19980121_2001_2105_NYT_NYT
- 19980122_2300_2400_VOA_TDY

The file ID's are used to name the file that contains the SGML text data for each sample unit, and are likewise used in all other files derived from or relating to that sample unit.

8.1 Reference Audio

In TDT, a sampling unit consists of the stories captured from one source during one contiguous time span. For each sampling unit from broadcast sources, there exists a single file of the audio stream sampled as single channel 16K/16bit audio with NIST SPHERE headers. These files are marked with the .sph extension

8.2 SGML-tagged text archive

For each sampling unit, there is one file of SGML-tagged text data, distinguished by the .sgm extension. Within each SGML sample unit file, the individual stories are marked off as with SGML <DOC> tags. The <END_TIME> tag applies to audio sources only.

- <DOC> units. Essential information about each story is provided in the following SGML tags that appear in the initial part of each DOC unit:
- <DOCNO> -- the unique identifier for each DOC unit
- <DOCTYPE> -- "NEWS STORY", "UNTRANSCRIBED" or "MISCELLANEOUS TEXT"; only "NEWS STORY" units are used for topic labeling annotation
- <DATE_TIME> -- contains the time-stamp associated with the unit; for audio sources, this serves to locate the starting time of the unit in the audio stream
- <END_TIME> -- contains the full time-stamp associated with the end of the unit in the audio stream

In the case of newswire sources (APW, NYT, XIN, ZBN), the SGML archival form is derived from the format (ANPA or other) in which the data are sent to LDC. The SGML sample files contain only the stories that have been selected for annotation. In the case of audio sources (ABC, CNN, NBC, MNB, PRI, VOA, VOM), the SGML archival form is derived from closed-caption text or commercially produced transcripts that have been annotated manually to establish time stamps and DOCTYPE labels for all DOC unit boundaries. All DOC unit types are retained for the audio sources.

8.3 Untagged, Tokenized Text Stream Files

This form of text data is derived from the SGML-tagged text archive. For each sample unit from each source, there is one file containing all the text content of the sample, in which each space-separated orthographic token is presented as a separate data record, and all other information from the original sample is excluded. These files are evident by the .tkn extension.

The excluded information consists of:
- all SGML tags
- all material that lies outside of the "<TEXT>...</TEXT>" units
- all material within "TEXT" that is enclosed inside the tags "<ANNOTATION> ... </ANNOTATION>"
- "dateline" strings at the beginnings of newswire articles

Regarding the last item, newswire stories typically begin with a "dateline" string that identifies the location from which the news report was originally sent, for example:

> *BELFAST, Northern Ireland (AP) _ With peace talks already threatened by a spate of killings, police in Northern Ireland detonated a car bomb Wednesday in a rural town near Belfast.*

Given this paragraph from the SGML-tagged archive, the untagged stream would contain only the space-separated tokens following the underscore character "_". The tokenization process simply splits the text stream using white space as the delimiter. All punctuation, hyphenation, quotations, and parentheses are retained without modification. In other words, every string of one or more contiguous non-space characters is output as one token. All strings of one or more white-space characters are treated alike as single delimiters; this has the effect of eliminating paragraph boundaries in newswire, which are marked in the SGML files by indentation. Except for this neutralization of white-space, and the elimination of datelines and ANNOTATION tags, it is possible to reconstruct the full content of the TEXT elements of the SMGL files from the tokenized text files. Figure 7 shows the relationship between reference and tokenized text.

8.4 Story Boundary Table for Tokenized, Untagged Text

Given the "fileid" and "recid" attributes in the tokenized text stream data, the story boundary table relates the original DOC units to the DOCSET data records, by providing the starting and ending record ID's for each DOCNO. Each line of the table provides the SGML file ID, the DOCSET file ID, the DOCNO string, and the starting and ending "recid" values within the DOCSET file that represent the boundaries of the DOC unit. For audio sources, the table also gives starting and ending time offsets for each DOC, in seconds, from the start of the file.

Figure 8 provides an example of two records in the boundary table and the tokenized text that they segment. In the case that a DOC unit from a broadcast source contains an empty TEXT portion, the table entry for that

DOC does not contain the "Brecid" and "Erecid" attributes. The "Bsec" and "Esec" attributes are always present for broadcast sources, but are not present for newswire sources.

8.5 ASR Output

For each audio sample file, there is a file of ASR output text, which is similar in format to the tokenized text stream: each orthographic word output by the ASR system is a separate data record, marked as follows:

<DOCSET type=ASRTEXT fileid=19980109_1830_1900_ABC_WNT>
<X Bsec=0.00 Dur=0.01 Conf=NA>
<W recid=1 Bsec=0.01 Dur=0.15 Clust=45 Conf=0.32> IS
<W recid=2 Bsec=0.16 Dur=0.40 Clust=45 Conf=0.68> WHETHER
<W recid=3 Bsec=0.74 Dur=0.32 Clust=45 Conf=0.90> FOR
<W recid=4 Bsec=1.06 Dur=0.38 Clust=45 Conf=0.87> OTHERS
<W recid=5 Bsec=1.44 Dur=0.19 Clust=45 Conf=0.80> IT
<W recid=6 Bsec=1.63 Dur=0.16 Clust=45 Conf=0.75> IS
<W recid=7 Bsec=1.79 Dur=0.49 Clust=45 Conf=0.78> SPRING
...
</DOCSET>

The additional attributes attached to each word ("Bsec, Dur, Clust, and Conf") are provided by the ASR system. The "<X>" tags represent periods of time in the audio signal where no speech was recognized. The "Conf=" attribute is a computed estimate of confidence in the correctness of a given recognized word, varying between 0 (no confidence) and 1 (highest confidence). At present, this measure does not apply to non-speech (X) segments, and it is also possible that the ASR system may be unable to assign a confidence score to some recognized words; in these cases, the attribute is given as "Conf=NA".

The ASR output spans the entire audio recording for each sample file. In some cases, the manual transcription or closed-caption text for the sample may begin or end at a different point in time than the audio recording, so that the ASR output may contain more (or less) material at the beginning or end than the corresponding SGML file. However, the boundaries of news-story segments should always be properly aligned. <DOC> units, whose <DOCTYPE> is "MISCELLANEOUS TEXT", absorb discrepancies in the amount of content at the beginning or end of a sample. The "Bsec" and "Dur" attributes do not necessarily account for every second of elapsed time in the broadcast; there may be time gaps between successive records in the ASR file.

<DOC>	<DOCSET type=NEWSWIRE
<DOCNO>APW19980104.0002</DOCNO>	fileid=19980104_0002_0418_APW_ENG
<DOCTYPE>NEWS STORY</DOCTYPE>	collect_date=19980104_0002
<DATE_TIME>01/04/1998	collect_src=APW src_lang=ENGLISH
00:02:00</DATE_TIME>	content_lang=NATIVE>
<HEADER>w2488 &Cx1f; wstm-u i &Cx13;	<W recid=1> The
&Cx11;	<W recid=2> mystery
BC-Cambodia-PolPot 01-04 0570</HEADER>	<W recid=3> surrounding
<BODY>	<W recid=4> Pol
<SLUG>BC-Cambodia-Pol Pot</SLUG>	<W recid=5> Pot
<HEADLINE>Pol Pot has fled Cambodia, Thai	<W recid=6> deepened
minister claims</HEADLINE>	<W recid=7> Sunday
&UR; By ROBIN McDOWELL &QC;	<W recid=8> after
&UR; Associated Press Writer &QC;	<W recid=9> Thailand's
<TEXT>	<W recid=10> foreign
PHNOM PENH, Cambodia (AP) _ The mystery	<W recid=11> minister
surrounding Pol Pot deepened Sunday after	<W recid=12> claimed
Thailand's foreign minister claimed that the Khmer	<W recid=13> that
Rouge leader had fled Cambodia.	<W recid=14> the
Earlier, Chinese diplomats here denied allegations	<W recid=15> Khmer
he had been granted asylum in China. They could not	<W recid=16> Rouge
be reached for comment Sunday.	<W recid=17> leader
One of modern history's most secretive figures, Pol	<W recid=18> had
Pot was last seen by independent observers last	<W recid=19> fled
October at the Khmer Rouge base of Anlong Veng in	<W recid=20> Cambodia.
northern Cambodia.	<W recid=21> Earlier,
Breaking an 18-year silence he denied to a Western	<W recid=22> Chinese
reporter that he had orchestrated the killings of as	<W recid=23> diplomats
many as 2 million of his countrymen in the mid-	<W recid=24> here
1970s.	<W recid=25> denied
(material deleted)	<W recid=26> allegations
</TEXT>	<W recid=27> he
(PROFILE (WS SL:BC-Cambodia-Pol Pot; CT:;	<W recid=28> had
(material deleted) (LANG:ENGLISH:)))	<W recid=29> been
</BODY>	<W recid=30> granted
<TRAILER>AP-NY-01-04-98	<W recid=31> asylum
0002EST</TRAILER>	<W recid=32> in
</DOC>	<W recid=33>China

Figure 7: A TDT story in both SGML-encoded reference form and in tokenized format. Note that paragraph information, headlines, slug-lines, datelines and the like have all been removed.

<BOUNDARY docno=APW19980104.0002 doctype=NEWS Brecid=1 Erecid=533>		
<BOUNDARY docno=APW19980104.0012 doctype=NEWS Brecid=534 Erecid=724>		
<W recid=515> Pressed	<W recid=531> Hun	<W recid=547> hit
<W recid=516> against	<W recid=532> Sen	<W recid=548> two
<W recid=517> the	<W recid=533> army.	<W recid=549> separate
<W recid=518> Thai	<W recid=534> Seven	<W recid=550> ski
<W recid=519> frontier,	<W recid=535> skiers	<W recid=551> parties
<W recid=520> the	<W recid=536> were	<W recid=552> in
<W recid=521> royalists	<W recid=537> killed	<W recid=553> the
<W recid=522> are	<W recid=538> and	<W recid=554> Selkirk
<W recid=523> continuing	<W recid=539> at	<W recid=555> Mountains
<W recid=524> to	<W recid=540> least	<W recid=556> in
<W recid=525> hold	<W recid=541> one	<W recid=557> southeast
<W recid=526> out	<W recid=542> person	<W recid=558> British
<W recid=527> against	<W recid=543> was	<W recid=559> Columbia,
<W recid=528> a	<W recid=544> missing	<W recid=560> police
<W recid=529> far	<W recid=545> after	<W recid=561> said
<W recid=530> superior.	<W recid=546> avalanches	<W recid=562> Saturday.

Figure 8: The two boundary table records above impose segmentation on tokenized text shown below them. Story APW19980104.0002 ends at word 533. Story APW19980104.0012 begins at word 534.

8.6 Story Boundary Table for ASR Output

Given the "fileid" and "recid" attributes in the ASR text stream data, the story boundary table relates DOC units to the ASR data records, by providing the starting and ending record ID's for each DOCNO. Each line of the table provides the DOCNO string, the DOCTYPE value, the starting and ending "recid" values, and the starting and ending time offsets in seconds. An example:

<BOUNDSET type=ASRTEXT
fileid=19980109_0100_0130_CNN_HDL>
<BOUNDARY docno=CNN19980109.0100.0000 doctype=NEWS
Bsec=0.00 Esec=8.00 Brecid=1 Erecid=21>
<BOUNDARY docno=CNN19980109.0100.0008
doctype=MISCELLANEOUS Bsec=8.00 Esec=19.00 Brecid=22
Erecid=49>
<BOUNDARY docno=CNN19980109.0100.0019 doctype=NEWS
Bsec=19.00 Esec=67.00 Brecid=50 Erecid=189>
<BOUNDARY docno=CNN19980109.0100.0067 doctype=NEWS
Bsec=67.00 Esec=89.09 Brecid=190 Erecid=251>

...

</BOUNDSET>

In some cases, a "DOC" of type "MISCELLANEOUS" will span a period of time in which the ASR system will not have found any recognizable speech. In such cases the "Brecid" and "Erecid" attributes will not be present in the BOUNDARY tag; the "Bsec" and "Esec" attributes are always present.

Rel. Table	<ONTOPIC topicid=20001 level=YES docno=ABC19980110.1830.1008 fileid=19980110_1830_1900_ABC_WNT comments=NO>
SGML Text of Story	<DOC> <DOCNO> ABC19980110.1830.1008 </DOCNO> <DOCTYPE> NEWS STORY </DOCTYPE> <DATE_TIME> 01/10/1998 18:46:48.41 </DATE_TIME> <BODY> <HEADLINE> CONSUMER ELECTRONICS SHOW </HEADLINE> Byline:JACK SMITH, AARON BROWN High:MAKING TECHNOLOGY WORK FOR YOU Spec:COMPUTERS / TECHNOLOGY / ELECTRONICS / CONSUMERS <TEXT> <TURN> <ANNOTATION> spkr:AARON_BROWN </ANNOTATION> President Clinton's point man on the financial crisis in Asia is heading towards Indonesia tonight. <ANNOTATION> (voice-over) </ANNOTATION> Deputy Treasury Secretary Lawrence Summers will pressure government and business leaders there to put in place the belt-tightening measures required by the I.M.F. in... <ANNOTATION> (on camera) </ANNOTATION> ...exchange for billions to bail out failing Indonesian banks and businesses. </TEXT> </BODY> <END_TIME> 01/10/1998 18:47:09.84 </END_TIME> </DOC>

Figure 9: The topic relevance table indicates which stories (as defined by the boundary table) discuss TDT topics. In this case, story number ABC19980110.1830.1008 discusses topic 20001, the Asian Economic Crisis.

8.7 Relevance Tables

For each of the target topics defined in a TDT corpus, the topic relevance table lists the DOCNO strings for all DOC units that were judged relevant to that topic. Figure 9 shows a snippet of text and a record in the relevance table

referring to it. The topics are identified by an index number. Each line of the table has the topic index number, the file ID in the SGML archive set, the DOCNO, and the level of relevance ("YES", "BRIEF" or "NO"). DOC units that were judged irrelevant to all topics are not listed in the TDT-2 relevance table but are included in TDT-3. If the annotator entered remarks (about something unusual or noteworthy in a given story or its relation to a given topic) the existence of a comment is noted, and all comments are assembled in a separate listing. Stories can be judged as relevant to more than one topic, in which case the same "docno" will appear more than once in this table, with different "topicid" values.

9. Some Properties of the Corpus

We will conclude by observing some properties of the corpus whose description may benefit system developers working in TDT.

9.1 Segment Duration by Source, Type and Time

At some point during the segmentation of TDT-2, staff began to note anecdotally that different sources seem to follow patterns in the length and position of news segments. Plotting the time of each segment in each broadcast for each source over the duration of the collection does indeed show some source specific patterns. Consider Figure 10; each dot shows the position of a segment boundary in the NBC data for TDT-3. The x-axis shows days of the collection from October 1 through December 31, 1998. The y-axis shows time in seconds into the broadcast.

Although the graph for NBC segments reveals the least obvious pattern of all sources, there are still some noteworthy properties. Empty columns show days when no data were collected; those are simply artifacts of the collection process. When segments line up horizontally this shows a tendency to start a new story at the same time in each broadcast. Except at the very beginning and very end of the broadcast (top and bottom of the chart), there are few recognizable horizontal lines. Note however, that the segments do seem to be evenly spaced. This becomes clearer in Figure 11. Here all segments for each day of NBC are plotted together. The x-axis now shows time in the broadcast while the y-axis shows the segment duration in seconds. Note that untranscribed segments tend to occur in the very beginning of the program and average 14 seconds in length. Miscellaneous text sections also tend to occur within the first 10 minutes of the broadcast and not thereafter and average two minutes in length. Finally there seem to be two types of news segments, those that average 30 seconds and those that average about two

minutes in length. As the broadcast progresses the proportion of short and long segments varies visibly.

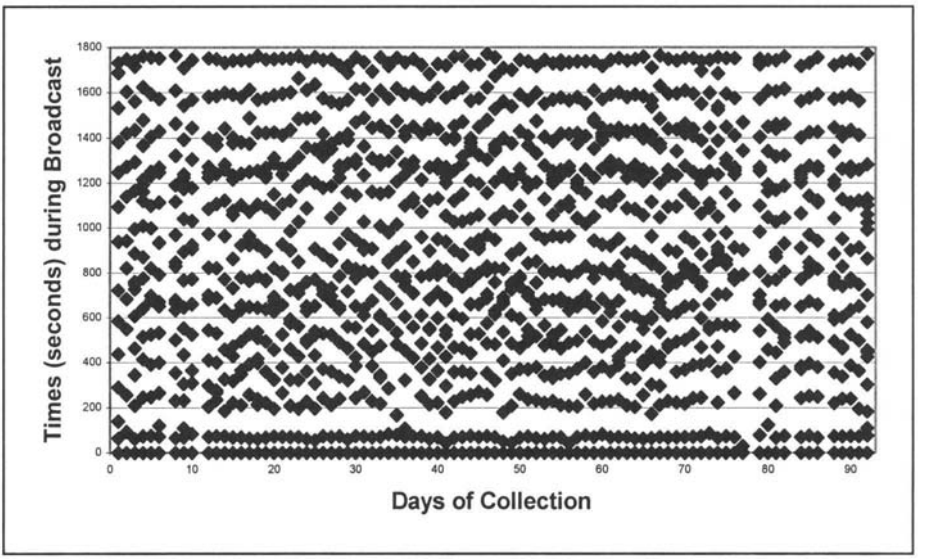


Figure 10. NBC Segments in TDT-3 as a function of date and time.

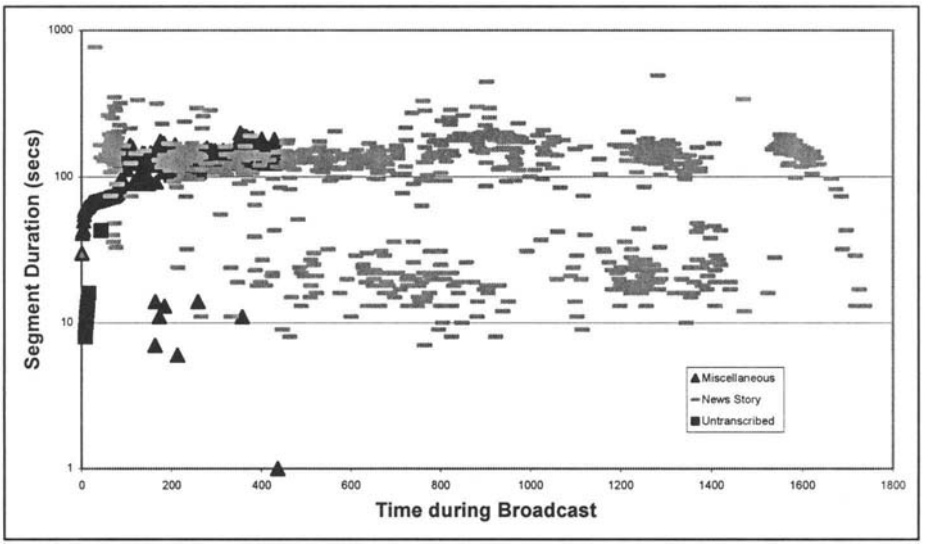


Figure 11. NBC segments by type, duration and time in the broadcast

The case for some other sources is much clearer. Consider the following plot for PRI, with quite clear demarcations of the broadcast hour into regions where short and long segments prevail. PRI broadcasts have an obvious structure that was maintained throughout our three months of collection.

Figure 12. The very evident structure of PRI broadcasts.

9.2 Topic Richness and Topic Type

Figure 13 shows the number of hits in English and Mandarin for TDT-3 1999 topics. The horizontal axis shows topics sorted by richness. The vertical axis shows the number of stories discussing each topic. The lower, darker sections of the stacked bars represent hits in Chinese; the lighter, top sections represent hits in English.

Of the 60 topics, two dozen were discussed in 100 or more stories. Two were discussed in 500 or more stories. Most topics have more hits in English than Chinese

The final graph in Figure 14 shows TDT-3 topics sorted by type (according to rule of interpretation) and then by size. The x-axis shows individual topics and topic-type averages (the darker bars). The y-axis shows the number of hits. The dark bars to the right of each group represent group means for that topic type. While we are hesitant to claim too much from this data, it seems clear, for at least these sources and topics and this time period, that there were more stories devoted to Finance, Legal Cases, Elections and Sports than Laws, Accidents, Scandals and Science. This is clearly subject to variation; the Clinton-Lewinsky scandal was one of the largest topics in

TDT-2. We also see outliers in several of the topic types that bias the group mean.

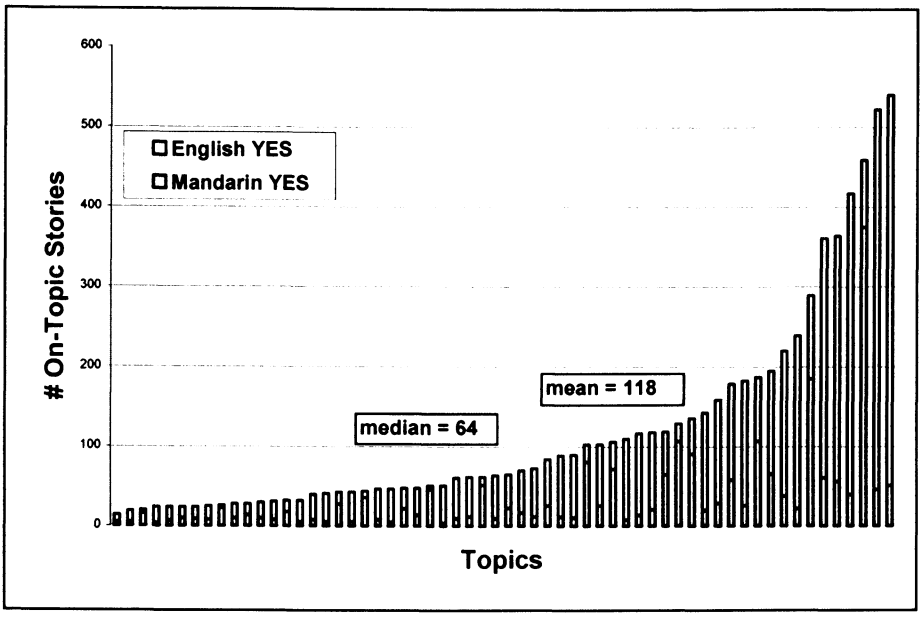


Figure 13. Topics by number of on-topic stories in English and Mandarin.

10. Conclusion

The TDT corpora, developed to support the DARPA-sponsored program in Topic Detection and Tracking, combine data collected over a nine month period (six months for TDT-2, three months for TDT-3) from 8 English and 3 Chinese sources. The published corpora contain audio, reference text including written news text and transcripts of the broadcast audio, boundary tables segmenting the broadcasts into stories and relevance tables resulting from millions of human judgments. Sections of the corpora have undergone topic-story, first story and story link annotation. Both the TDT-2 (LDC Catalog Number:LDC2001T57, ISBN:1-58563-183-3) and TDT-3 (LDC Catalog Number:LDC2001T58, ISBN:1-58563-193-0) text corpora and the accompanying broadcast audio are now broadly available from the Linguistic Data Consortium.

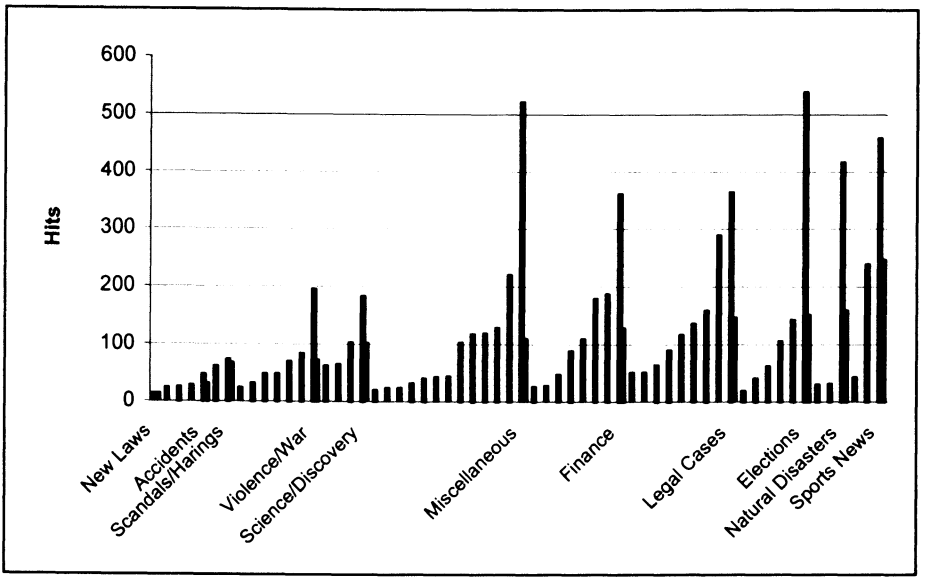


Figure 14. Topics by type and number of on-topic stories.

References

Cieri, Christopher, et al., 2000 Large Multilingual Broadcast News Corpora for Cooperative Research in Topic Detection and Tracking: The TDT2 and TDT3 Corpus Efforts, Proceedings of the Second International Language Resources and Evaluation Conference, Athens, Greece, May 2000.

CLSP - The Johns Hopkins University Center for Language and Speech Processing, 1999, Topic-Based Novelty Detection, http://www.clsp.jhu.edu/ws99/projects/tdt/index.html

Doddington, George, The Topic Detection and Tracking Phase 2 (TDT-2) Evaluation Plan: Overview & Perspective, Proceedings of the Broadcast News Transcription and Understanding Workshop, Lansdowne, Virginia, February 1998.

Doddington, George, 1998, The Topic Detection and Tracking Phase 2 (TDT-2) Evaluation Plan http://www.nist.gov/speech/tdt98/doc/tdt2.eval.plan.98.v3.7.pdf

Garofalo, et. al., 2000, The TREC Spoken Document Retrieval Track : A Success Story, April 2000.

Linguistic Data Consortium, 2000, Topic Detection and Tracking Pages, http://www.ldc.upenn.edu/TDT/

NIST - National Institute for Standards and Technology, 1999, 1999 NIST Broadcast News Evaluation, http://www.nist.gov/speech/tests/bnr/bnews_99/bnews_99.htm

NIST - National Institute for Standards and Technology, 2000, ACE - Automatic Content Extraction, http://www.nist.gov/speech/tests/ace/

NIST - National Institute for Standards and Technology, 2000, The 2000 NIST Hub-5 Evaluation, http://www.nist.gov/speech/tests/ctr/h5_2000/index.htm

NIST - National Institute for Standards and Technology, 2000, Topic Detection and Tracking, http://www.nist.gov/speech/tests/tdt/tdt2000/index.htm

Strassel, Stephanie, et al., 2000), Quality Control in Large Annotation Projects Involving Multiple Judges: The case of the TDT Corpora Proceedings of the Second International Language Resources and Evaluation Conference, Athens, Greece, May 2000.

Wayne, Charles, 1998, Topic Detection & Tracking: A Case Study in Corpus Creation & Evaluation Methodologies, Proceedings of the First International Conference on Language Resource and Evaluation, Granada, Spain, May 1998.

Wayne, Charles, 1998, Topic Detection and Tracking (TDT): Overview & Perspective, Proceedings of the Broadcast News Transcription and Understanding Workshop, Lansdowne, Virginia, February 1998.

Chapter 4

Probabilistic Approaches to Topic Detection and Tracking

Tim Leek, Richard Schwartz, and Srinivasa Sista
BBN Technologies
Speech and Language Processing
Cambridge, MA 02140

Abstract BBN's systems for TDT use probabilistic models for higher accuracy and easy training. They generate measures that are normalized across topics, so that only one threshold is necessary to make decisions. These systems make little or no use of deep linguistic knowledge, and therefore are easy to modify for new languages and domains. At the same time their performance has consistently been in the top tier.

1. Introduction

BBN's most significant contributions to the TDT program have been (1) to introduce an array of effective probabilistic models with which to measure the similarity between a story and a topic, (2) to offer practical solutions to the difficult problem of score normalization, and (3) to field systems that are easy to adapt to new domains and languages.

A sound probabilistic measure for the distance between a story and a topic is the heart of the TDT systems we have deployed. Why is this preferable to an ad-hoc score cobbled together from intuition and experience?

- The predictions of probabilistic models are resilient to missing data and surprise events, even when they appear in quantity, e.g. when the stories are the product of automatic speech recognition (ASR) or automatic translation. Ad-hoc systems with countless fudge-factors break easily when confronted with noise, novel examples, or different domains. It is rarely clear how to fix them.

- We can use standard, well-understood, statistical methods (like Expectation-Maximization or Maximum A Posterior or ...) to estimate *thousands* of parameters in detailed probabilistic models using vast quantities of real data. There are rarely good training procedures for ad-hoc systems; you must resort to parameter-fiddling. [1]

- Because it is based upon a model of the underlying process, it is easy to extend a probabilistic model. It is usually unclear how to refine or extend an ad-hoc system.

- Because their parameters and outputs are probabilites (or derived quantities such as log likelihoods), we can reason about probabilistic models in ways that are useful and diagnostic. It is notoriously difficult to reason about the parameters or outputs of ad-hoc systems; you never really know how they work.

Score normalization is crucial to success in TDT; it is a requirement of the program that the scores for different topics be comparable, that there be only one threshold that works for every topic. The simple methods we have invented (see section 2.2) are fast and effective for all datasets we have so far encountered.

BBN's TDT systems make little or no use of deep linguistic knowledge (see section 3). Those procedures we do employ, removing stop words and stemming, are themselves of questionable value; experiments indicate that they have no effect upon our performance, which has consisently been quite good in evaluations. These two facts, taken together, are a powerful combination: our systems are both strong performers and are easy to modify to work on new domains and languages. This was exactly our experience in the crosslingual task of TDT 1999. Modifying our tracking and detection systems to work on Mandarin was a simple matter at the lowest level and we were thus free to concentrate on more interesting crosslingual modelling and automatic translation issues.

2. Core TDT Technologies

Our approaches to Tracking and Detection tasks are unified. The underlying mathematics are essentially identical in the systems developed for these rather different tasks. Both use the same probabilistic models to measure the similarity between a story and a topic. And both employ the same score normalization, combination, and thresholding techniques. The differences between

[1] A notable exception to this claim would appear to be Neural Networks, for which there exist powerful training procedures. But Neural Networks are not ad-hoc systems. They are best understood as statistical models in disguise. See Chris Bishop's excellent book "Pattern Recognition for Neural Networks" for a good exposition of this link.

the systems are at a higher level, in the algorithms built out of the models and techniques.

2.1 Measures of Story-Topic similarity

Finding a good measure of the similarity between a story and a topic is the cornerstone of both the Tracking and Detection tasks. We have invented four such measures.

Probabilistic Topic Spotting (TS) measure. This measure is based upon our work developing the BBN topic classification system, OnTopic [1]. OnTopic uses an HMM to model the words in a story as being generated by several topics known to be relevant to a document. The HMM consists of one state per topic plus one extra state for the required topic "General English". The emission probabilities for each state are simply a discrete probability distribution on topic keywords, initially taken to be the pooled words in every story on the topic. The transition probabilities in the HMM reflect how likely it is that the next word in the story would be generated by a particular topic. The parameters of this HMM, i.e. the emission and transition probabilities, are re-estimated with an iterative EM-like procedure, using a corpus of stories each annotated with multiple topics.

In TDT, each training story has only two topics, one of which is General English. Given some number of training stories, we use the OnTopic re-estimation procedure to sharpen the topic distribution $P(w|T)$. Finally, the TS story-topic similarity measure used in TDT is a log-likelihood ratio (we work in the log domain to avoid numerical underflow), and it represents how much more probable the topic T is, given the evidence of the particular story, S, than it is for the average story. We use Bayes' rule to derive this in terms of quantities we know how to model, and assume that words are generated independently to obtain

$$M_{TS}(S,T) = \log \frac{P(T|S)}{P(T)} = \log \frac{P(S|T)}{P(S)} \approx \sum_{w \in S} \log \frac{P(w|T)}{P(w)}, \qquad (4.1)$$

where $P(T|S)$ is the probability that T is the topic, given the evidence (all the words) in S, $P(T)$ is the prior probability that T is the topic, $P(S|T)$ is the probability of generating (observing) all the words in S, given that T is the topic, and $P(S)$ is the prior probability of generating all the words in S. $P(w)$ is the probability of word w estimated on some large background corpus and $P(w|T)$ is the probability of word w, given that T is the topic, estimated as described above.

Probabilistic Information Retrieval (IR) measure. This measure is based upon our work developing the BBN IR system [4]. We use the training stories

for a topic to form a large query, Q. Our similarity measure is an estimate of the log posterior probability that a test story is relevant, given the query. Using Bayes' rule to re-write this posterior in terms easier to estimate, we have

$$P(S \text{ is } Rel|Q) = P(S \text{ is } Rel)\frac{P(Q|S \text{ is } Rel)}{P(Q)}. \tag{4.2}$$

where $P(Q)$ is constant across stories in the test. While $P(S \text{ is } Rel)$, the prior probability that a story is relevant, *could* be modeled so that it would differ from story to story, we choose to take it as constant. We can therefore safely compose the IR measure using only the conditional probability of the topic query, Q, being generated (that is, all of the words in all training stories for T being generated), under the hypothesis that S is a story relevant to the query. We construct $P(Q|S \text{ is } Rel)$ as a mixture model, one state generating the words in the query by drawing from the story, according to $P(w|S)$, and the other by drawing from a background corpus, according to $P(w)$. Again, asuming that the words are generated independently and moving to the log domain, we have

$$\begin{aligned} M_{IR}(S,T) &= \log P(Q|S \text{ is } Rel) \\ &\approx \sum_{w \in T} \log(aP(w|S) + (1-a)P(w)), \end{aligned} \tag{4.3}$$

The mixture weight, a, constrained to be $0 \leq a \leq 1$, is estimated using actual IR queries and associated relevance judgements from past TREC (Text Retrieval Conference) evaluations, with the EM algorithm.

Notice that the IR model is, in some sense, the reverse of the topic spotting model. Here, the word distribution is estimated from the test story, and we compute the likelihood of the model generating all of the words in the topic query, i.e. the training. For the TS model, we estimate the word distribution from the training examples for the topic, and compute the likelihood of the model generating all of the words in the test story. This disparity will be addressed in section 2.2.

Probabilistic IR with Relevance Feedback measure. This measure, $M_{RF}(S,T)$, is also based upon our work developing the BBN IR system. The typical IR system employs some form of *unsupervised relevance feedback*. This means that the IR system is used to make a first retrieval, and then the top ranked (by relevance) documents are mined for words with which to augment the query. These new terms are added to the query and weighted by some function of their frequency in the retrieval results. Relevance feedback routinely adds hundreds of terms to a query.

For TDT, we are in the pleasant position of being able to do the equivalent of *supervised* relevance feedback. If we are given four training stories, we

can assume that they are all relevant to some query (which we do not have) and proceed to compose that query using terms chosen from those stories, weighting each as we would in a standard relevance feedback query.

In BBN's IR system, this re-weighting is achieved by refining the mixture weight, a, in equation 4.3. This quantity has a useful interpretation that allows us easily to see how to improve our estimate of it. It can be thought of as the probability that a given word will be found in a test story, under the hypothesis that the story is relevant to the query (topic), i.e. $P(S \text{ has } w | S \text{ is } Rel)$. We model this as the *excess* probability that w is in S that is actually due to the fact that S is Rel,

$$\begin{aligned} a(w) &= P(S \text{ has } w | S \text{ is } Rel) \\ &\approx P(S \text{ has } w | S \text{ on } T) - P(S \text{ has } w), \end{aligned} \tag{4.4}$$

where $P(S \text{ has } w | S \text{ on } T)$ is the probability of w being in S, given that S is a training story for T, and $P(S \text{ has } w)$ is the probability that w will be in any story at all, regardless of whether it is on topic T. For M_{RF}, we estimate $P(S \text{ has } w | S \text{ on } T)$ as the fraction of training stories containing w, and $P(S \text{ has } w)$ as the fraction of stories in some background corpus containing w.[2]

$$\begin{aligned} P(S \text{ has } w | S \text{ on } T) &\approx \frac{N(S \text{ has } w, S \text{ on } T)}{N(S \text{ on } T)}, \\ P(S \text{ has } w) &\approx \frac{N(S \text{ has } w)}{N(S)}. \end{aligned} \tag{4.5}$$

Thus, for $M_{RF}(S, T)$, we still compute the conditional probability of all the words in the topic query, but the model takes into account more information about the features of the individual words.

Probabilistic Word Presence Measure. This measure is intended to capture the notion of *required* words. For many topics, there are words that alone are a necessary (but not sufficient) condition for the topic to be relevant to the story. This is true for topics that are person names, e. g. if the topic is "George Washington" then every story with that topic will be certain to contain the name (or an alias) at least once. If we have enough training stories, then we are more willing to believe the approximation for $P(S \text{ has } w | S \text{ on } T)$ in equation 4.5, and can take the riskier position of forming the likelihood ratio

$$M_{WP}(S, T) = \log \frac{P(Q | S \text{ is } Rel)}{P(Q)} \approx \sum_{w \in T} \log \frac{P(S \text{ has } w | S \text{ on } T)}{P(S \text{ has } w)} \tag{4.6}$$

[2]The BBN IR system, described in [4] uses a more elaborate estimate for $P(S \text{ has } w | S \text{ on } T)$.

Table 4.1. Comparison of various story-topic similarity measures. They differ dramatically in terms of units, range, and length dependence. This underscores the need for careful normalization if these scores are to be combined or compared.

Measure	Units	Range	Length Dependence
M_{TS}	log-likelihood	$-\infty ... +\infty$	Story (test)
M_{IR}	log-probability	$-\infty ... 0$	Topic (training)
M_{RF}	log-probability	$-\infty ... 0$	Topic (training)
M_{WP}	log-likelihood	$-\infty ... +\infty$	Topic (training)

When the number of stories in the topic query Q is small, this is riskier than computing M_{RF}, in particular, since that measure is a mixture with an extremely robustly estimated distribution $P(w)$. For M_{WP}, not only do we *not* smooth the potentially poorly estimated value $P(S \text{ has } w|S \text{ on } T)$, we also divide it by $P(S \text{ has } w)$, which can magnify the contribution of each word dramatically.

2.2 Score Normalization

Table 4.1 compares the various story-topic measures in terms of units, range, and dependence upon the length of (number of words in) either the story or the topic. Clearly these measures have very different qualities and we shall have to find a common scale if we want to compare or combine them. And they must, at least, be comparable. For tracking, we use only two thresholds, one for adaptation (adding stories to the topic model) and another for accepting stories as *on the topic*, so the measures must be comparable both *across stories* in a single tracking run and *across topics*, i.e. across different tracking runs. For detection, the problem is essentially the same. We must choose which, if any, topic (cluster) a story belongs to, so the scores for different topics must be on the same scale. And we must make decisions on different stories using the same threshold, so the scores for different stories must also be comparable.

Using the Off-Topic Score Distribution. It is tempting to normalize the score for each test story using the distribution of scores for the training stories. This is unacceptable for two reasons. First, there are far too few training stories to obtain reliable statistics. Second, since the training stories were used to create the topic model, they typically get a much higher score than any test story ever will.

Our solution to this problem has been to make use of statistics of the scores of *off-topic* stories to scale the scores of test stories, and then use a simple hypothesis test to decide if a story is on or off topic. The procedure is very similar for tracking and detection. Here, we will consider only the conceptually simpler tracking case.

We have a topic, defined only by the list of stories said to be on-topic, and we choose one of the story-topic similarity scores described above. We proceed to compute this score for a large number of stories presumed to be *off-topic*. For tracking, we assume that every story pre-dating the known on-topic stories for a given query is actually off-topic. We assume that these scores will have a roughly gaussian distribution, and estimate the mean, μ_{OFF} and standard deviation, σ_{OFF}. Since there are typically thousands or even tens of thousands of off-topic stories, we can have high hopes that these estimates will be reliable. Normalizing the similarity scores for test stories is then simply a matter of subtracting the mean and dividing by the standard deviation,

$$score_{\text{N}}(S,T) = \frac{score(S,T) - \mu_{\text{OFF}}}{\sigma_{\text{OFF}}} \tag{4.7}$$

This procedure, known as "z-scoring", tries to ensure that the scores for documents on a given topic will have zero mean and unit variance (assuming that most of the test stories are off-topic). This ensures that scores will be comparable *across documents*. And it allows us to use a simple hypothesis test to discriminate between on-topic and off-topic test stories. For instance, we can claim that a test story for which $score_{\text{N}}(S,T) > 3$ has about a 0.001 probability of being off-topic and therefore a 0.999 probability of being on-topic. Figure 4.1 is a histogram of scores for off-topic stories for one tracking topic. We find that on-topic test stories have scores at least $6\sigma_{\text{OFF}}$ higher than the mean μ_{OFF}, while the training stories have scores more like $20\sigma_{\text{OFF}}$ away from the mean.

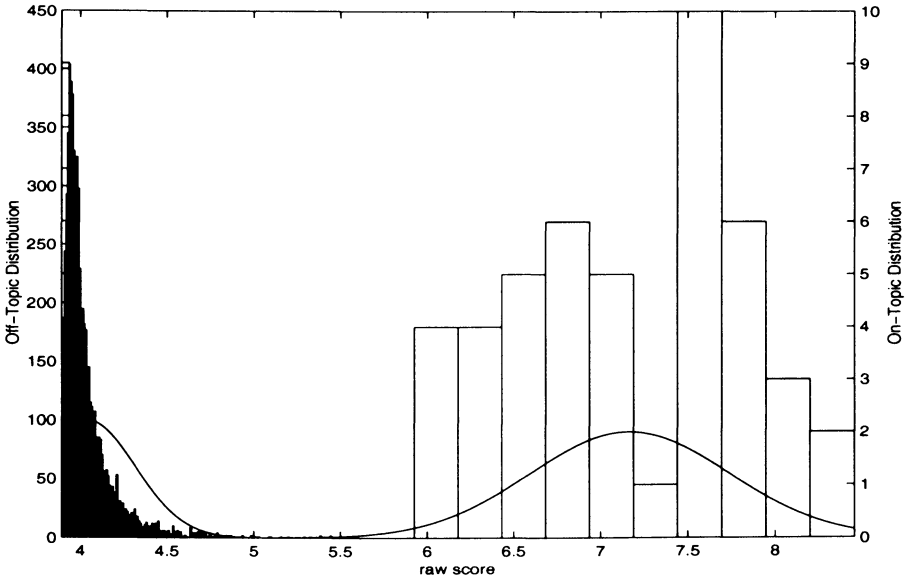
Robust Statistics. This simple normalization procedure can run aground if there are on-topic stories lurking amongst the presumed off-topic ones. In our experience, on-topic story scores are typically more than $10\sigma_{\text{OFF}}$ higher than μ_{OFF}. If they are unknowingly included in the calculation of off-topic score statistics, then μ_{OFF} and σ_{OFF} can be corrupted and compromise the normalization scheme. To obtain more robust statistics, we re-compute the off-topic score statistics, excluding from that calculation any stories that are likely to be on-topic because their normalized scores are more than k standard deviations[3] away from the off-topic distribution, i.e. for which $score_{\text{N}}(S,T) > k$. Once we have more trustworthy estimates for μ'_{OFF} and σ'_{OFF}, we re-normalize using equation 4.7, giving us the robustly normalized $score_{\text{RN}}(S,T)$.

2.3 Score Combination

As we have developed four different measures of story-topic similarity, it was natural to employ standard ways of combining them. Many of the measures

[3] In all evaluations we have used $k = 8.5$, but performance is not particularly sensitive to this parameter.

Figure 4.1. On and Off-Topic story score histograms for one topic. The distribution to the left ($\mu = 4.04$ and $\sigma = 0.156$) consists of 6820 background stories assumed to off-topic. The distribution to the right ($\mu = 7.2$ and $\sigma = 0.65$) consists of 32 stories known to be on-topic. Deleted from this histogram is the large number of scores for stories with the lowest possible score, which represent the score for no words in common between the story and the topic. Note that the off-topic distribution is clearly not gaussian (it looks somewhat *lognormal* but even that is not quite right) But it is close enough to normal for our normalization scheme to work.



use very different evidence and therefore make uncorrelated errors. In this situation, a voting strategy can increase accuracy while reducing variance in the final decision. We use logistic regression to combine the various measures, training the combination weights using development data. Our final system for the tracking tasks is this combination system.

2.4 Thresholding

The thresholding problem is the dual of the normalization problem. We consider this problem essentially solved by score normalization. Choosing an appropriate threshold is something we do once, working with normalized scores and development data. Our normalization schemes are very successful and scores are surprisingly comparable across domains; we have not had to revise our choice of thresholds.

3. Corpus Processing

Our approach to the input stories has been largely hands-off. We find word tokens by splitting text on whitespace and certain punctuation characters. The only operations we perform on the tokens themselves are to conflate case, stem using Porter's algorithm, and ignore anything in a stop list of 400 common words. We attempt no deeper linguistic analysis of the input. In our experience even the superficial transformation of the corpus we perform is likely unnecessary. Disabling it does not materially effect the performance of our TDT systems.

4. Tracking

4.1 System Description

BBN's tracking system is built from the core technologies detailed in the previous section. The various probabilistic story-topic scores are normalized using the techniques described in section 2.2. We use logistic regression as described in section 2.3 to combine scores, and use the final combination score along with *two* threshold values to track a topic over a stream of subsequent test stories. One threshold is for discriminating between stories that are on and off topic, while the other is used for adaptation.

Adaptation. For tracking, it is typically the case that training is limited, e.g. we often have fewer than four example stories for a topic we must track. This is a problem for several reasons. Our main job in TDT is to discover the set of words that are evidence for the topic. However, it is unlikely that this vocabulary is completely specified in only four stories, the maximum number of training examples we are permitted. Furthermore, most topics are not stationary; a

topic evolves over time by changing its vocabulary, both adding *and* subtracting words. Unless we have a way to adapt our models by incorporating evidence from test stories we can be relatively sure are on-topic, our systems will not generalize well, and in fact will perform *worse* as time goes on and the topic vocabulary drifts farther and farther away from its origin. We need to be able to detect and remember, for instance, that "John Doe 1" has been renamed "Timothy McVeigh".

For the tracking task, we add to the topic any stories whose scores meet a threshold, θ_{ADAPT}, that is higher than the accept threshold, θ_{ACCEPT}. For some measures (M_{IR} for instance), adaptation is simply a matter of re-estimating a unigram distribution. For $M_{TS}(S,T)$, we also re-run our EM-like procedure to sharpen the unigram distribution.

Using Time. Presumably, the prior probability that a test story is on-topic decreases as time increases; most topics occupy only a limited span of time. For instance, the prior might be assumed to decay exponentially, as a function of the difference in time between the test story and the first on-topic story. Our tracking systems have included, as a separate score available to the combination system, a time-decay log prior term,

$$M_{TIME}(S,T) = logP(S \text{ on } T) \approx -k(time(S) - time(T)) \qquad (4.8)$$

where $time(T)$ is the time the first training story appeared.

4.2 Results

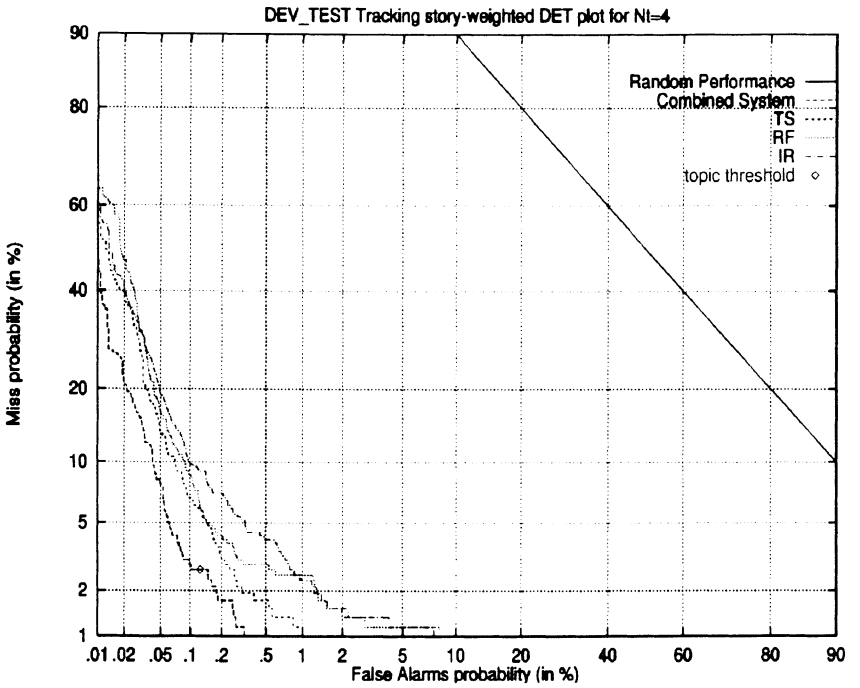
Our Tracking systems perform very well in evaluations. The reasons for this are various: system combination, score normalization, adaptation, and careful modelling all have a role to play.

Figure 4.2 compares the performance, on the TDT 1998 devtest, of tracking systems built upon the various story-similarity measures described in section 2 The combination system dramatically outperforms the individual systems. At about 0.1% false alarm rate, the best individual system had a miss rate of about 6%. The miss rate for the combined system was only 3% for the same false alarm rate. This is a dramatic improvement and is clear evidence that the systems are making uncorrelated errors.

The experimental evidence for the importance of score normalization is similarly clear. A story-topic similarity measure such as M_{IR}, the magnitude of which depends upon the number of training stories for the topic (more precisely, it depends upon the total number of words in all training stories) performs worse than random guessing using the prior (i.e. very badly indeed) if we make no attempt to normalize its scores.

We also see a large and robust gain for our adaptation techniques in tracking. For a system based only upon the M_{TS} measure, we see the miss rate fall from

Figure 4.2. Effect of System Combination: On TDT 1998 development data the performance of the combination system is a factor of two better in terms of miss rate than the best individual system.



18% to 6% when we turn on adaptation, holding the false alarm rate constant at 0.1%.

The performance of our TDT systems when subjected to noise in the form of ASR (automatic speech recognition) is equally impressive, and a testament to the power of probabilistic models. The word-error rate for the ASR was about 23%, and this corruption resulted in no perceivable change in the scores for our tracking system on the TDT 1998 test.

5. Detection

5.1 System Description

BBN's detection system is constructed from the same core technologies as the tracking system. It can use the same array of story-topic similarity measures and employs the same normalization techniques. But the task is quite different, and required significant algorithm development.

Incremental K-means. The task for detection is to cluster the stories in the corpus by topic, where it is stipulated that no story can be on more than one topic. We must form this partition on the stories incrementally, by considering stories as they arrive, in time order, deciding topic membership for each before moving on to the next.

Our detection system uses an incremental version of the K-means clustering algorithm to form this partition. If N is the total number of stories to be clustered, then we have the following statement of the algorithm, in which the final number of clusters is a function only of the similarity measure $M(S,T)$ and the threshold θ_{SELECT}.

```
T_1 ← S_1 ;
C ← 1 ;
for i ← 2 to N
  for j ← 1 to C
    Compute M(S_i, T_j) ;
  k ← ARGMAX_j(M(S_i, T_j)) ;
  if (M(S_i, T_K) > θ_select)
    Assign S_i to topic T_k ;
  else
    C ← C + 1;
    Create new cluster T_C from S_i ;
  Re-estimate all of T_i for i ← 1 to C ;
```

In words, the procedure is to start with a single cluster, T_1, which contains only the first story, S_1. Then compare subsequent stories, S_i, to each of the current clusters, T_j, using some measure of similarity. Use the threshold θ_{SELECT} to decide whether to add the story, S_i, to the cluster it is closest to (T_k, the cluster for which the similarity measure is maximized), or to use it as the seed for a new cluster. Always re-estimate a topic cluster model as soon as its membership changes.

Score Normalization for Detection. The normalization procedure for Detection is slightly different from that in tracking, but serves the same purpose. It is necessary for the similarity measure to be normalized in such a way that we can set the threshold θ_{SELECT} used in deciding whether to merge S_i with its closest topic, or to use it as the seed for a new topic.

Prior to considering any of the test documents, we cluster a background corpus of documents from a similar domain. This gives us C_{back} background topics to use in estimating normalization statistics. In addition to computing the raw similarity measure, $M(S_i,T_k)$, for a given test story and its closest test

cluster, T_k, we also compute the raw scores for that test story and each of the background topics. It is this distribution of background topic scores that we use to estimate μ_{OFF} and σ_{OFF}, which we finally employ, exactly as we do in tracking, in order to normalize $M(S_i, T_k)$. This normalization permits us to use a single threshold in making decisions.

Clustering the background corpus. In order for the detection normalization scheme to work, we must have clustered a background corpus of stories. This would appear to be a chicken-and-egg problem. How can we cluster the background corpus without another background corpus, and so on. We cluster the background corpus exactly as we cluster the test corpus, with incremental k-means. For the first $N = 100$ stories, we do not bother to normalize the similarity scores and use only M_{TS}, which is naturally normalized at least across topics. For $N > 100$, we use the distribution of scores for the current set of *background* clusters to estimate μ_{OFF} and σ_{OFF}.

Look-ahead not useful. There are versions of the detection task that permit *look-ahead*, i.e. that allow us to reverse decisions for stories some fixed time in the past, potentially capitalizing upon an observed shift in topic. We have not found this glance over the shoulder into the past (or peek into the future, depending upon your perspective) particularly useful. Perhaps the topics evolve too slowly for a revisionist approach to help.

5.2 Results

While our detection system is one of the better research systems around, it is far from perfect. For example, in the cross-lingual detection evaluation (using BBN's term translation), our system received a C_{det} of 0.3456. This corresponds to 53% precision, meaning that the average topic cluster is 47% off-topic, and 67% recall, meaning that the average topic cluster is missing about 33% of the stories judged on that topic by the annotators.

Detection is a difficult task; it is essentially like tracking without any training. Errors compound and there is little recourse from early mistakes. Perhaps this is why we have seen far greater sensitivity to high ASR word error-rate for the detection task than for tracking. In the TDT 1998 evaluation we saw the miss rate for a detection task roughly double for using ASR documents.

It is our firm belief that the best way to improve performance on detection is to concentrate on the cleaner technical task of improving monolingual tracking performance.

6. Crosslingual TDT

TDT 1999 was the first year in which there was a required cross-lingual test: English and Mandarin. Our systems required very few modifcations to enable them to work cross-lingually, so we spent some time building a simple translation system for Mandarin using only publicly available domain-specific resources.[4] It was our belief that this was the primary thrust of TDT 1999: how well and how quickly can you adapt your TDT systems to work on an additional language, given limited resources.

We were surprised to discover that our tracking performance using very simple term translation was on-par with the performance if we availed ourselves of translations performed by a commerical machine-translation system. In retrospect this is perhaps not surprising. The noise introduced by translation is most prominent in words that translate *many ways*. These words also tend to be very common ones, unuseful for the purposes of TDT; they are, in fact, typically *stop words*. Extremely specific content words generally translate correctly and few ways. Much of the error introduced by tranlsation is therefore automatically filtered out by corpus pre-processing.

6.1 System Description

Adequate Translation. Our approach to crosslingual TDT was to translate everything into English and then use the monolingual TDT systems, but to make use of the fact that a story has been translated if that should be useful. Our simple translation process works term-by-term. After segmenting a Mandarin story into words[5], we look up each Mandarin word in a bilingual dictionary, and choose the translation with maximum probability. Prior translation probabilities are estimated using a parallel corpus and an iterative procedure [6].

We made some limited use of domain knowledge in order to increase translation coverage of the original Mandarin. Using a list of Chinese surnames, we were able to detect likely person names which we would proceed to *transliterate*, i.e. spell-out using a pinyin dictionary to map from Mandarin to English sounds. This is the proper way to translate person names.

Within-Language Score Normalization. The only modification necessary to use our TDT systems for this cross-lingual task (we consider segmentation and translation to be corpus pre-processing) was to normalize stories from different languages using different statistics. We can't really expect the score for a *translated* Mandarin test story to have the same statistics as an English

[4]Mandarin resources: word segmentation programs, stop list, bilingual dictionary, pinyin dictionary, and parallel corpus.

[5]using segmentation tools for Mandarin built by ourself, previously, and by the LDC.

Training	Translation		
	SYS	BBN	rel increase
E	0.0989	0.1133	+15%
M	0.1122	0.1320	+18%
E+M	0.0800	0.0853	+7%

Table 4.2. Crosslingual Tracking Evaluation Result: Performance is similar for using the SYSTRAN and BBN translations (NB: lower scores mean better performance). Loss for using Mandarin instead of English training is small. There is a gain for adding Chinese training to English. English training is the primary result here.

test story. So, we compute normalization statistics separately for the two languages and always use language-specific normalization parameters on a test story. Specifically, equation 4.7 becomes

$$score_{\text{N}}(S,T) = \frac{score(S,T) - \mu_{L(S),\text{OFF}}}{\sigma_{L(S),\text{OFF}}}, \tag{4.9}$$

where $L(S)$ is the language story S is in.

6.2 Results

Two rather significant discoveries came out of this cross-lingual task. The first is that a very simple term-translation system such as we built appears to be adequate for the purposes of TDT; better translation is quite difficult and appears to provide only small gains. The second is that working crosslingually is nevertheless introducing errors that dramatically decrease performance.

Looking at table 4.2, we see that the BBN tracking system performs almost as well using the simple BBN translation as it does using the output of SYSTRAN, a commercial machine-translation system several years in development; the scores are between 7 and 18% higher (worse) when we use the BBN translation. At the same time, it is certain that working crosslingually *is* introducing errors that are degrading results. We compared the performance of our system on a monolingual task to its performance on a crosslingual task and measured the loss due to be 23-30%.

7. Conclusions and Future Work

BBN's systems for TDT perform in the top-tier of current research systems, are remarkably resilient to noise in the training and test stories, and are easy to adapt to new languages and domains. This does not mean that there is no need for more research on these problems.

The cost measures (C_{track} and C_{det}) that drive evaluations and algorithm development in TDT conceal important details. When we report, for instance,

that in the TDT 1999 evaluation, our cross-lingual tracking system output received a C_{track}[6] score of 0.0989, this does not tell us much about the actual result some real person is likely to try to use, i.e. the set of stories judged to be on-topic by the tracking system. If we examine this set of retrieved stories, we find that it contains 97% of the stories the LDC annotators found to be on the topic (good), but that 78% of the stories in this set are off-topic (bad). In other words, the off-topic stories outnumber the on-topic stories 4 to 1 in the result. This is far from perfect. The mismatch between recall (97%) and precision (22%) here is partly due to an unhealthy weight in the C_{track} measure, which our optimization programs dutifully sought to exploit in order to obtain the best performance, and partly a reflection of the fact that there is still real work to be done here. With focused research on the core TDT algorithms (preferably on the monolingual tracking problem, specifically), we expect to be able to reduce this unacceptably high *perceived* false alarm rate without increasing the miss rate.

8. Acknowledgments

This work was supported by the Defense Advanced Research Projects Agency and monitored by Ft. Huachuca under contract No. DABT63-94-C-0063 and by the Defense Advanced Research Projects Agency and monitored by NRaD under contract No. N66001-97-D-8501. The views and findings contained in this material are those of the authors and do not necessarily reflect the position or policy of the Government and no official endorsement should be inferred.

References

[1] R. Schwartz, T. Imai, L. Nguyen, and J. Makhoul, "A maximum Likelihood Model for Topic Calssification of Broadcast News," in *Proc. Eurospeech*, Rhodes, Greece, September, 1997.

[2] F. Walls, H. Jin, S. Sista, and R. Schwartz, "Topic Detection in Broadcast News," in *Proceedings of the DARPA Broadcast News Workshop*, Herndon, Va, 1999.

[3] H. Jin, R. Schwartz, S. Sista, and F. Walls, "Topic Tracking for Radio, TV Broadcast, and Newswire," in *Proceedings of the DARPA Broadcast News Workshop*, Herndon, Va, 1999.

[4] D. Miller, T. Leek, and R. Schwartz, "A Hidden Markov Model Information Retrieval System," in *Proceedings of the ACM Sigir '99*.

[6] NB: these results are not comparable with those from TDT 1998 since the cost functions changed drastically from one evaluation to the next.

[5] S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M.Gatford, "Okapi at TREC-3." in D. K. Harman, editor, Proceedings of the Third Text Retrieval Conference (TREC-3), NIST Special Publication 500-226 (1995).

[6] T. Leek, S. Sista, R. Schwartz, "The BBN Crosslingual Topic Detection and Tracking System", Topic Detection and Tracking Workshop paper, 1999, http://www.nist.gov/TDT/tdt99/papers.

Chapter 5

Multi-strategy Learning for Topic Detection and Tracking

A joint report of CMU approaches to multilingual TDT

Yiming Yang, Jaime Carbonell, Ralf Brown, John Lafferty,
Thomas Pierce, and Thomas Ault
School of Computer Science
Carnegie Mellon University (CMU)
Pittsburgh, PA 15232

Abstract This chapter reports on CMU's work in all the five TDT-1999 tasks, including segmentation (story boundary identification), topic tracking, topic detection, first story detection, and story-link detection. We have addressed these tasks as supervised or unsupervised classification problems, and applied a variety of statistical learning algorithms to each problem for comparison. For segmentation we used exponential language models and decision trees; for topic tracking we used primarily k-nearest-neighbors classification (also language models, decision trees and a variant of the Rocchio approach); for topic detection we used a combination of incremental clustering and agglomerative hierarchical clustering, and for first story detection and story link detection we used a cosine-similarity based measure. We also studied the effect of combining the output of alternative methods for producing joint classification decisions in topic tracking. We found that a combined use of multiple methods typically improved the classification of new topics when compared to using any single method. We examined our approaches with multi-lingual corpora, including stories in English, Mandarin and Spanish, and multi-media corpora consisting of newswire texts and the results of automated speech recognition for broadcast news sources. The methods worked reasonably well under all of the above conditions.

1. Introduction

Topic Detection and Tracking consists of a set of functionally interrelated tasks, described earlier in this book, and summarized here:

- *Segmentation:* An incoming broadcast news-stream must be segmented into individual, topically-coherent stories. The underlying task is the placement of story boundaries with minimal temporal error.
- *First-story detection:* The onset of newly-breaking news should be signaled, optimally by detecting the first story on a new topic or topic. The underlying task is a pure detection one, minimizing false alarms and misses.
- *Topic detection:* Given all the stories reported in a time window, group the stories into topically-coherent clusters corresponding to individual topics or topics. The underlying task is one of clustering stories, primarily by content similarity but taking into account other factors such as temporal proximity.
- *Topic tracking:* Given one or more news-stories on a given topic or topic, find each future one on that topic or topic as it is reported. The underlying task is supervised learning to induce a classifier based on minimal positive training data (the one or more examples) and much larger sets of negative training data (past history).
- *Story-link detection:* Find all pairs of stories that are linked to each other, primarily by sharing a common topic or topic. The underlying task is one of accurate similarity assessment, but the task could change if different linking criteria were given.

Carnegie Mellon University developed and fielded a variety of methods for these tasks, as described in sections below corresponding to each task. This chapter focuses more on recent CMU results, especially on multi-classifier tracking and story-link detection; previous results have been reported in the literature[21][20]. In general, each of the three years of TDT research witnessed progress in terms of increasing sophistication of methods used, more challenging data sets, and better or more robust overall results.

Each year of active TDT research produced new training and testing corpora, starting from TDT pilot study[1] (which consisted of monolingual newswire data) and continuing through TDT-1998 and TDT-1999 (which consisted of English and Mandarin data from newswire and broadcast news sources). Moreover, CMU labeled Spanish data collected contemporaneously with TDT-1998 and TDT-1999, but not part of the "official" TDT program, and we report on results obtained from that data as well. Although the topics labeled in the Spanish data are the same ones as in the English and Mandarin, the reporting periods do not coincide, making exact comparisons difficult. Nonetheless, Spanish results, as reported below, are in the equivalent-to-better range, compared to English and Mandarin ones. Each set of results reported herein is labeled according to the data set(s) used, although no additional system tuning was done for Spanish.

2. Segmentation

The CMU segmentation system used for TDT-1999 employed discriminative statistical models, treating each inter-sentence position as a binary classification problem. The modeling framework used is closely related to stepwise logistic regression, with features selected to yield the greatest increase in training data likelihood.

first story的界定

The TDT-1999 system was very similar to CMU's TDT-1998 system, and is presented in detail in [3]. As described in this paper, the basic approach is to train a model that uses both "topicality" features to detect shifts in the subject of the story, and "cue-word" features to model the style of language that is characteristic of the beginning and end of a story. The language models used were trained on 10 million words of CNN data from the years 1992–1994. The "cue-word" features depended only on the presence of individual words in the neighborhood of a candidate boundary.

The set of features used by the language models is quite large, so we reduce the sizes of sets of topicality and cue-word features by feature selection. The candidate set for feature selection made use of all of the fields in the TDT-1999 ASR-text files except for the confidence score. Thus, the features incorporated information about the individual words, the speaker cluster id, duration of periods where the recognizer did not generate any output, and the source of the story. Story boundaries were only hypothesized where there was a silence annotation in the ASR-text files. Feature selection was done using the January-April subset of the TDT-1999 October dry-run corpus with the May-June subset held out as validation data; no additional training data were used. In contrast to the results reported in [3], no improvement was obtained by combining the logistic model with a decision tree.

The segmentation results are presented in Figure 5.1. We note that IBM reported significantly better scores than these in the official evaluation, achieving an normalized segmentation cost of 0.3857 on the same test data.

Source	# Files	*P(miss)*	*P(false)*	Cseg	Cseg (norm)
ABC_WNT	76	0.1234	0.1549	0.0695	0.3311
CNN_HDL	348	0.1604	0.2707	0.1050	0.4998
MNB_NBW	51	0.1877	0.1568	0.0892	0.4248
NBC_NNW	87	0.2170	0.1992	0.1069	0.5092
PRI_TWD	65	0.1234	0.2193	0.0831	0.3957
VOA_ENG	103	0.1724	0.2798	0.1105	0.5261
Sums	730	0.1620	0.2345	0.0978	0.4659
Means	121	0.1640	0.2134	0.0940	0.4478

Figure 5.1. CMU TDT-1999 segmentation results on speech-recognized English broadcast news, with a deferral period of 10,000 words.

3. Topic and Event Tracking

CMU developed several different methods for tracking topics and events, including several versions of k-nearest-neighbors (kNN), decision-trees (Dtrees), a Rocchio variant, and language modeling methods (LM). Additionally, these methods were combined into BORG (Best Overall Results Generator), yielding higher performance than any individual method, as described below. Each method has some inherent parameters (e.g. k in kNN or the mixing-coefficient in LM's), and their optimal values differed with the training set. Hence, producing near-optimal performance on held-out test sets with different properties (e.g. different topics from different sources in different media) presented a real challenge.

3.1 Parameter optimization problem

Parameter tuning has been a tough and puzzling problem in TDT evaluations. Systems tuned on dry-run collections have often had disappointing performance when the results on evaluation collections were released; parameters optimized for previous topics are often not optimal for the detection and tracking of later topics. Although one can question the consistency of human judgments in the topic identification and story labelling process when the TDT corpora were constructed, we believe that the difficulty in parameter optimization for TDT systems comes from the tough conditions of the TDT tasks: the wide range of subjects covered by the TDT topics, their short duration by nature, the dynamic evolution of a topic over time, and the very small number of positive training examples per topic in tracking. As a result, it is difficult to optimize parameters in TDT systems to produce consistent performance across topics in different time periods. These together make TDT a tougher problem from a statistical learning point of view than conventional text classification problems where classes are larger and more stable over time.

Our approach to reducing performance variance in TDT is to combine the output of multiple classifiers with different learning strategies[21, 20]; we name the resulting system BORG (Best Overall Results Generator). We have conducted a set of experiments comparing BORG with the single-method classifiers on the corpora (or subsets thereof) for the TDT pilot study and the TDT-1999 October dry-run under the following conditions:

- tune and evaluate a system on the same data collection (but on separate subsets of that collection), and

- tune each system on one collection and evaluate on the other collection

where the two collections are temporally and topically disjoint data sets, e.g., the TDT pilot corpus and the TDT-1999 dry-run corpus. Moreover, the TDT pilot

corpus includes only newswire data, whereas TDT-1999 includes newswire and uncorrected automatically-recognized speech from broadcast news sources.

These experiments allow us to observe the performance variance of the systems under intra-corpus and inter-corpus conditions. The intra-corpus condition corresponds to the evaluation scenario in the TDT pilot study, where a system's parameters are tuned and evaluated on the same corpus for which complete corpus statistics and relevance judgements are known. The inter-corpus condition corresponds to the TDT-1998 and TDT-1999 evaluation scenarios, where parameter tuning is performed on a retrospective corpus (such as the TDT-1999 October dry-run corpus) for which complete corpus statistics and relevance judgements are known but evaluated on a corpus for which neither complete corpus statistics nor relevance judgements are available (such as the TDT-1999 evaluation corpus). The tracking methods and the experimental results are described in the following sections.

3.2 Rocchio

Rocchio is a common approach in information retrieval[8, 13]. It uses a vector to represent each class and story, computes their similarity using the cosine value of these two vectors, and obtains a binary decision by thresholding on this value. The vector representation for a story consists of the weights of terms (words or phrases) which occur in it. In this study, we use a common version of the TF-IDF scheme[13] for term weighting, defined to be

$$w(t,d) = \frac{(1+\log_2 tf(t,d)) \times \log_2(N/n_t)}{\sqrt{\sum_{t' \in d} w(t',d)^2}}$$

$w(t,d)$ is the weight of term t in story d;

$tf(t,d)$ is the within-story term frequency (TF);

$\log_2(N/n_t)$ is the Inverted Document Frequency (IDF);

N is the number of stories in the training set;

n_t is the number of training stories in which t occurs.

The vector representation for a class, called the *prototype* or *centroid*, is constructed using a set (R) of positive training examples (u_i) and a set ($\bar{R}$) of negative training examples (v_i) of that class. We use a variant of Rocchio where $\bar{R}$ consists of the n top-ranking stories ("the query zone"[14]) retrieved from the negative training examples when using the centroid (vector sum) of the positive training examples as the query. The prototype vector is defined to

be:

$$\vec{c}(\gamma, n) = \frac{1}{|R|}\sum_{i}^{|R|}(\vec{u}_i \in R) + \gamma\frac{1}{n}\sum_{i}^{n}(\vec{v}_i \in \bar{R}_n) \tag{5.1}$$

where $\bar{R}_n$ is the "query zone". The scoring function for a test story with respect to the class is:

$$f_{roc}(\vec{x}|\vec{c}, \gamma, n) = cos(\vec{x}, \vec{c}) \tag{5.2}$$

The pre-specified parameters in the Rocchio method are n (the size of the local zone), γ (the weight of the negative centroid) and t, the decision threshold.

3.3 kNN

The kNN method is an instance-based classification method. In contrast to Rocchio, which uses a static centroid per class, kNN uses the training stories "local" to each test story to classify it. We have developed two new variants of kNN to address the potential problems that arise when the number of positive training examples is extremely small – a typical situation in topic tracking.

These new variants, namely kNN.avg1 (Formula 5.3) and kNN.avg2 (Formula 5.4), use the following scoring functions for each test story:

$$f_{avg1}(\vec{x}|k) = \frac{1}{|P_k|}\sum_{\vec{u}\in P_k} cos(\vec{x}, \vec{u}) - \frac{1}{|Q_k|}\sum_{\vec{v}\in Q_k} cos(\vec{x}, \vec{v}) \tag{5.3}$$

$$f_{avg2}(\vec{x}|kp, kn) = \frac{1}{|U_{kp}|}\sum_{\vec{u}\in U_{kp}} cos(\vec{x}, \vec{u}) - \frac{1}{|V_{kn}|}\sum_{\vec{v}\in V_{kn}} cos(\vec{x}, \vec{v}) \tag{5.4}$$

where stories $\vec{x}$, $\vec{u}$ and $\vec{v}$ have the same meaning as for Rocchio; P_k (Q_k) is the set of the positive (negative) instances among the k nearest neighbors of $\vec{x}$ in the training set D; U_{kp} consists of the kp nearest neighbors of $\vec{x}$ among the positive stories in the training set; and V_{kn} consists of the kn nearest neighbors of $\vec{x}$ among the negative stories in the training set. Binary decisions are obtained by thresholding on f_{avg1} and f_{avg2}.

kNN.avg1 is similar to the conventional versions of kNN ("knn.sum") where a weighted sum of the scores from the nearest-neighbors training stories is computed for a class label, but kNN.avg1 computes the average instead of the score sum. This modification allows the parameter k to be large without

permitting negative examples to dominate the classification decisions all the time. kNN.avg2 has even greater differences from conventional kNN than kNN.avg1. Instead of using one zone (k nearest neighbors) local to each test story, it uses two zones (kp positive nearest neighbors and and kn negative nearest neighbors), guaranteeing that the system uses both positive and negative examples to score the test story, even if the similarity radius of zone kp is larger than that of zone kn. Thus the system will not lose any discriminatory power when using small local zones that may otherwise exclude the sparser positive instances.

The new variants of kNN significantly improved the performance of our original kNN system ("kNN.sum"), as shown in Section 3.6. However, the two variants have different performance characteristics: one tends to produce high-precision results while the other yields better recall benefits. Their error trade-off patterns are to the choices for the values of k, kp and kn, the pre-specified parameters in these methods.

3.4 Language Modeling

A

Various forms of Language Modeling (LM) have been applied to TDT, including the KL-divergence based clustering approach by Dragon Systems Corporation, the 2-state Hidden Markov Models (HMM) by BBN, the exponential LM and the hierarchical LM using deterministic annealing by Carnegie Mellon University (CMU) [17, 16, 7, 15, 4]. For the study in this paper, we implemented the BBN Topic Spotting (BBN/TS) approach to topic tracking as an additional method to Rocchio and kNN, so that we can test our hypothesis about using diverse classifiers to reduce performance variance of combined system in topic tracking. Our results in this paper should not be interpreted as the exact results of BBN's topic tracking systems; there are potential differences in implementation and parameter tuning strategies.

The BBN/TS method is essentially a Naive Bayesian classifier using a specific form of smoothing using a Expectation Maximization (EM) algorithm; detailed descriptions can be found in BBN's papers[16, 7, 15]). The scoring function for test story D with respect to class C is defined to be

$$f_{lm}(D|C,\lambda) = \log P(C) + \sum_{w_j \in D} \left[\log \frac{P'(w_j|C)}{P(w_j)} \right] \tag{5.5}$$

$$P'(w_j|C,\lambda) = \lambda P(w_j|C) + (1-\lambda) P(w_j) \tag{5.6}$$

A binary decision is obtained by thresholding on this score. The parameters tuned in this method are the choice of λ and the decision threshold.

3.5 BORG.track

system combination

A combination system, namely BORG.track (Best Overall Results Generator for tracking), is constructed as the following:

1 Run each classifier (Rocchio, kNN.avg1, kNN.avg2 and LM) with different parameter settings, resulting in a set of system-generated scores and a DET curve per run.

2 Select the runs whose DET curves are either globally optimal, or significantly better than other runs in a local region, e.g., for low false alarm rate (high precision), low miss rate (high recall), or minimized cost. Allow more than one run to be selected for a classifier, if needed.

3 Combine the system output of selected runs by first normalizing the scores by each system and then compute the sum of the scores of multiple runs per test story. The normalization formula is

$$x' = \frac{x - \mu}{s.d} \tag{5.7}$$

where x is the original score, μ is the mean of the scores for the stories up to and including (but not past) the current story in the run, and $s.d.$ is their standard deviation. This results in a set of scores of BORG.track; re-normalize these scores in the same way.

It is possible to give different weights to the individual component systems in the combination. In this paper, we use an equal weight for all the classifiers for simplicity. The only pre-specified parameter in BORG.track is the threshold for binary decisions.

3.6 Tracking Results

The performance measure (TDT-1998 and TDT-1999 official) is the *Cost* which is a linear combination of two kinds of errors and is defined to be:

$$C_{trk}(E_j) = 1 \times 0.02 \times m_{trk}(E_j) + 0.1 \times 0.98 \times f_{trk}(E_j) \tag{5.8}$$

where E_j is the jth topic, m_{trk} is the miss rate, and f_{trk} is the false alarm rate in tracking. The per-topic costs are then averaged over all topics ("topics") to obtain a global measure, namely the *topic-weighted average tracking cost*, or simply notated as C_{trk}.

Table 1 summarizes the evaluation results of the individual classifiers and BORG.track. Under the inter-corpus condition AB, we obtained a 58% reduction in C_{trk} by using BORG.track instead of the best single-method classifier

(Rocchio with $\gamma = 2$ and $n = 200$) on the TDT pilot study. Under the inter-corpus condition BA, the C_{trk} reduction was 7-13% when using BORG.track instead of the best single-method classifier (kNN.avg1 with $k = 2000$ or kNN.avg2 with $kp = 4$ and $kn = 2000$) on TDT-1999.

Table 5.1. C_{trk} of classifiers evaluated on the corpora for the TDT pilot study and the TDT-1999 dry-run

System (parameters tuned on pilot corpus)	AA	δ	AB	δ
kNN.sum ($k = 3$)	.0056	+52%	.0085	+71%
kNN.avg1 ($k = 5$)	.0033	+18%	.0063	+60%
kNN.avg2 ($kp = 4, kn = 0$)	.0030	+10%	.0076	+67%
Rocchio ($\gamma = -2, n = 200$)	.0022	-23%	.0060	+58%
LM ($\lambda = 0.025$)	.0035	+23%	.0045	+44%
BORG (combining above four)	.0027	-	.0025	-
System (parameters tuned on TDT-1999)	**BB**	δ	**BA**	δ
knn.sum ($k = 3$)	.0080	+68%	.0080	+65%
kNN.avg1 ($k = 2000$)	.0023	-13%	.0030	+ 7%
kNN.avg2 ($kp = 4, kn = 2000$)	.0023	-13%	.0032	+13%
Rocchio ($\gamma = -.25, n = 200$)	.0026	+0%	.0033	+15%
LM ($\lambda = 0.25$)	.0040	+35%	.0040	+30%
BORG (combining above four)	.0026	-	.0028	-

AA: tuned and tested on TDT pilot corpus;
AB: tuned on pilot corpus and tested on TDT-1999 dry-run corpus;
BA: tuned on TDT-1999 dry-run corpus and tested on TDT pilot corpus;
BB: tuned and tested on TDT-1999 dry-run corpus;
δ: Increase in C_{trk} by using the individual classifier instead of BORG.

Perhaps the most important point is the combination of high overall performance and small performance variance exhibited by BORG.track between intra-corpus ("AA" and "BB") and inter-corpus ("AB" and "BA") conditions. Although our Language Modeling showed cross-corpus variances comparable to BORG.track (other methods had much larger performance variances), its actual performance was much worse than the BORG.track (C_{trk} of 0.0045 vs. 0.0025). Thus, by choosing BORG.track over the best-performing single method in validation, one is much more likely to "win" during evaluation on a different data set with potentially different characteristics than the training set.

Another observation from Table 1 is the much improved performance of kNN.avg1 and kNN.avg2 over the original kNN (kNN.sum) for topic tracking. The kNN.avg1 method showed a 26% and 63% improvement over kNN.sum under the "AB" and "BA" simulated evaluation conditions (e.g. tuned on one of the TDT pilot study or TDT-1999 corpora and evaluated on the other) respectively. Likewise, the kNN.avg2 method showed improvements of 11% and 60% under the same "AB" and "BA" conditions.

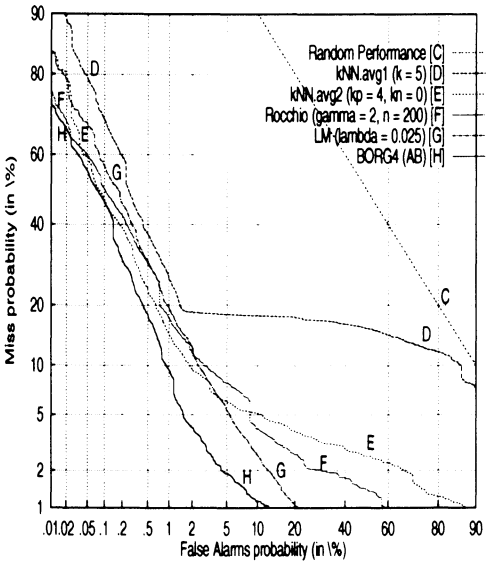


Figure 5.2. Classifiers evaluated under the AB condition

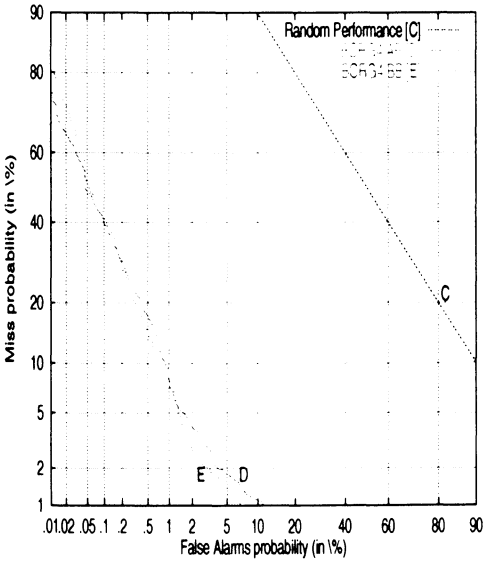


Figure 5.3. BORG evaluated under AB and BB conditions

Figure 5.2 shows the DET curves of all the methods under the AB condition; amazingly, BORG.track outperformed all the single-method classifiers in the entire DET space, integrating the good parts of all the single-method classi-

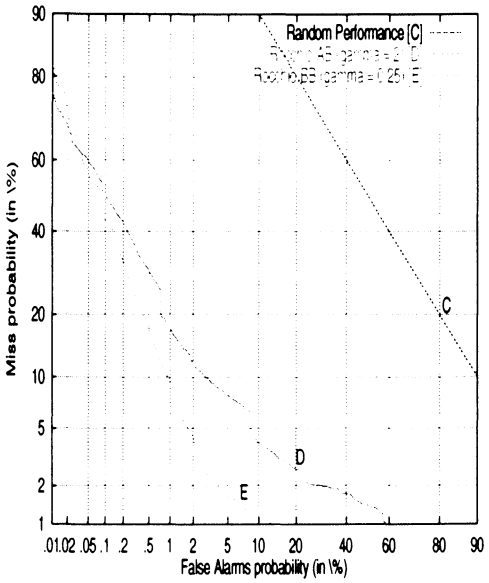


Figure 5.4. Rocchio evaluated under AB and BB conditions

fiers. This suggests that the "bad" scores generated by individual classifiers are highly uncorrelated on individual stories, thus the combined scores are globally improved. Figures 5.3 and 5.4 compare the performance degradation of BORG with Rocchio under the conditions of AB (tuned on TDT pilot study corpus and tested on TDT-1999 Dry-run corpus) and BB (tuned and tested on TDT-1999 Dry-run corpus), respectively. While Rocchio had the best performance among the single-method classifiers under the AA condition, its performance degradation when moving to the AB condition is much larger than the corresponding degradation in BORG. In other words, when optimizing parameters in the validation phase and using these parameters in the evaluation phase, BORG's performance was more stable than Rocchio's. We observed similar pattern with other tracking methods as well (Figure 5.5, for example).

To summarize our experiments, we adapted several supervised learning algorithms to topic tracking, and addressed the parameter tuning difficulty using a combination of classifiers with diverse learning strategies and performance characteristics. We examined this approach using the corpora for the TDT pilot study the TDT-1999 dry-run in which the topic sets do not overlap. We observed strong evidence for the effectiveness of BORG in topic tracking, reducing 38-65% in C_{trk}, compared to using the individual classifiers alone.

BORG is not the first system to combine classifiers to improve results. The benefits of combining systems have been studied in various fields, including

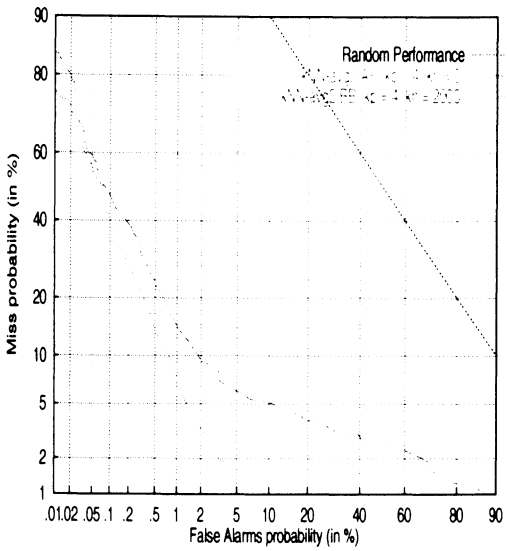


Figure 5.5. kNN.avg2 evaluated under AB and BB conditions

information retrieval[9, 2, 11], speech recognition[5] and text categorization [10, 6]. In the TDT domain, BBN's tracking method combining three language models (including the one we re-implemented) for the TDT-1998 evaluations [7] is the closest to our work in this paper. Nevertheless, all of this work has been focussed on improving the performance of the combination system on a fixed set of classes. Our work with BORG is the first to explicitly identify the problem of cross-class parameter tuning and evaluate the effectiveness of combining classifiers as a solution to this problem. Further discussion of this problem, the new kNN variants, and the BORG approach can be found in our SIGIR and ICML papers[19, 18].

For future research, we would like to study this approach with a larger number of diverse classifiers, including support vector machines, neural networks. etc.

4. Topic Detection

For topic detection, we combined agglomerative clustering and single-pass clustering with different term-weighting schemes (TF-IDF and language-modeling based) as described blow.

4.1 GACIncr and Incr.VSM

The GACIncr system is a two-phase online clustering system. In the first phase, the system processes arriving stories in a *look-ahead* window corre-

sponding to the TDT *deferral period*. A agglomerative clustering algorithm named *Group Average Clustering* can optionally be applied, resulting in a set of clusters in this window; details of GAC clustering based on time proximity and content similarity of stories is described in our previous papers[21, 20]. If the GAC option is turned off, then the look-ahead window is treated as a set of "singleton" clusters with one story in each cluster. In the second phase GAC-Incr, each cluster in the look-ahead window is compared with all the previously seen clusters in the *look-back* window: it is either merged to the closest cluster in the past – if their similarity is above a pre-specified threshold – or it is added to the past as an individual cluster (representing a new topic). Clusters in the look-back window are periodically pruned to remain a constant size (in terms of the time span) of the window, based on a time-stamp parameter.

A cluster or story (treated as an singleton cluster) is represented using a vector of term weights for which we use the TF-IDF term weighting scheme described in Section 2.1. The IDF statistics are initialized using a historical collection (the TDT pilot study corpus, for example), and are updated throughout a run to reflect data that arrives into the look-ahead set. The similarity between two clusters is measured using the conventional cosine similarity, i.e., the cosine value of the angle between two vectors.

When generating the results using GACIncr for our TDT-1999 submissions (as well as other results reported for GACIncr in this paper), the GAC option was turned off, resulting in only singleton clusters in the look-ahead window. We call this system with such an option "Incr.VSM" where VSM stands for Vector Space Model.

4.2 Incr.LM

The Incr.LM system works almost identically to the GACIncr or Incr.VSM system; the primary differences are a lack of clustering in the look-ahead window and the comparison criteria for merging stories into clusters in the look-back window. In Incr.LM, clusters are represented as *language models*- vectors of estimated probabilities of words in the on-target stories for a given topic. These probabilities are estimated through an expectation-maximization algorithm on the member stories of a cluster and smoothed with a background model whose word distribution is obtained from historical data (the training set) and all the test stories seen so far. When a new story in the look-ahead window is evaluated, the system computes a likelihood score for each model in the look-back window that the new story was generated by that model and compares these scores to a pre-specified threshold. Stories whose likelihood scores exceed the threshold are merged into the cluster with the highest score, while those that fall below the threshold are placed into the look-back set as a new (singleton) cluster.

4.3 BORG.det

The BORG.det system incorporates both the GACIncr and Incr.LM methods. Currently we are using a very simple voting scheme, i.e., a Boolean "or" operation, though more sophisticated algorithms are certainly possible and may yield better results. Each method runs using its own distinct threshold. Whenever a hard decision on a cluster in the look-ahead set is required, each method votes on whether to combine it with an existing member of the look-back set, or create a new entry. If *either* method votes to combine with an existing member, this action is taken. Otherwise, a new member is added to the look-back set.

4.4 Detection Results

Table 2 summarizes the results of our detection systems on the October dry-run and official evaluation corpora for TDT-1999; the performance scores are normalized topic-weighted detection costs (C_{det}). Table 3 and figure 5.6 summarize the performance of our systems and the results of other top-performing detection systems in the official evaluation of TDT-1999.

Table 5.2. Results of CMU's Detection Systems on Different Collections

SYSTEM	opt-threshold on dry-run99	C_{det} on dry-run99	C_{det} on eval99
Incr.VSM	cos = 0.03	0.2154	0.3044
Incr.LM	log p = -6.95	0.1250	0.2792
BORG.det = Incr.VSM+LM	cos = 0.1 log p = -6.9	0.1430	0.3093

Table 5.3. Results of Top-performing Detection Systems on Eval'99 Set

System	Date Submitted	C_{det}
CMU-1 official (Incr.VSM)	(Dec-20-99)	0.7233
CMU-1 fixed (Incr.VSM)	(Dec-28-99)	0.3044
CMU-1 improved (Incr.LM)	(Feb-00)	0.2792
CMU-1 BORG.det	(Feb-00)	0.3093
IBM official	(Dec-20-99)	0.2645
UMass official	(Dec-20-99)	0.3023
BBN official	(Dec-20-99)	0.3368

Incr.LM had a better performance than Incr.VSM on both corpora, while BORG.det (combining Incr.VSM and Incr.LM) had a performance falling between the other two on the dry-run corpus and the worst performance on the

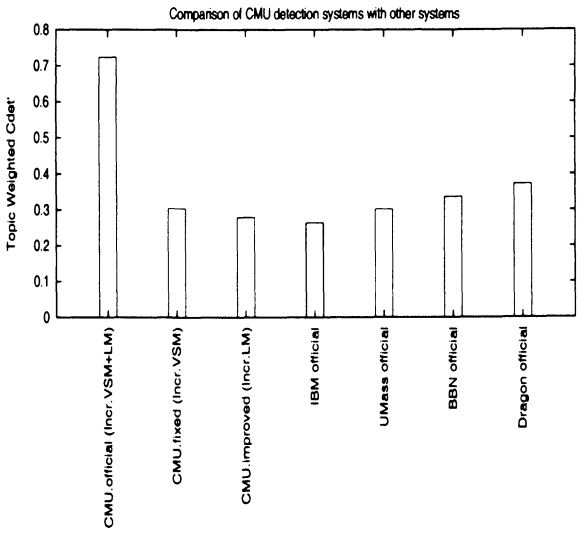


Figure 5.6. Results of detection systems on TDT-1999 evaluation corpus

evaluation corpus. In other words, we did not obtain the kind of performance improvement with BORG in detection compared to what we obtained in tracking. It is possible that the voting scheme (essentially a Boolean "or") in BORG.det is too simple, and that its parameters have not been tuned sufficiently to produce optimal results. Figure 5.7 shows how BORG.det's parameters affects its performance. Our tracking experiments also suggest that BORG.det may also benefit from employing a more sophisticated combination scheme, e.g., using the weighted sum of system-generated scores. Further research is required to fully understand the behavior and relative utility of these results-combination methods.

5. First Story Detection

We define the novelty of a story as the dis-similarity to its nearest-neighbor cluster in the the look-back window ("Past"):

$$\text{novelty}(\vec{d}|\text{Past}) = 1 - max_{\vec{c}\in\text{Past}}\{cos(\vec{d},\vec{c})\}$$

We use the Incr.VSM system to compute the novelty of each new story. If the novelty score of a story exceeds a pre-specified threshold, then it is declared as the first story of a new cluster (topic).

While the underlying system (Incr.VSM) is the same for both First Story Detection ("FSD") and for topic detection ("EDT"), the objectives and opti-

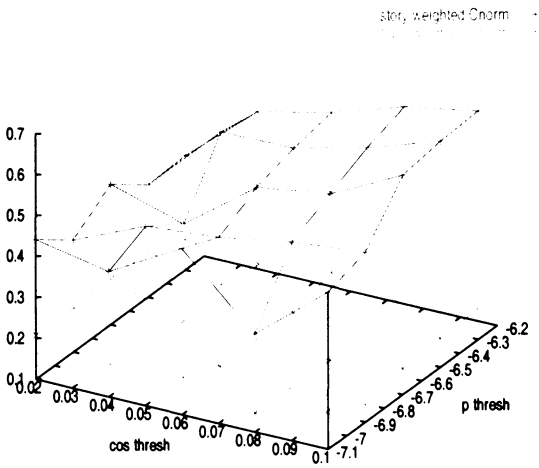


Figure 5.7. Joint parameter tuning in BORG

mization criteria in these two tasks are different. For EDT, the resulting clusters by the system should correspond to topics, meaning that the expected size of a cluster should be comparable with the expected size for an topic (about 350 stories per topic on average in TDT-1999 corpus). For FSD, on the other hand, reliable discrimination between first-story and new topic and background stories or following stories is all that matters; it is irrelevant whether or not it can reliably generate clusters corresponding to topics.

Our experiments in the TDT pilot study showed that allowing clusters to grow in the look-back window usually decreased the performance of FSD[21]. Our new experiments on the TDT-1999 dry-run corpus yielded a consistent observation as shown in Table 5.4 and Figure 5.8. Clearly, the FSD performance was inversely related to the granularity of clusters in the look-back window, and as an extreme case, without clustering stories at all the method worked better than allowing clusters to grow. Our interpretation of this phenomenon is that when clustering is disabled in the look-back window, the system is forced to compare each new story to *every* story in the past instead of comparing it with the *centroid* of a cluster of stories, and that requiring a new story to be sufficiently different from *all* the past stories is much stronger a condition than requiring it to be different from the closest cluster centroid.

Table 5.4. Incr.SVM on FSD-dryrun'99 Set (63,618 stories)

clustering threshold	#clusters	min C_{fsd} (normalized)
1	62871	0.645
0.5	49630	0.65
0.2	18935	0.69
0.1	5757	0.82
0.03	799	0.95

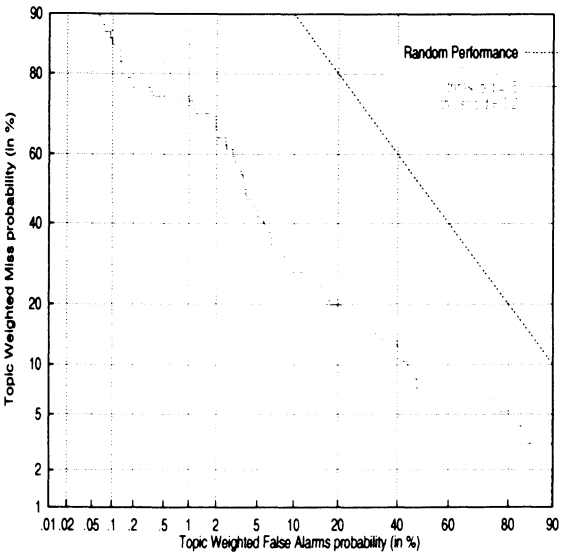


Figure 5.8. Threshold tuning for stories clustering in FSD

6. Story Link Detection

6.1 Experiment Design

This section describes our two story-link detection systems, and examines why their performance on the TDT-1999 evaluation data was considerably worse than on the TDT-1999 October dry-run data while performance on an alternate held-out evaluation set generated from the TDT-1999 evaluation data matched the performance on the October dry-run data.

Both CMU systems were based on cosine similarity and shared the same code library, although each tested different enhancements in the similarity computation and confidence measures. This common library is an outgrowth of the Dtree topic tracker from the TDT-1998 project [4, 20].

These systems were run on three distinct data sets. The first ("dry run") consisted of story pairs selected for the TDT-1999 October dry-run from the TDT-

1999 October dry-run corpus; the second ("evaluation") consisted of previously-unseen pairs selected for the TDT-1999 official evaluation from the corpus used for that evaluation; and the third ("alternate") consisted of additional pairs selected from the corpus used for the official TDT-1999 evaluation, which was provided in response to the dismal performance on the evaluation set of all submitted systems from all participants. Both the dry run and alternate sets selected their story pairs from among those stories which had received topic labels for the tracking task, while the evaluation set contained 120 candidate matches for each of 180 seed stories selected at random.

6.2 SLD Methods

The decision threshold for determining if two stories are linked is actually a split threshold, with different values depending on whether or not the two stories come from the same source; this permits a laxer threshold when the two stories are from sources which may have different writing styles, but may nevertheless report the same topic. Best performance was obtained for our first system (CMU-1) if the New York Times and AP newswire were treated as one source and all other TDT sources were combined into a second source.

CMU-1 uses incremental TF*IDF-weighted cosine similarity measure (COSINE in our results below) to determine whether or not two stories discuss the same topic, and that similarity is compared to a preset threshold to yield binary decisions.

For the evaluation, the TF*IDF values were initialized from the complete collection of English stories in the dry-run data set but were incrementally updated to adapt to changing patterns of use over time. The COSINE system can additionally apply a time-based decay to the similarity score, making temporally distant story pairs less likely to be declared linked, but this feature did not improve results.

The second system, identified as CMU-2 in the evaluation, is also based on weighted cosine similarity measures, using log(TF) as in the SMART document-retrieval system[12], and the TF*IDF statistics were derived solely from the test stories as they were processed, rather than having been initialized from the six-month training corpus.

6.3 SLD Results

Table 5.5 lists the results of four of the runs CMU submitted to the December 1999 evaluation, showing the normalized cost measure C_{link} for each run (smaller cost is better). While the CMU-2 system performed worse than it had on the training data, it far outperformed all CMU-1 runs on the test data. In fact CMU-2 significantly outperformed submissions from all other TDT sites as well on the official evaluation data. Why would two such similar methods

System	Transcription	Deferral	Norm(C_{link})
CMU-1	ASR	1	1.1260
CMU-1	ASR	10	1.0943
CMU-1	ASR	100	1.0921
CMU-2	ASR	10	0.4667

Table 5.5. Official Evaluation Results

yield results that are so different? Most of the variance is in the threshold tuning. CMU-2 was not tuned, contrary to CMU-1 and other submissions. Lack of tuning normally hurts performance, unless training and evaluation sets are markedly different – sampled from very different distributions. This indeed proved to be the case for story linking.

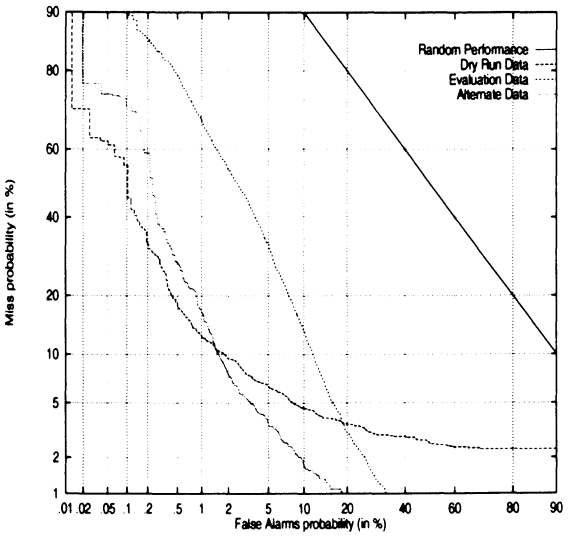


Figure 5.9. Performance Variation by Data Set: CMU-1 System

Figure 5.9 shows how errors and false alarms made by the CMU-1 system may be traded against one another by varying the threshold on the similarity measure for each of the three test sets. Figure 5.10 plots the equivalent Detection-Error Tradeoff (DET) curves for the CMU-2 system.

Another way to present the performance is with the F_1 measure which is commonly used in the broader information-retrieval community. F_1 is defined as $2pr/(p+r)$, where p is precision (proportion of retrieved stories which should have been retrieved) and r is recall (proportion of stories which should have been retrieved that actually were retrieved). When tuned for F_1 on the dry-run data set, CMU-1 currently achieves micro-averaged F_1 values of 0.92,

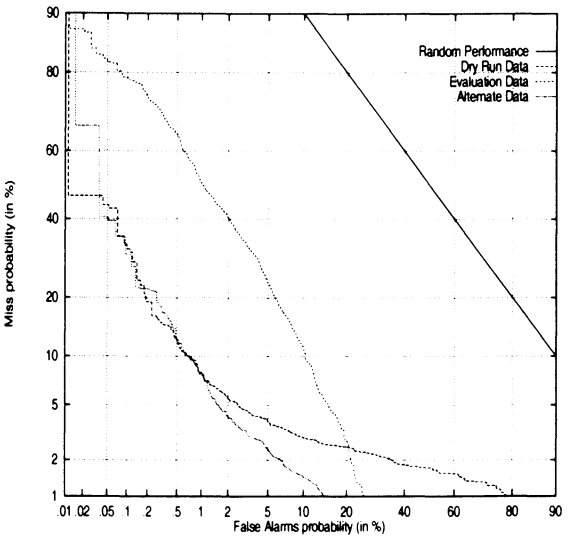


Figure 5.10. Performance Variation by Data Set: CMU-2 System

0.56, and 0.93 on the dry-run, evaluation, and alternate data sets – very good except on the evaluation set.

6.4 Confounding Evaluation Data

Cross-validation on the training data using the COSINE system led us to expect that the cost measure would be 20 to 30% higher on the evaluation data than on the training data, yet that system had a cost some six times as high and the CMU-2 system had a cost measure more than twice as high. Why did the two systems fare so much more poorly on the evaluation data than on the training data? The answer proved to be a difference in the method for generating the evaluation data that made it substantially different and "harder" than the training data.

The dry-run training data (as well as the alternate test set created after the December 1999 evaluation) was generated by using the topic-labeled stories in the English portion of the collection and associating 120 random other labeled stories with one story from each topic. Of the randomly-selected stories, those which had the same label as the initial story were considered linked, while those which did not were considered not linked.

In contrast, the evaluation data was not limited to the subset of the collection which had been labeled, and further was heavily biased toward confounding unlinked pairs. Half of the 120 stories associated with each of the 180 seed stories were selected at random, while the other 60 were selected from the

top of a ranked list of retrieval results. This latter half consists entirely of stories similar to the seed story, even though the subsequent manual labeling determined that the stories were not linked. Those pairs which are not linked are confounders which would occur in far lower numbers if not for the "confounder enrichment" of the evaluation data set. In other words, the evaluation set was drawn from a totally different statistical distribution than either the training set or the alternate evaluation set.

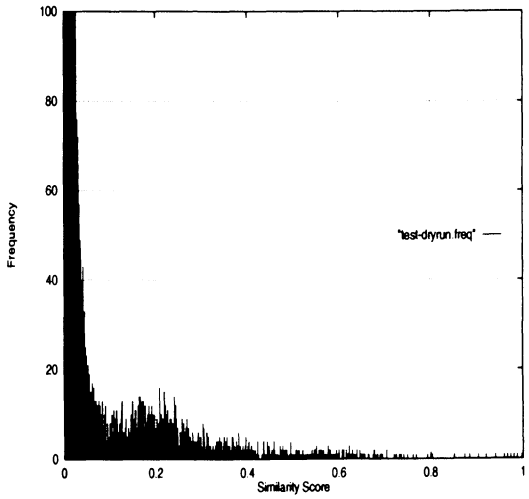


Figure 5.11. Comparing the Distributions of Similarity Scores: Dry-run data

Figures 5.11 to 5.13 compares the distribution of similarity scores computed by the CMU-1 system on each of the three data sets. The score for each story pair was placed into one of 1000 bins by truncating the score to 3 decimal places, and the number of elements in each bin was then plotted. The Y axis of each plot has been truncated somewhat to better illustrate the behavior in the region 0.1 to 0.2. As is clear from the figure, the distribution of scores are very similar between the dry run (figure 5.11) and alternate (figure 5.13) data, and quite different for the evaluation data (figure 5.12). Note the local minimum near 0.1, followed by a local maximum at 0.2 in the top and bottom graphs; this may be an indication of two well-separated Gaussian distributions for linked and non-linked story pairs. In contrast, this local minimum does not occur on the evaluation data (center), because the confounders introduced by using retrieval results to enrich the data set also fill in this local minimum. In terms of the optimum decision thresholds, the dry-run and alternate test sets have the lowest C_{link} at 0.065-0.075 for the CMU-1 system, while the evaluation set performs best at 0.18-0.22.

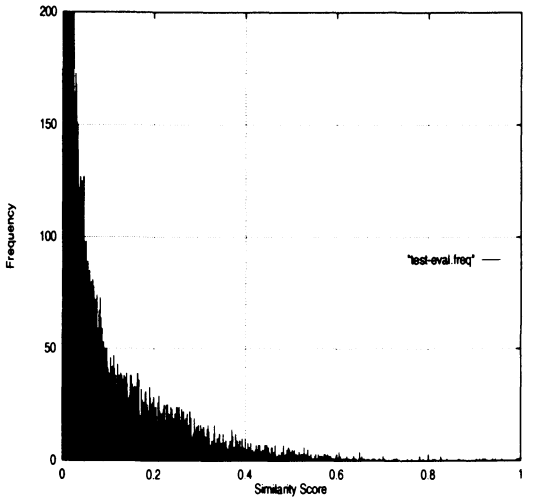


Figure 5.12. Comparing the Distributions of Similarity Scores: Official evaluation corpus

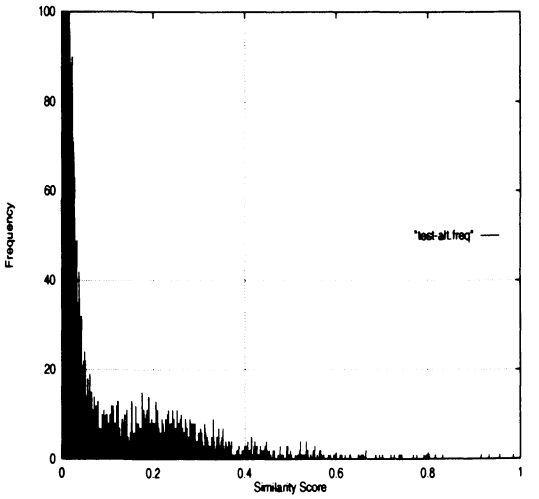


Figure 5.13. Comparing the Distributions of Similarity Scores: Alternate evaluation corpus

6.5 Discussion

Although both Carnegie Mellon SLD systems performed very well on the training and alternate evaluation data sets, performance on the official December 1999 "official" evaluation data was far worse. This demonstrates that tuning parameters to validation data for story-link detection is useful only if that validation data is sufficiently representative of the evaluation data. If sufficiently representative validation data is not available, systems should be tuned to an

operating point that is stable over a wide variety of corpora, even though this point may not be optimal for any of them. An alternative to the latter which was not explored for link-detection would be the use of a BORG-like multi-method system that proved far less sensitive to differences between training and evaluation conditions on the tracking task, discussed previously and in the literature [18, 19]. Of course the differences were far more pronounced – almost deliberately confounding – in the story linking task. The held-out alternate evaluation data confirms that story-linking systems performed far better when the training data distribution was predictive of the evaluation distribution.

7. Multilingual TDT

7.1 Multilingual Tracking in TDT-1999

A new aspect of TDT research is to investigate the feasibility of our technologies in dealing with multilingual stories. The TDT-1999 primary tracking evaluation was conducted under condition of using English stories for training and using a mixture of English and Mandarin stories for testing. The Mandarin test stories can either be machine-translated versions (using SYSTRAN) or original stories (Chinese-character based). Our primary tracking submission consisted of the result of kNN.avg2 which performed well, as shown in Table 5.6 and Figure 5.14. Our additional experiment with the same system but with the Mandarin stories not translated yielded much worse performance ($C_{trk} = 0.3971$), due to a bug in our current program for decoding the Mandarin characters.

Table 5.6. TDT-1999 Primary Tracking Results

Condition 1. Mandarin stories translated by SYSTRAN		
Site	Method	C_{trk}
BBN	language modeling	0.0922
CMU	kNN.avg2	0.1376
Dragon	language modeling	0.1596
GE		0.3778
Condition 2. Mandarin stories not translated		
Site	Method	C_{trk}
BBN	language modeling	0.1057
UPenn	Rocchio	0.2575
UIowa		0.6051
UMd		0.9662

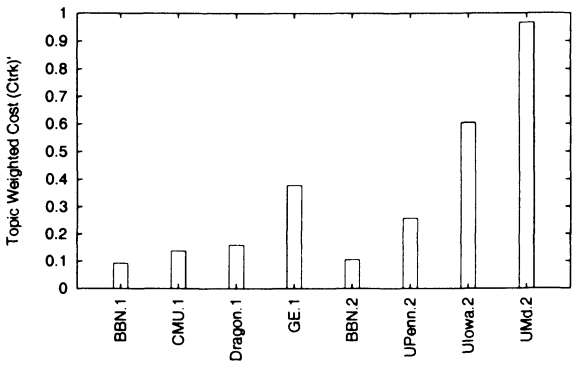


Figure 5.14. Results of TDT-1999 Primary Evaluation

7.2 A New Spanish Corpus

In order to further evaluate the multilingual capability of our TDT systems, we recently obtained a Spanish corpus from Linguistic Data Consortium, consisting of 27,046 news stories (42 MB) from Agency France Press during May and June of 1998. This collection would likely be included in future TDT data covering Spanish language sources; but at this point, it is a preliminary version, supplied without annotation.

Spanish-speaking graduate students were hired to annotate this Spanish corpus with TDT labels. We chose to use the TDT-1998 evaluation topics 67 through 100 because these topics were active during that time period in the English data[1]. As in TREC query-relevance labeling (and also the TDT-1998 and TDT-1999 topic labeling), we used a "pooling" strategy to obtain topic relevance judgments: the labellers used the LDC topic descriptions and IR systems to locate candidate on-topic stories, and a labelling tool which recorded both positive and negative judgements for viewed stories. Both labellers covered all the candidate topics, and an adjudication process is currently underway to address residual disagreements. We used partially-adjudicated labels in our experiments. Of the 34 candidate topics, 8 were not assigned to any stories. The topic with the largest number of stories was "India, A Nuclear Power? India begins nuclear testing." (topic 70), for which we found 402 on-topic stories. The topics for which we found no stories include, for example, "Food Stamps", "Human Rights Conference in Ethiopia." In all, 1,129 stories (4% of the total) received labels, and no story received more than one label.

[1] See http://morph.ldc.upenn.edu/tdt/newtopics/eval.html for full topic explanation.

7.3 Preliminary Results on Spanish

We tested our tracking and detection systems on the Spanish corpus, with only a 4-rule stemmer as Spanish-specific pre-processing. The preliminary results are shown in Figures 5.15-5.16 and Tables 5.7-5.8.

The preliminary results of our tracking systems on the Spanish corpus are summarized in Table 5.7; for reference we also listed the results of these systems on the TDT-1999 Dryrun Corpus which contains English and Mandarin news stories from January to June 1998. Note that those results are not directly comparable because the stories in the different languages are not translations of each other, they were from different news agencies, and they overlap only partially in time. The parameters in all systems were tuned on the TDT-1999 Dryrun Corpus. In the experiments with the Spanish data, we used the same set of parameters tuned for English and Mandarin; we have not yet fully explored the space of parameter optimization. Figures 5.15 shows the DET curves of our tracking systems on the Spanish corpus.

Table 5.7. Tracking results on multilingual corpora

System	opt-parameters on TDT-1999 Dryrun Corpus	$C^{\sharp}_{trk}$ on TDT-1999 Dryrun (Jan-Jun. Eng+Mandarin)	Results on May-Jun.Spanish			
			$C^{\sharp}_{trk}$	F_1	Recall	Prec
kNN.avg1	$k = 2000$	.0023	.0009	.53	.99	.40
kNN.avg2	$kp = 4, kn = 2000$	.0023	.0010	.60	.98	.48
Rocchio	$\gamma = -.25, n = 100$	.0026	.0010	.66	.98	.54
LM	$\lambda = 0.025$	.0040	.0020	.31	.96	.21

The preliminary detection results of our systems on the Spanish corpus are shown in Table 5.8 and Figure 5.16. We found LM had better performance than Incr.VSM, and the combined BORG.det system outperformed both individual method. Figure 5.16 and Table 5.8 compare the results of these three methods under the condition that parameters in each system were tuned on the TDT-1999 October dry run corpus (news stories in March to June in 1998). Surprisingly, all these systems had better performance on the Spanish data than with any official TDT set to date. It is not clear why it was so and further analysis is needed to determine the cause.

Over all, both the tracking and the detection results observed so far are surprisingly good, compared to our TDT-1999 results in English. More encouraging, perhaps, than those raw scores is the fact that they were achieved with same parameter values for the algorithms as used for English experiments. We have observed previously that these parameters generalize across data sets, and the Spanish experiment suggests that they may also generalize across languages. A limited exploration of the parameter space shows that the English parameters

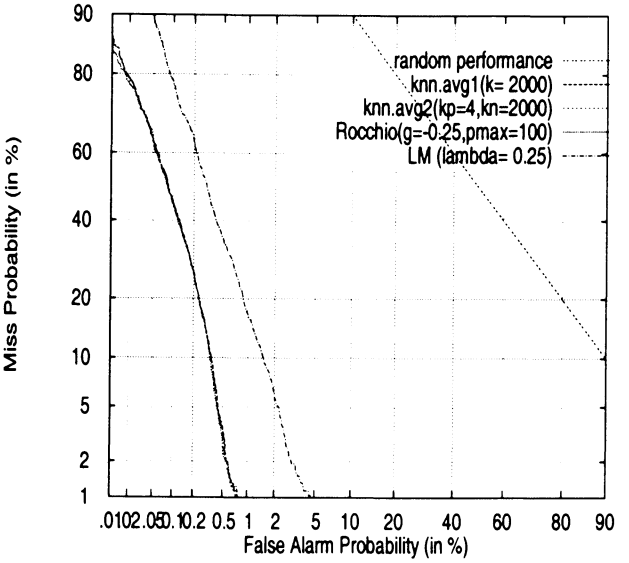


Figure 5.15. Tracking methods on the Spanish corpus

are at least near-optimal for the Spanish set. For example, by slightly perturbing a few parameters in the ballpark where our systems were optimized for the English data, we obtained the normalized $C_{det} = 0.1255$ (at thresholds of $cos = 0.02$ and $p = -7.0$) for BORG.det.

Table 5.8. Detection results on multilingual corpora

SYSTEM	opt-threshold on Mar-Apr.English	C_{det} on May-Jun.Spanish
Incr.VSM	cos = 0.03	.1348
Incr.LM	log p = -6.95	.1339
BORG.det = Incr.VSM+LM	cos = 0.1 log p = -6.9	.1293

To summarize our research in multilingual TDT, the topic is just now being explored. Our preliminary results with Spanish (and Mandarin) are indicative, but further analysis is required for thorough conclusions. Nevertheless, given the limited effort with which we applied and tuned our systems for Spanish, the results are quite impressive and encouraging, showing the potential of effectively solving TDT problems in different languages without costly manual coding of human knowledge or extensive linguistic analysis.

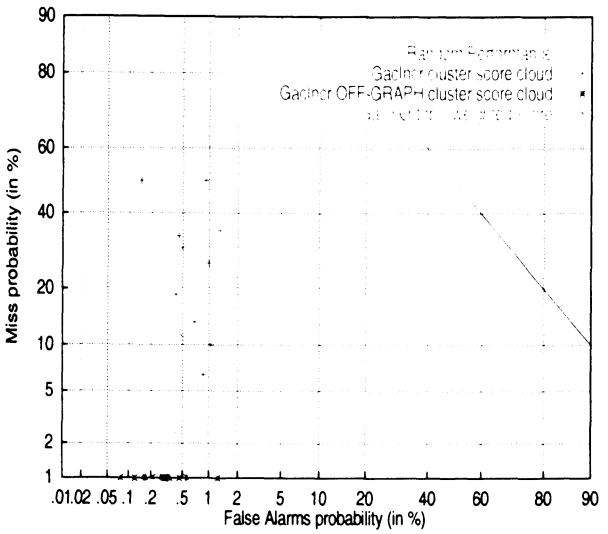


Figure 5.16. Detection results of Incr.VSM on Spanish

8. Concluding Remarks

Our contributions to the Topic Tracking and Detection initiative spanned all TDT tasks: segmentation, first-story detection, topic detection, topic tracking and story-link detection. In each case we investigated different technical approaches with varying degrees of success, and draw the following general conclusions:

- Machine Learning and Information Retrieval methods apply to all TDT tasks, performing quite well on tracking, relatively well topic and link detection and not as well on first-story detection.
- Maximum-entropy methods (exponential models) provide good performance in the segmentation task.
- Several methods provide good performance on the topic-tracking task, including new variants of kNN, Rocchio, and language modeling.
- Multi-method combination (our BORG approach) for *tracking* worked very well at reducing result variance when tuning and evaluation sets differed significantly, and improved overall performance over the best single individual methods.
- Multi-method combination has not yet shown similar improvements for topic detection.

- Radical differences in the distributions from which training and evaluation data are sampled had a significant adverse performance for story-link detection, especially for systems with tunable parameters on training data. In contrast, sampling from similar distributions for training and for held-out evaluations (the alternate evaluation set in story linking) yielded far superior results.
- The CMU approaches to the TDT tasks work well for new languages with minimal or no adaptation. This result is borne out by better-than-expected Spanish tracking and detection results utilizing methods not tuned or adapted for Spanish.

Acknowledgement

The work reported in this chapter was funded in part by the National Science Foundation (award number IIS-9873009), and in part by the US Department of Defense.

References

[1] Allan, J., J. Carbonell, G. Doddington, J. Yamron, and Y. Yang: 1998, 'Topic Detection and Tracking Pilot Study: Final Report'. In: *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*. San Francisco, CA, pp. 194–218, Morgan Kaufmann Publishers, Inc.

[2] Bartell, B. T., G. W. Cottrell, and R. K. Belew: 1994, 'Automatic Combination of Multiple Ranked Retrieval Systems'. In: *Proceedings of the Seventeenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, pp. 173–181, The Association for Computing Machinery.

[3] Beeferman, D., A. Berger, and J. Lafferty: 1999, 'Statistical Models for Text Segmentation'. In: *Machine Learning*, Vol. 34. pp. 1–34.

[4] Carbonell, J., Y. Yang, J. Lafferty, R. D.Brown, T. Pierce, and X. Liu: 1999, 'CMU report on TDT-2: Segmentation, Detection and Tracking'. In: *Proceedings of the DARPA Broadcast News Workshop*. San Francisco, CA, pp. 117–120, Morgan Kaufmann Publishers, Inc.

[5] Fiscus, J.: 1997, 'A post-processin system to yield reduced word error rates: Recognizer Output Voting Error Reduction (ROVER)'. In: *IEEE Workshop on Automatic Speech Recognition and Understanding*.

[6] Freitag, D.: 1998, 'Multistrategy Learning for Information Extraction'. In: *Proceedings of the Fifteenth International Conference on Machine Learning*. San Francisco, pp. 161–169, Morgan Kaufmann.

[7] Jin, H., R. Schwartz, S. Sista, and F. Walls: 1999, 'Topic Tracking for Radio, TV Broadcast and Newswire'. In: *Proceedings of the DARPA Broadcast News Workshop*. San Francisco, CA, pp. 199–204, Morgan Kaufmann Publishers, Inc.

[8] Jr., J. J. R.: 1971, 'Relevance feedback in information retrieval'. In: G. Salton (ed.): *The SMART Retrieval System: Experiments in Automatic Document Retrieval*. Englewood Cliffs, New Jersay, pp. 313–323, Prentice-Hall, Inc.

[9] Katzer, J., M. MacGill, J. Tessier, W. Frankes, and P. Dasupta: 1982, 'A study of the overlap among document representations'. In: *Information Technology: Research and Development*, Vol. 1. pp. 261–274.

[10] Larkey, L. S. and W. B. Croft: 1998, 'Combining Classifiers in Text Categorization'. In: *Proceedings of the Nineteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, pp. 289–297, The Association for Computing Machinery.

[11] Lee, J. H.: 1995, 'Combining Multiple Evidence from Different Properties of Weighting Schemes'. In: *Proceedings of the Eighteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, pp. 180–188, The Association for Computing Machinery.

[12] Salton, G.: 1989, *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Reading, Pennsylvania: Addison-Wesley.

[13] Salton, G. and C. Buckley: 1990, 'Improving retrieval performance by relevance feedback'. *Journal of American Society for Information Sciences* **41**, 288–297.

[14] Schapire, R. E., Y. Singer, and A. Singhal: 1998, 'Boosting and Rocchio Applied to Text Filtering'. In: *21th Ann Int ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'98)*. pp. 215–223.

[15] Schwartz, R., T. Imai, L. Nguyen, and J. Makhoul: 1997, 'A Maximum Likelihood Model for Topic Classification of Broadcast News'. In: *Proceedings of Eurospeech*. Rhodes, Greece.

[16] Walls, F., H. Jin, S. Sista, and R. Schwartz: 1999, 'Topic Detection in Broadcast News'. In: *Proceedings of the DARPA Broadcast News Workshop*. San Francisco, CA, pp. 193–198, Morgan Kaufmann Publishers, Inc.

[17] Yamron, J., I. Carp, L. Gillick, S. Lowe, and P. van Mulregt: 1999, 'Topic Tracking in a News Stream'. In: *Proceedings of the DARPA Broadcast News Workshop*. San Francisco, CA, pp. 133–136, Morgan Kaufmann Publishers, Inc.

[18] Yang, Y., T. Ault, and T. Pierce: 2000a, 'Combining multiple learning strategies for effective cross-validation'. In: *Proceedings of the 17th International Conference on Machine Learning (ICML00)*. San Francisco, pp. 1167–1182, Morgan Kaufmann.

[19] Yang, Y., T. Ault, T. Pierce, and C. Lattimer: 2000b, 'Improving text categorization methods for event tracking'. In: *Proceedings of the Twenty-Third Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'00)*. New York, pp. 65–72, The Association for Computing Machinery.

[20] Yang, Y., J. Carbonell, R. Brown, T. Pierce, B. T. Archibald, and X. Liu: 1999, 'Learning Approaches for Detecting and Tracking News Events'. *IEEE Intelligent Systems, Special Issue on Applications of Intelligent Information Retrieval* **14**(4), 32–43.

[21] Yang, Y., T. Pierce, and J. Carbonell: 1998, 'A study on retrospective and on-line event detection'. In: *Proceedings of the 21th Ann Int ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'98)*. pp. 28–36.

Chapter 6

Statistical Models of Topical Content

J.P. Yamron, L. Gillick, P. van Mulbregt
formerly of Dragon Systems/Lernout & Hauspie
320 Nevada Street
Newton, MA 02460

S. Knecht
Dragon Systems/Lernout & Hauspie
320 Nevada Street
Newton, MA 02460

Abstract In this chapter we explore the behavior of two different statistical models, one based on simple unigrams and another based on the beta-binomial distribution, as applied to the problem of modeling story generation. We describe how these models can be incorporated into information extraction applications, particularly Tracking and Detection engines built for the Topic Detection and Tracking evaluations sponsored by DARPA. Tracking systems based on the two models have complementary strengths and weaknesses: a Beta-Binomial system yields high precision at high decision threshold, but performance quickly degrades as the threshold drops; a Unigram system is not as strong at high decision threshold, but is very good at suppressing false-alarms at lower threshold. We will describe the features of these systems that give rise to this behavior, and discuss ways that each system might be improved by borrowing from the other. We will also discuss our Detection system, and how improvements in Tracking should lead to improvements in Detection.

1. Introduction

The purpose of this research [1, 2, 3, 4, 5, 6, 7, 8, 9] is to explore statistical models of the topical content of news stories. Our goal is to develop a basic framework in which a variety of information extraction tasks can be investigated.

Our focus is on data-driven techniques, which facilitate the updating and porting of models to new domains and languages.

The approach we take here is to model story *generation*. We imagine that there is a probability distribution P^T associated with a particular topic T, from which a story s (from the universe of all possible stories) may be generated with probability $P^T(s)$. $P^T(s)$ is a direct measure of the likelihood that the story s is on topic T. For example, if the topic T is *sports*, then we would expect $P^T(s)$ to be large if s were a story on the Super Bowl, but we would expect it to be small if s were on Washington politics. In a retrieval (or similar) application, we could use $P^T(s)$ to help determine whether story s should be returned to a user who has indicated an interest in stories on topic T.

When the distribution P^T is constructed from a collection of stories on topic T, the story probability $P^T(s)$ can be thought of as a measure of the distance between the collection and s. Similarly, when the distribution P^T is constructed from a single story, $P^T(s)$ is a measure of the distance between that story and s. Measures of this type can be used to build applications for a variety of tasks.

For example, consider the Tracking and Detection tasks from the Topic Detection and Tracking (TDT) program sponsored by DARPA. In the Tracking task, a system is given a small number of news stories on a particular event and asked to retrieve similar stories from an incoming news stream. In the Detection task, which is essentially a problem in real-time clustering, a system must bootstrap a set of event clusters from an incoming news stream, which means either assigning each story as it comes in to an existing cluster or seeding a new cluster. For both applications, the key element is a comparison between an incoming story and a story collection (the training stories in the case of Tracking, the stories making up a particular cluster in the case of Detection). The same statistical model can be used in both cases.

Implementing multiple applications from a common statistical model has the advantageous feature that advances in our modeling techniques can be leveraged to yield performance improvements in those multiple applications. Furthermore, as long as the training procedure for these models is data-driven, the porting of an application to new domains or languages is as straightforward as supplying new training data.

In this chapter we will look at two different statistical models of story generation, one based on unigram statistics and another based on the beta-binomial distribution, and will examine their behavior in Tracking and Detection systems submitted for the TDT evaluations. In this investigation, it is our intention to keep the models simple in order to facilitate comparisons between the models. For this reason, we will not be applying some techniques that may be familiar from other information extraction engines, such as morphological analysis, use of bigrams or collocations, or shallow parsing. It is important to note that there

is nothing about our modeling that prevents the incorporation of such features, and indeed we are considering them for future versions of our models.

In Section 2 we present the two models, Unigram and Beta-Binomial, that we will be using in this study. Section 3 describes three Trackers based on these models, and Section 4 describes our Detection engine. Section 5 contains a summary and closing comments.

This research was performed primarily at Dragon Systems. The authors would like to acknowledge the support of DARPA, under the TDT and TIDES (Translingual Information Detection, Extraction, and Summarization) programs.

2. Models of Story Generation

We explore the behavior of two different statistical models of document generation, one based on *unigram statistics* and another based on the *beta-binomial distribution* [7, 8].

2.1 Unigram Model

As a model of story generation from a topic T, the unigram model works as follows: we associate a unigram distribution $p^T(w)$ to the topic T, which gives the probability of generating a single word w in a test story s. The probability $P^T(s)$ that an entire test story s was generated by $p^T(w)$ is just the product of the probabilities of each word being drawn from this distribution. If we define V_s to be the set of *distinct* words in the story s, then $P^T(s)$ can be written

$$P^T(s) = \prod_{w \in V_s} \left[p^T(w)\right]^{n_w} ,$$

where n_w is the number of occurrences of the word w in s. In our implementation, we incorporate a stop list of common words.

In many applications, it is convenient to train $p^T(w)$ from examples of text on the topic T—a particularly sensible approach is to assume that the probability $p^T(w)$ is proportional to the number of times w occurs in the training data (the maximum likelihood estimate, or MLE). A problem which often arises, however, is that the amount of text on which to train is very small (in the Tracking task, for example, the training consists of four stories, typically comprising fewer than 1000 words). The "raw" MLE distribution generated by counting occurrences in the topic training data assigns zero probability to many words that may occur in a test story, and therefore requires *smoothing*. Typically, smoothing involves combining the sparse distribution derived from the training data with a *background* distribution built from other available material. Ideally, this available material is as similar to the training data as possible.

To this end, our approach to unigram smoothing makes use of a technique called *targeting*, [6, 9] in which we take a large number of background unigram

distributions, find the mixture of these that best approximates the sparse topic model, and use this mixture as a smoothing distribution.

More concretely, given a sparse topic unigram model $\tilde{p}^T(w)$ built from the topic training data, and a set of *background models* $p_i^B(w)$, we find the best mixture

$$p^B(w) = \sum_i \lambda_i \; p_i^B(w) \; , \; \sum_i \lambda_i \; = 1 \; ,$$

such that the Kullback-Leibler distance between $\tilde{p}^T(w)$ and $p^B(w)$,

$$d = \sum_w \tilde{p}^T(w) \log \frac{\tilde{p}^T(w)}{p^B(w)} \; ,$$

is minimized. This leads to an implicit equation for the λ_i :

$$\lambda_i \; = \sum \frac{\tilde{p}^T(w) \lambda_i \; p_i^B(w)}{\sum_j \lambda_j \; p_j^B(w)} \; ,$$

which is easily solved by iteration. The distribution $p^B(w)$ is used to smooth the sparse topic model $\tilde{p}^T(w)$ to produce $p^T(w)$ (for more details see [6]).

In previous work [1, 3, 4], the targeting of the unsmoothed topic data was done against about 100 unigram distributions built by automatically clustering a large number of background news stories into broad topics. The result of the targeting step was the mixture of these broad topic distributions that best approximated the specific topic defined by the training data. Because a *topic* in TDT is defined very narrowly (corresponding more to the notion of an *event*), we have switched to targeting against the individual background stories, typically numbering in the tens of thousands. This provides a mixture that is much more focused on the topic training.

For this investigation, we have concentrated on some of the details of how to smooth the sparse topic model with the targeting mixture. More will be said about this below.

2.2 Beta-Binomial Model

In the Beta-Binomial model [7, 8] the story generation process is more complicated than in the unigram case. Instead of associating a single unigram distribution with a topic T, one instead associates a *distribution* of unigram distributions. Generating a story s consists of a two step process:

- Drawing a distribution $p^T(w)$ from the space of possible unigram distributions for the topic T;
- Drawing the words of s from $p^T(w)$.

This model incorporates the idea that there may be variations across a story collection defining a topic T that are lost if that collection is represented only by the unigram statistics of the combined data.

As described in [7, 8], using beta-binomial statistics one can derive a distribution $p^T(n_w|N, \mu_w^T, \nu_w^T)$ which gives the probability of seeing n_w occurrences of word w in a story of size N. The two parameters μ_w^T and ν_w^T are topic-dependent and are derived from the word occurrence statistics of the story collection on which topic T is based. The parameter μ_w^T is the expected value of the word probability in the collection, and ν_w^T is related to the *variance* in the word probability across the collection. Given a test story s of size N, and the set V_s of distinct words in s, the probability that s is generated by the distribution $p^T(n_w|N, \mu_w^T, \nu_w^T)$ is given by:

$$P^T(s) = \prod_{w \in V_s} p^T(n_w|N, \mu_w^T, \nu_w^T) \ .$$

(Strictly speaking, the distributions $p^T(n_w|N, \mu_w^T, \nu_w^T)$ corresponding to different words w in s are not independent, because the counts n_w must sum to N. This means that $P^T(s)$, as defined, is not properly normalized. A more careful treatment that properly normalizes $P^T(s)$ does not yield significantly different results, so we ignore this complication here.)

The introduction of the variance of the word probability as a parameter is a distinguishing characteristic of the beta-binomial model. It permits a more accurate modeling of the word distribution across a story collection than is possible with unigram statistics alone. One interesting fact that the model quickly discovers is that the variance in function word occurrence across a story collection is typically larger than is expected from a unigram model, and that variability in the appearance of these words is therefore not a signal of changing topic—a fact which allows the Beta-Binomial system to run without a stop list [7, 8]. Note that, although we chose in this system to model only word occurrences, the model is capable of being extended to include other features, such as bigram occurrences or collocations.

In order to derive a beta-binomial distribution from a set of training stories that defines the topic, we must once again face the problem that the topic training data might be extremely sparse. We resort to the following approximations in the computation of the model parameters:

- To reduce noise in the model score, we restrict the computation to the contribution of a small number of keywords K, defined as those words whose probability on the topic training set compared to some background exceeds a threshold.
- For an infrequent keyword, its variance is set to a fixed function of the keyword's mean probability.

One consequence of this is that the contributions to the Beta-Binomial tracking score $P^T(s)$ for a test story s come only from the keywords K selected from the topic training data.

3. Tracking Systems

We describe the behavior of two different trackers, one based on unigram statistics and another based on the beta-binomial distribution [1, 4, 6, 7, 9]. The job of a tracker is to search for instances of a topic T, defined by story examples, in an incoming stream of test stories. It does this by assigning a likelihood score to all stories, which can be thresholded to yield a hard decision on whether a story in on- or off-topic. A high threshold selects very few stories, and therefore tends to miss (falsely label as off-topic) many on-topic stories, while triggering very few false-alarms. A low threshold, on the other hand, triggers more false-alarms, but misses fewer on-topic stories.

Trackers built around the unigram and beta-binomial models were originally submitted for the TDT 1998 evaluation. Their performance on that evaluation, as measured by the percentage of on-topic stories missed *vs.* off-topic stories falsely returned, is shown in Figure 6.1.

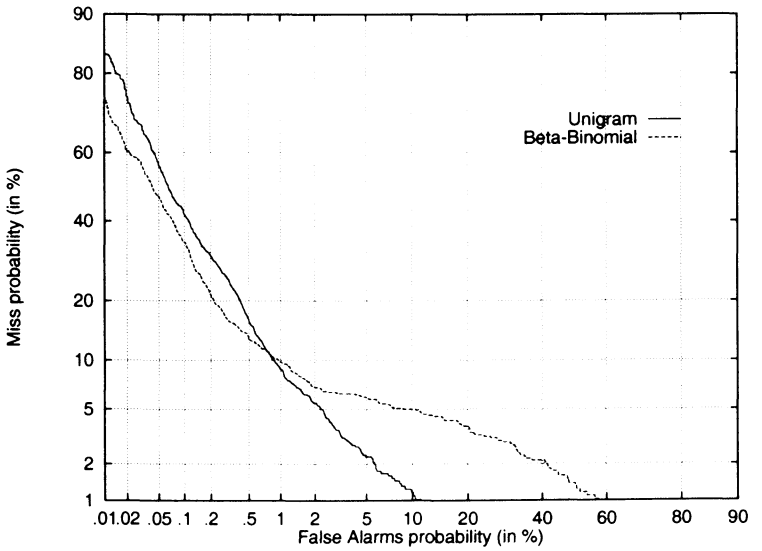


Figure 6.1. Tracker Comparison, TDT 1998 Evaluation

(Points on these curves correspond to choices in the decision threshold for each tracker.) This graph shows the strikingly different behavior of the models at the threshold extremes. The Unigram Tracker performed best at low decision threshold, while the Beta-Binomial Tracker was best at high threshold. Said

another way, the Unigram Tracker was found to be very good at separating poorly scoring on-topic stories from the bulk of the off-topic material, while the Beta-Binomial Tracker was good at distinguishing high-scoring off-topic stories from on-topic material.

For the TDT 1999 evaluation, we chose to exploit the complementary behavior of the two trackers by submitting an interpolated system.

3.1 Unigram Tracker

Our Unigram Tracker consists of two unigram models of the type described above: a *topic model* P^T (with an associated word distribution $p^T(w)$) built from the supplied topic training stories, and a *discriminator model* P^X (with an associated word distribution $p^X(w)$) built from available background material. The *tracking score* S_U of a test story s is defined as the log ratio of its score in these two models—in other words, the log ratio of the probability $P^T(s)$ that the topic model generated s (distance of s from the topic T) *vs.* the probability $P^X(s)$ that s was generated by background (distance of s from background).

A virtue of using unigram distributions to model document generation is that the tracking score S_U for a test story s consisting of N words is very simple to compute from the unigram distributions $p^T(w)$ and $p^X(w)$ that characterize the topic and discriminator:

$$S_U = \log \frac{P^T(s)}{P^X(s)} = \sum_{w \in V_s} n_w \left[\log p^T(w) - \log p^X(w) \right] ,$$

where the sum is over all distinct words V_s in the story s, and n_w is the number of occurrences of the word w in s.

Much of the work on improving the performance of the Unigram Tracker for TDT 1999 centered on the algorithm used to smooth the topic model with the targeting mixture described in the previous section. Our experience in language modeling for speech recognition had led us to the use of a discount-backoff method [10, 11] for smoothing in this application. For the case of very sparse distributions, however, this method is not sufficiently aggressive, in the sense that there is a limit to how many counts will be redistributed to unseen words (standard discounting methods do not take from any count more than the smallest count that appears in the entire unsmoothed distribution). The effect of this is that, by doing discount-backoff smoothing of the topic model to the targeting mixture, the topic model ends up being too narrow. This means that an on-topic test story s scores a probability $P^T(s)$ that is very small unless it happens to match extremely well with the small number of stories making up the training data.

To counter this behavior, we switched to interpolation in the smoothing step, which allows us to put as much or as little weight as desired on the unsmoothed

distribution by tuning the interpolation weight. This seemingly small change resulted in a significant gain in performance, with the interpolation weight on the topic data tuned to a surprisingly low 10–20%.

Another change made for TDT 1999 was in the discriminator model. In previous systems [1, 4, 6], the discriminator used in the Unigram Tracker actually consisted of multiple unigram distributions (typically 100), and computing the tracking score involved choosing the best scoring of these models to compare to the topic score. The measured improvement over using a single discriminator was sharply reduced when the smoothing enhancements described above were introduced, and is apparently confined to the region of high decision threshold. The improvement is not enough to challenge the performance of the Beta-Binomial system in that region, however, and since we were mainly concerned with the performance of a combined system, and because the use of multiple discriminators drastically increases processing time, we chose to use only a single discriminator.

With a single discriminator, contamination by on-topic material becomes a critical problem. To eliminate potential on-topic stories from the training material used to build the discriminator (the 82710 stories comprising the TDT-2 development corpus), we ran a simple tracker over this material and removed stories that scored higher than a specified threshold. The discriminator for the evaluation run was then built from the unigram distribution of the remaining stories.

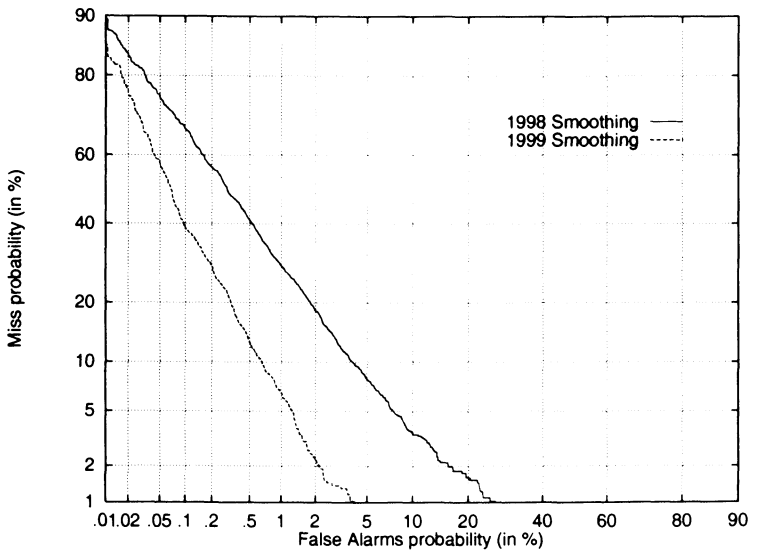


Figure 6.2. Smoothing Comparison, 1998 vs. 1999.

The effect of the improvements due to improved smoothing and to filtering of on-topic stories from the discriminator is shown in Figure 6.2, which compares single discriminator systems using the TDT 1998 (discount smoothing) and TDT 1999 (interpolated smoothing, filtered background) model building techniques, on the TDT 1998 evaluation data. A significant improvement in performance was achieved.

3.2 Beta-Binomial Tracker

Like the Unigram Tracker, the Beta-Binomial Tracker consists of a topic model P^T built from the supplied topic training stories, and a discriminator model P^X built from background material. The *tracking score* S_{BB} of a test story s is, again, the log ratio of its score in these two models

$$S_{BB} = \sum_{w \in K} \left[\log P^T(n_w | N, \mu_w^T, \nu_w^T) - \log P^X(n_w | N, \mu_w^X, \nu_w^X) \right] ,$$

where μ_w^T and ν_w^T are computed from the topic training stories, μ_w^X and ν_w^X are computed from the background material, and the values $P^T(n_w|N, \mu_w^T, \nu_w^T)$ and $P^X(n_w|N, \mu_w^X, \nu_w^X)$ are the probability of word w occurring n_w times in story s as computed in the topic model and the discriminator model, respectively. Because the topic training data is not smoothed in any way, the sum is restricted to a set of keywords K whose probability on the topic training set compared to the discriminator exceeds a tuned threshold.

High Decision Threshold. At high decision threshold (when only a small number of high-scoring documents are declared on-topic), the Beta-Binomial system outperforms the Unigram system significantly. This indicates that the beta-binomial distribution is better able to model the topic training stories, and to more accurately and reliably identify test documents that closely resemble this data. One difference between the systems that one might think could account for some of this performance gap is in the length of the word list used in the scoring computation: for the Beta-Binomial system, this list was quite restricted (typically 5-30 words), while for the Unigram system it consisted of the entire 60,000 word vocabulary—these extra words, most of them irrelevant to the topic, could be introducing noise into the unigram scoring, resulting in false-alarms. However, experiments on restricting the word list used by the Unigram system to a small number of keywords indicate that very little, if any, of the performance degradation relative to the Beta-Binomial system can be explained by the difference in the word lists.

This leaves as the most likely explanation for the performance difference the fact the beta-binomial model has an extra parameter, the variance, with which to model the behavior of each word. Interestingly, there was rarely enough data in the topic training stories to train the keyword variances, forcing the system to approximate them. It is possible, therefore, that the performance

advantage that the Beta-Binomial system holds over the Unigram system at high decision threshold comes not from modeling word occurrences via a beta-binomial distribution, but simply from modeling them via a distribution with a variance different from that imposed implicitly by the unigram distribution.

Low Decision Threshold. The more striking contrast between the behavior of the two systems occurs at low decision threshold, where the Unigram system performs well and the Beta-Binomial system exhibits a long false-alarm tail. To our surprise, this tail proved to be even more pronounced on the TDT-2 development corpus than on the TDT 1998 evaluation data. An exploration of this phenomenon revealed that the difference in performance between the data sets could be explained by the presence in the TDT-2 corpus of more large topics (each with several hundred on-topic stories), on which the Beta-Binomial system did particularly poorly. Figure 6.3 shows the behavior of the Beta-Binomial system on large and small topics, on the TDT-2 development corpus.

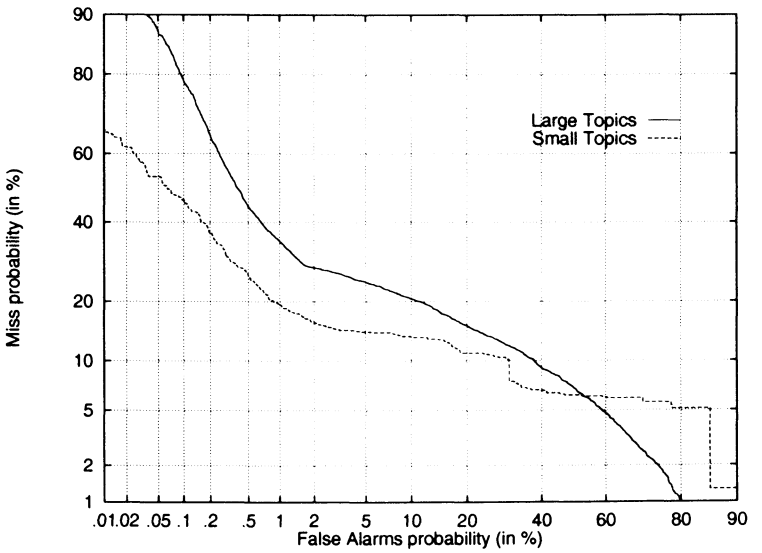


Figure 6.3. Beta-Binomial Tracker, Large vs. Small Topics

These two characteristics of the Beta-Binomial system, a long false-alarm tail at low decision threshold and particularly degraded performance on large topics, suggest that the system is unable to generalize well from the topic training material to capture on-topic stories that are related to, but do not closely resemble, that material. The Unigram system, on the other hand, generalizes quite well.

It might be guessed that the ability of the Unigram system to generalize comes from its use of the full 60,000 word vocabulary. This is partly right—the larger

set of words makes it possible to meaningfully score documents that share very few words with the original topic training stories. What makes this larger list effective in the Unigram system, however, is the aggressive smoothing of the training material, including the addition of the extra material generated through targeting. The Beta-Binomial system does not smooth the topic training stories or make use of any additional data to train the topic model. It is not surprising, therefore, that experiments testing the effect of simply increasing the number of keywords used in the beta-binomial computation showed little improvement on the false-alarm tail. If we were to supplement the data used to build the topic model, however, it should be possible to build a beta-binomial topic model that generalizes as well as the unigram model.

In certain cases, a more effective approach to improving the low threshold performance of the Beta-Binomial system was the introduction of adaptation on high-scoring test stories. Each such story was added to the list of topic training stories, the topic model was rebuilt, and tracking continued from that point with the new model. As Figure 6.4 shows, for some large topics adaptation was extremely effective at reducing the false-alarm tail, although for most there was usually a price to be paid in the high-threshold region. Unfortunately, adaptation resulted in degraded performance on small topics, probably because the small number of on-topic test stories for such topics truly are narrowly focused, and attempts to adapt invariably lead to contaminating the topic model with off-topic material. Contamination is also probably the cause for the loss of performance in the high-threshold region on large topics.

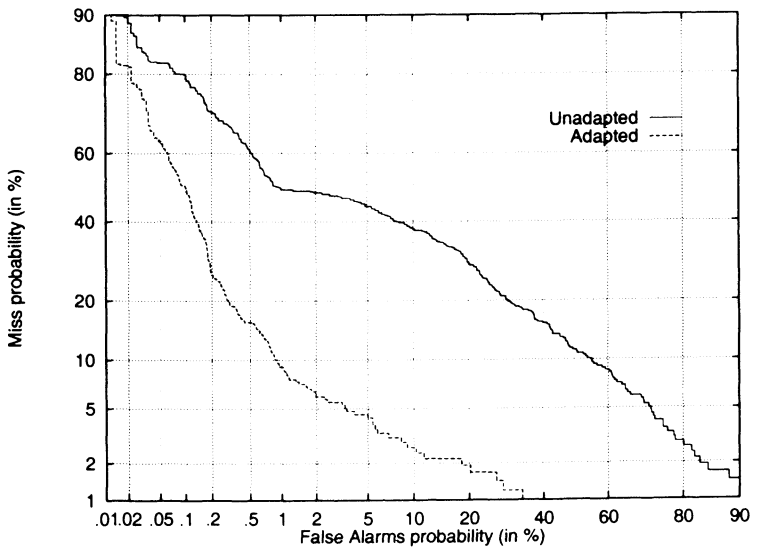


Figure 6.4. Beta-Binomial Tracker, Adaptation on Topic 20002

Because of the harm done to small topics, and because the effect of adaptation was usually detrimental to the performance at high threshold (where we were tuning the Beta-Binomial Tracker for best performance), we did not use it in the Beta-Binomial Tracker for TDT 1999 evaluation. We suspect that the loss of performance on the small topics can be avoided by a more judicious use of adaptation than we tried here; for example, adaptation could be deferred until the system could make a determination of the likely size of the topic (perhaps by recording the frequency of high scoring test stories), then dropped if the topic was found to be too small.

English vs. Mandarin. For the TDT 1999 evaluation, we ran our systems on the SYSTRAN-translated Mandarin data. Figure 6.5 shows the gap in performance between English and Mandarin for the Beta-Binomial Tracker on the TDT development data (there is a similar gap in performance for the Unigram Tracker). We explored several ideas in an effort to understand this difference.

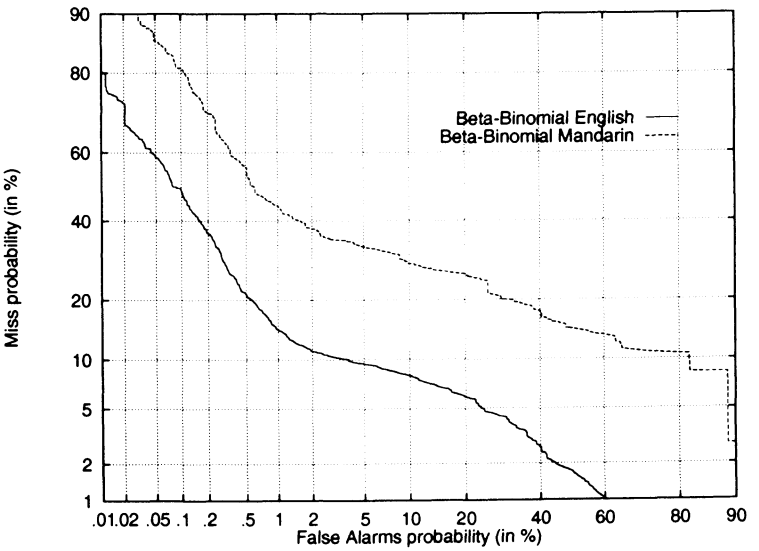


Figure 6.5. Beta-Binomial Tracker, English vs. Mandarin

One hypothesis is that the Mandarin test stories suffer from being compared to a background model which is trained predominately (about 75%) from English stories. (The mismatch with the topic model, which is trained exclusively from English stories, is a test condition and therefore unavoidable.) To test this idea, we built a background model from Mandarin stories only and compared the performance of the Mandarin test stories when using this *vs.* the standard background. We found virtually no difference in performance.

Another worry is that if the scores of Mandarin stories are not commensurate with the scores of English stories, then simply pooling the scores for the purpose

of applying a threshold (which is what we do for the evaluation) might not be optimal—we might want to shift or scale the Mandarin scores before pooling.

An analysis of scores shows that, on off-topic test stories, the average scores and deviations are similar between English and Mandarin, suggesting the shifting or scaling the Mandarin is not appropriate. (The average scores on on-topic test stories are quite different, but this is presumably because the on-topic Mandarin test stories are not well modeled by the English topic training stories.)

As a further check, we scored the English and Mandarin separately, found the decision threshold corresponding to the optimal value of the evaluation metric C_{track}, and compared the English and Mandarin scores at this threshold. They were comparable, another indication that shifting the Mandarin scores would not improve the overall performance of the tracker. We continue to research this difference in performance between the two languages.

3.3 Interpolated Tracker

Given a system optimized for performance at low decision threshold (Unigram), and a system optimized for performance at high threshold (Beta-Binomial), it should be possible to construct a system with the advantages of both.

Our first attempt was a simple linear interpolation of the Unigram and Beta-Binomial scores. The result on the TDT 1998 evaluation data was very encouraging, yielding better performance than both components at virtually all decision thresholds. On the TDT-2 development corpus, however, linear interpolation does not achieve the performance of the Unigram model at low threshold (although it does as well as the Beta-Binomial model at high decision threshold).

Because the trackers achieve their best performance in different decision regions, and these regions correspond to different output score values (high decision thresholds correspond to high scores, and low thresholds to low scores), we devised a score-dependent interpolation that explicitly favored the better model in the appropriate region. The form of this interpolation is:

$$S_I = (S_{BB} - b)\exp\big[m(S_{BB} - b)\big] + (S_U - b)\exp\big[-m(S_U - b)\big] \ ,$$

where S_{BB} and S_U are the scores assigned by the Beta-Binomial and Unigram Trackers, respectively, S_I is the score assigned by the Interpolation Tracker, and m and b are tunable parameters. The interpretation of this formula is as follows. The parameter b represents the *crossover* score, above which the Beta-Binomial Tracker is favored and below which the Unigram Tracker is favored. Above this point the score from the Beta-Binomial Tracker is scaled up exponentially, causing it to dominate, while the score from the Unigram Tracker is scaled down. Below the crossover the roles of the trackers are reversed. The parameter m controls the scaling rate.

The parameters m and b were tuned on both the TDT 1998 evaluation data and the TDT-2 development corpus. The result of interpolating the models by this method is shown in Figure 6.6, on the TDT-2 development corpus.

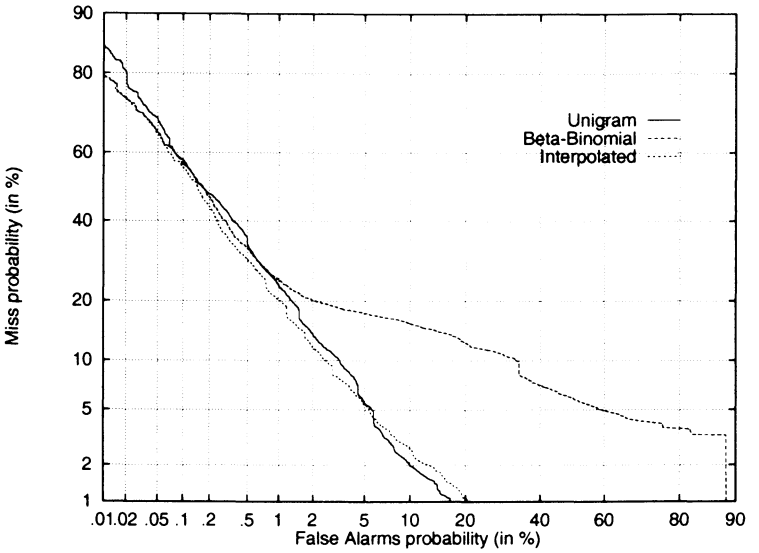


Figure 6.6. Interpolation, Score-Dependent Scaling

The Interpolated curve achieves the best performance of both components. This system was submitted as our primary tracker for the TDT 1999 evaluation.

4. Detection System

The Detection task is a problem in real-time (or near real-time) clustering. The goal is to "discover" events in an incoming stream by grouping similar stories. In TDT, Detection systems are allowed a limited look-ahead (the *deferral*) before committing an incoming story to an existing cluster or seeding a new cluster.

Our Detection system [2, 7] is based on a simple k-means clustering engine, and operates as follows:

- At the moment that an incoming story is received, assume that there are k story clusters, each cluster characterized by a set of statistics. The distance between each of these clusters and the story is computed.
- The story is inserted into the closest cluster, unless its distance to this cluster exceeds a threshold, in which case a new cluster is created.
- Before accepting another incoming story, the cluster assignments of all uncommitted stories (stories for which the incoming story is within their

deferral window) are reevaluated. Stories may move, seed new clusters, or remain in their assigned clusters. This step is repeated as often as desired.

This algorithm relies on the existence of a collection–story distance measure, with the "collection" in this case corresponding to a cluster. The measure used in the TDT 1998 Detection evaluation was constructed as follows:

- The counts for all stories in a cluster were combined to create a single cluster unigram distribution.
- The cluster distribution was smoothed via discount-backoff [10, 11] with a discriminator unigram distribution built from background material.
- For a test story s of size N, the cluster story distance S_D is given by

$$S_D = \sum_{w \in V_s} n_w \left[\log p^C(w) - \log p^X(w) \right] - t \ ,$$

where $p^C(w)$ is the unigram probability of word w in the smoothed cluster model, and $p^X(w)$ is the probability of word w in the discriminator model. The parameter t is an optional *time penalty,* which depends on the separation in time between the test stories and the stories that make up the cluster, and can be used to suppress the clustering of new stories with "stale" clusters.

Note the similarity between this measure (with a time penalty of zero) and the Tracking measure used for the Unigram Tracker. In fact, the only differences are that the Detection measure does not filter the background distribution, and uses a smoothing algorithm that we have rejected for poor performance in the Tracking task.

Given our interest in leveraging the statistical models, an obvious place to look for a candidate for a better Detection measure is one of the Tracking models described in the previous section—in other words, at each point in the clustering process that a story is compared to a cluster, the cluster can be treated as a story collection for which we can build a "topic model," and, given a discriminator model built from background data, one of the Tracking measures can be used as the cluster–story distance. In fact, since the Tracking problem is computationally much simpler than the Detection problem (at least as they are formulated by us), we can go one step further and use Tracking experiments to compare candidate Detection measures.

To that end, Figure 6.7 shows the effect of using the TDT 1998 Detection measure in the Tracking task, compared to the Unigram and Beta-Binomial measures, on the TDT 1998 evaluation data. The detection measure is clearly inferior.

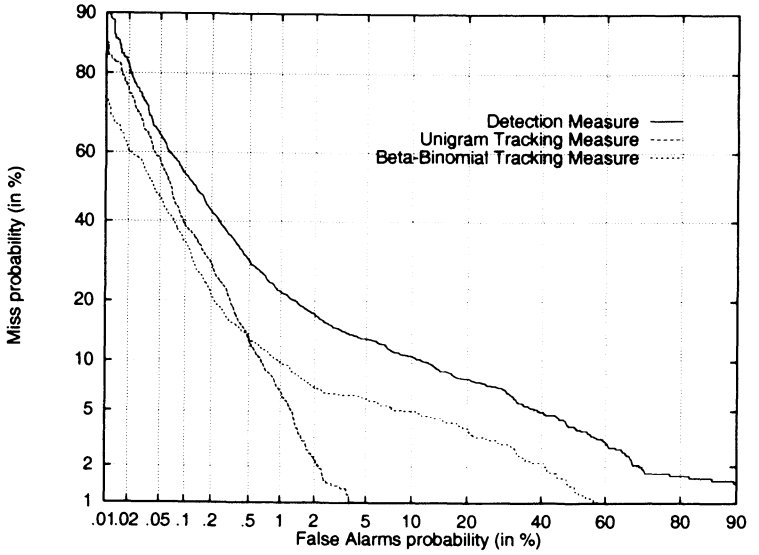


Figure 6.7. Detection Measure vs. Tracking Measure

Unfortunately, the obvious idea of applying one or both of the Tracking measures to the Detection task has so far proven too difficult to implement, due to computational complexity. (In the Detection task, the measure must be computed every time a cluster is updated, which is quite frequent due to the iterative nature of the clustering algorithm.) For the Beta-Binomial measure the obstacle is the computation of the beta function; for the Unigram measure the problem is the targeting step.

A new measure. As described earlier, some experiments leading up to the TDT 1999 evaluation showed that simply switching to smoothing via interpolation in the Tracking measure resulted in a significant gain in performance. The same idea can be applied to the Detection measure: keep the same form of the measure as above, but replace the discount-backoff smoothing with interpolation. Figure 6.8 shows the performance of this measure on a tracking task, compared to the TDT 1999 Detection measure (which, like the TDT 1998 measure, used discount-backoff smoothing) and the TDT 1999 Tracking measure.

The obvious next step is to try the new measure at Detection. Figure 6.9 shows a comparison of the Detection performance of the old *vs.* the new measure. Despite the extremely jumpy behavior, the new smoothing is clearly superior, and achieves gains of 15–20% in the value of C_{det}.

The wide variability in the value of C_{det} as the number of clusters is varied is a problem for both measures, and makes it difficult to choose an operating point (which corresponds to tuning the number of output clusters) that optimizes

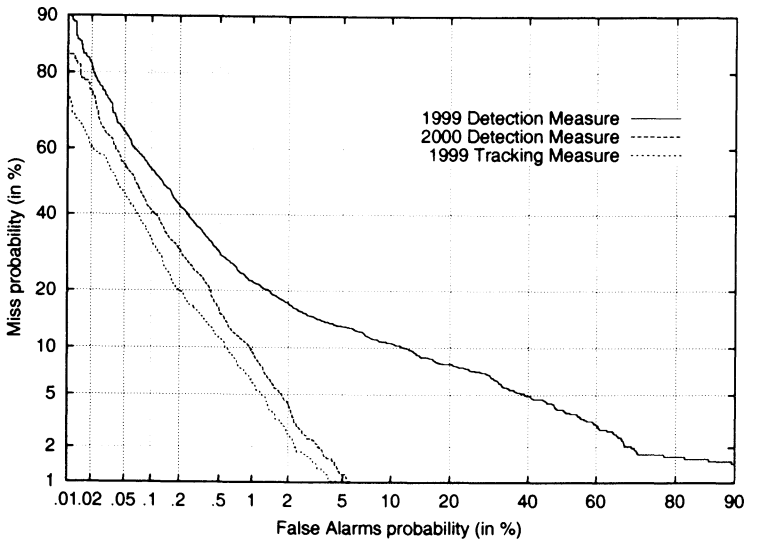


Figure 6.8. Detection Measure vs. Tracking Measure

performance. Some of the modifications described below have the effect of smoothing this behavior somewhat.

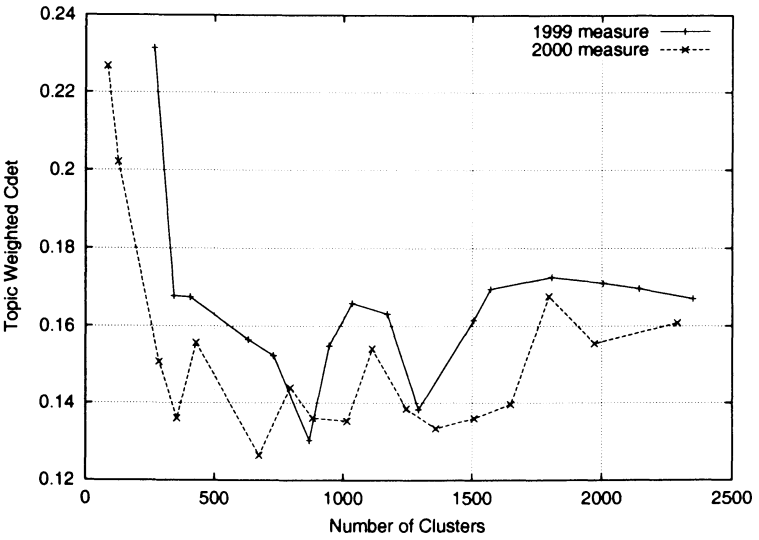


Figure 6.9. Performance of 1999 Measure vs. 2000 Measure

Use of targeting. We made an attempt to take advantage of targeting to improve the smoothing used in our detection system. The computation is too

expensive to incorporate into the update of the clusters, so instead we used targeting to smooth the incoming stories. This is backwards—we *should* be broadening the cluster distributions to better score the sparse incoming stories, not smoothing the incoming stories to create more overlap with the clusters. Nevertheless, smoothed incoming stories do lead to smoother clusters, since the story statistics are merged into the cluster to which they are assigned (although the result is not the same as if the cluster itself had been smoothed by targeting).

For each story in the development test corpus we computed a targeting mixture distribution. This distribution was then used to smooth via interpolation the story from which it was derived. Detection was run using the smoothed stories as the incoming stream instead of the original stories. The interpolation weight was globally tuned, and the best choice (15% weight on the targeting mixture) is shown in Figure 6.10, compared to the run without targeting. The result is slightly worse, but substantially flatter and more predictable. On the basis of this it was submitted as our primary system for the TDT 2000 evaluation, with the untargeted run submitted as a secondary.

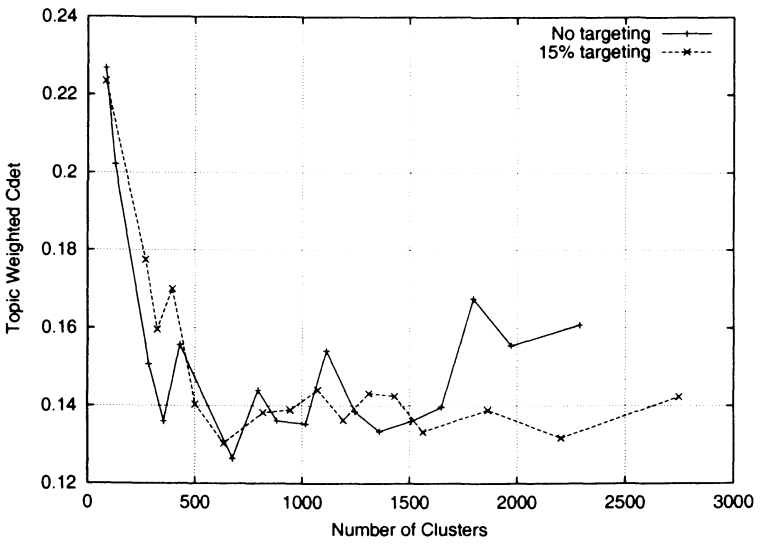


Figure 6.10. Targeting the Incoming Stories

5. Summary

We have described the application of two statistical models of story generation, one based on simple unigram distributions and another based on the beta-binomial distribution. The unigram model is computationally simple, and the research issue that arises in its application is the same one familiar from

speech recognition (although in a more extreme form): smoothing of a sparse distribution. We have developed the targeting procedure and experimented extensively with smoothing algorithms to optimize the smoothing for the TDT tasks. As we approach the limits of what can be done with unigrams, we will be looking at extending the model with other features, the most obvious being bigrams.

The application of the beta-binomial distribution to information extraction tasks is new. The model is computationally complex, but its addition of an extra parameter in the modeling of word statistics (the variance) allows the model to better measure the topical significance of words in a story collection. The structure of this model makes it possible to add virtually any feature that can be counted in training data; future candidates include bigrams and collocations.

When applied to the problems of topical analysis embodied by the TDT tasks, the models exhibit different strengths. The Unigram system is very good at generalizing from small amounts of available training data, but not as good at distinguishing the topic training stories from closely related, but off-topic, material. The Beta-Binomial system is very good at distinguishing high-scoring false alarms from the topic training material, but not particularly good at generalizing from that material. Some ideas we are pursuing to improve the performance of both systems include stemming (to improve word statistics), "smart" adaptation that distinguishes small topics from large, and better use of available background data (particularly for the Beta-Binomial system).

References

[1] J.P. Yamron, I. Carp, L. Gillick, S.A. Lowe, and P. van Mulbregt, "Event Tracking and Text Segmentation via Hidden Markov Models," *Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop,* Santa Barbara, December 1997.

[2] J. Allan, J. Carbonell, G. Doddington, J.P. Yamron, and Y. Yang, "Topic Detection and Tracking Pilot Study: Final Report," *Proceedings of Broadcast News Transcription and Understanding Workshop,* Lansdowne, Virginia, February 1998.

[3] J.P. Yamron, I. Carp, L. Gillick, S.A. Lowe, and P. van Mulbregt, "A Hidden Markov Model Approach to Text Segmentation and Event Tracking," *Proceedings ICASSP-98,* Seattle, May 1998.

[4] P. van Mulbregt, J.P. Yamron, I. Carp, L. Gillick, and S.A. Lowe, "Text Segmentation and Topic Tracking on Broadcast News via a Hidden Markov Model Approach." *Proceedings ICSLP-98,* Sydney, December 1998.

[5] P. van Mulbregt, I. Carp, L. Gillick, S.A. Lowe, and J.P. Yamron, "Segmentation of Automatically Transcribed Broadcast News Text," *Proceedings of the DARPA Broadcast News Workshop,* February 1999.

[6] J.P. Yamron, I. Carp, L. Gillick, S.A. Lowe, and P. van Mulbregt, "Topic Tracking in a News Stream", *Proceedings of the DARPA Broadcast News Workshop,* February 1999.

[7] S.A. Lowe, "The Beta-Binomial Mixture Model and Its Application to TDT Tracking and Detection," *Proceedings of the DARPA Broadcast News Workshop,* February 1999.

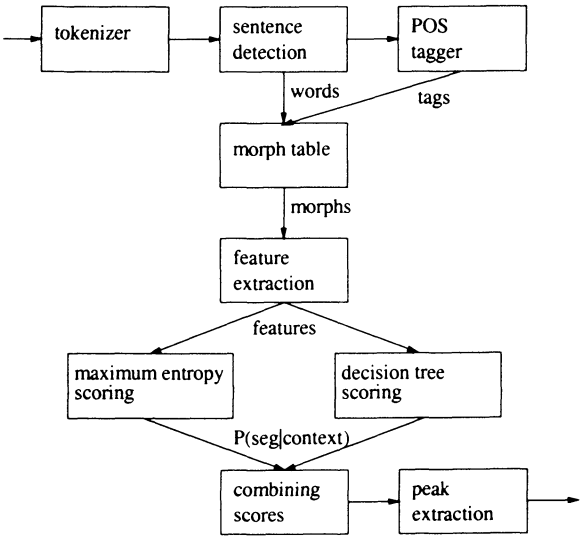
[8] S.A. Lowe, "The Beta-Binomial Mixture Model for Word Frequencies in Documents with Applications to Information Retrieval," *Proceedings of Eurospeech '99,* Budapest, September 1999.

[9] J.P. Yamron, L. Gillick, S. Knecht, and P. van Mulbregt, "Statistical Models for Tracking and Detection", *Proceedings of the DARPA Topic Detection and Tracking Workshop,* February 2000.

[10] S. Katz, "Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer," *IEEE Transactions on Acoustics, Speech and Signal Processing,* ASSP-35(3):400–401, March 1987.

[11] H. Ney, U. Essen, and R. Kneser, "On Structuring Probabilistic Dependences in Stochastic Language Modelling," *Computer Speech and Language,* 8:1–38, 1994.

Figure 7.1. Components of the Segmentation System



P_{target} and $P_{non-target}$ are the a priori target probabilities. In this paper we use $P_{target} = 0.3$ and $P_{non-target} = 0.7$. We also report a normalized score

$$C_{norm} = C_{seg}/min(C_{miss}P_{target}, C_{FA}P_{non-target}) \tag{7.2}$$

which scales the scores to the 0 to 1 range.

1.2 System Overview

IBM's story segmentation algorithm combines two different classes of probability models to compute the probability that each of the ASR transcript's non-speech events represents a story boundary. The first of these models is a binary decision tree model. The second is a maximum entropy model [1, 2] which incorporates additional features based on n-grams, the rate of speaking, and the time elapsed since the beginning of the show. To decode the sequence of story boundaries in a broadcast, we consider each non-speech event in turn, combining the two models' probabilities, and placing a story boundary on events whose score exceeds a threshold. Additionally, we experimented with training separate models for individual domains. C_{norm} values for this algorithm on the TDT'99 evaluation are 0.3857 and 0.3192 for the English and Mandarin corpora, respectively.

The architecture of our English segmentation system is depicted in Fig. 7.1. The upper five components form a feature extraction pipeline which converts ASR transcripts into input features for the probability models. The sentence detector chunks the transcript into "sentences" consisting of a string of words

delimited on either end by non-speech events such as silence. These sentences are tagged by an HMM part of speech tagger and then stemmed by a morphological analyzer which uses the part-of-speech information to reduce stemming ambiguities. The feature extractor extracts many features from this input stream, for example, the number of novel nouns in a short lookahead window. The probability of the presence of a story boundary at any given position in the text is then computed separately using decision-tree and maximum entropy models, discussed in more detail below. A weighted linear combination of the probabilities produced by the above models is then computed. A peak-picking algorithm with source-specific thresholds makes the segmentation decision based on the combined probabilities. We have also experimented with a refinement stage, based on our detection similarity measure, in which we remove posited boundaries between stories that are topically very similar. This refinement stage yields considerable improvements in situations where false-alarms are costly, such as they were in the TDT'98 evaluation [3, 4].

1.3 Decision Tree Model

The decision tree model is similar to the one we described in the 1999 Broadcast News workshop [3, 4] and used in our TDT'98 system. It uses three principal groups of input features to model the probability of a story boundary. The most important group of features are questions about the duration of non-speech events in the ASR output, many of which are silences or pauses. The second group of features depends on the presence of key words and bigrams indicating story boundaries (automatically learned by a mutual information criterion.) The last family of input features compares the distribution of nouns on the two sides of the proposed boundary. The top three layers of the decision tree, the feature questions, and the resulting probabilities of segmentation are shown in Figure 7.2. We use a mixture of three decision trees of various depths in our scoring.

Figure 7.2. Top layers of the segmentation decision tree

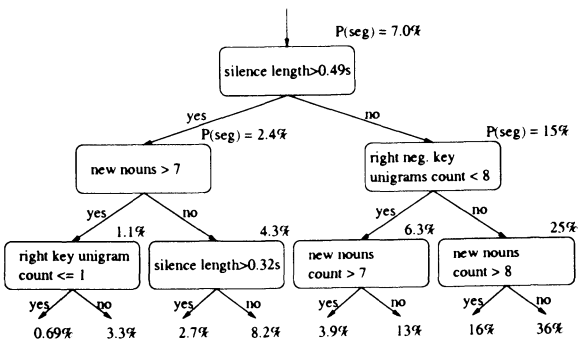


Table 7.1. Examples of ME model n-gram features, indicating (+) and counter-indicating (−) boundaries

window	n-gram	indicator
right	sport	+
right	weather	+
right	be next	+
right	time for	+
left	story and more	+
right	we come back	+
right	talk of	−
right	one eight hundred	−
left	so	−
left	yeah	−
left	rest of the	−

1.4 Maximum Entropy Model

Our Maximum Entropy model uses three categories of features. First, there are features encoding the same information used by the decision tree model. The next group of features looks for n-grams extracted from windows of ten words to the left and right from the current point. The n-grams ($n \leq 3$) are selected automatically during the training. Table 7.1 shows examples of n-gram features indicating (+) and counter-indicating (−) the presence of a boundary. The last category of features are structural, and detect large-scale regularities in the broadcast programming, such as specific time slots for commercials. Note that these types of features are frequently source-specific. The rate of speaking is also used as a feature, motivated by the observation that the speaker tends to speak faster at the beginning of a new story.

1.5 Mandarin Data Processing

We train our models for the Mandarin part of the corpus in a manner very similar to the English subsystem. The most important differences arise from the

Table 7.2. ASR vs Manual Transcripts, TDT-3, English

	P_{miss}	P_{fa}	C_{seg}	C_{norm}
ASR	0.1430	0.1814	0.0810	0.3857
transcripts	0.1561	0.1963	0.0881	0.4194
trans.+non-sp.	0.1492	0.1768	0.0819	0.3900
ASR+transcripts	0.1249	0.1734	0.0739	0.3519

Figure 7.3. Performance characteristic - ASR vs Manual Transcripts

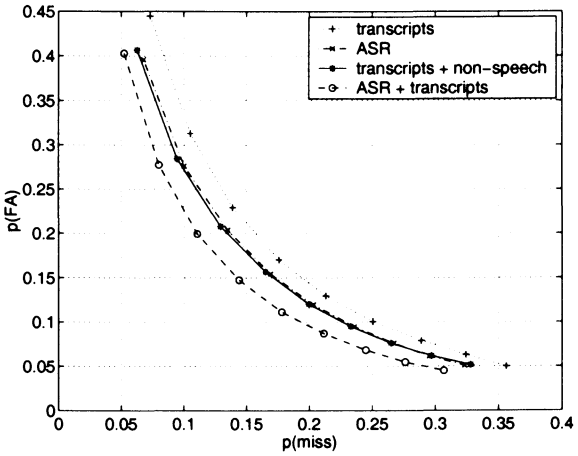


Table 7.3. Summary of TDT-3 English segmentation experiments

	P_{miss}	P_{fa}	C_{seg}	C_{norm}
DT	0.2148	0.2131	0.1092	0.5200
ME	0.1501	0.1818	0.0832	0.3962
DT + ME	0.1430	0.1814	0.0810	0.3857

Table 7.4. Summary of TDT-3 Mandarin segmentation experiments

	P_{miss}	P_{fa}	C_{seg}	C_{norm}
DT	0.1719	0.1245	0.0777	0.3701
ME	0.2288	0.0647	0.0822	0.3916
DT + ME	0.1528	0.1009	0.0670	0.3192

fact that we do not (yet) have a tagger and a morphological analyzer available for Mandarin. This means we are not able to identify nouns in the Mandarin

Table 7.5. Summary of TDT-3 Mandarin segmentation experiments, optimal thresholds

	P_{miss}	P_{fa}	C_{seg}	C_{norm}
DT	0.1428	0.1606	0.0765	0.3645
ME	0.1176	0.1655	0.0700	0.3334
DT + ME	0.1203	0.1351	0.0645	0.3070

transcript. Instead, features inquire about the distribution of content words. We use a simple set of rules to identify content words: i) First we use a list of 1975 "stop" words and 568 content words. ii) If a given word in not found in either of the two lists, we check it for the presence of one of the 306 "stop" characters. iii) If it contains any of them, the word is considered to be a "stop" word; otherwise, it is marked as a content word.

Figure 7.4. Performance characteristic - TDT-3, English

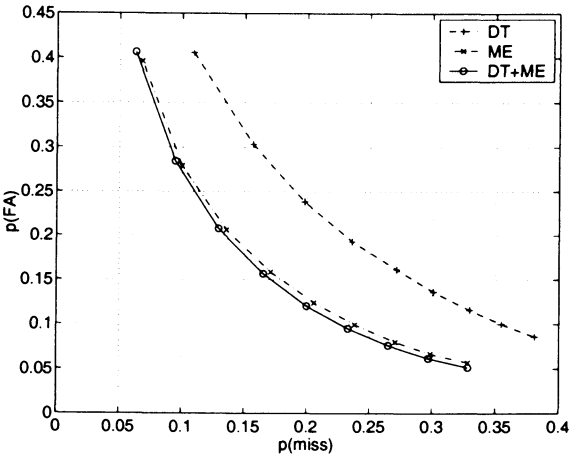
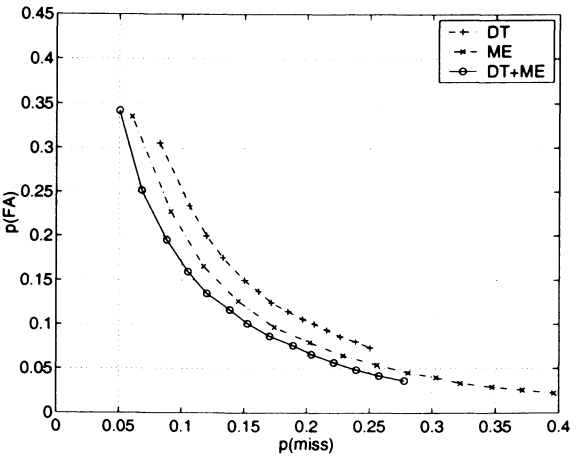


Figure 7.5. Performance characteristics - TDT-3, Mandarin



1.6 Effect of ASR Errors

We have investigated the effect of speech recognition errors on segmentation cost by segmenting the manually transcribed (close-captioned and FDCH transcribed) text with our system. For a fair comparison, it is necessary to control for the effects of the most important prosodic feature, the length of non-speech events. In order to control for this feature, we used a Viterbi alignment algorithm based on individual letters to find corresponding word positions in the ASR and manual transcripts. Then, for each non-speech event in the ASR, we inserted a non-speech event at the corresponding word boundary in the manual transcript. We note that the alignment is not perfect because the transcripts occasionally paraphrase the spoken information, and for some stories the transcripts are very short or missing. With the ASR and manual transcripts thus aligned, it is possible for us to include the duration of non-speech events, which is an important source of information, as a feature in models trained on manually transcribed text. Table 7.2 and Figure 7.3 compare the performance of our models on ASR and manual transcripts. The C_{seg} values for the ASR output and the manually transcribed data with non-speech event length included in the model are very close, suggesting that the performance degradation caused by the ASR errors is quite small. We note that C_{seg} values for manual transcripts were computed based on corresponding word positions in the ASR, not actual word positions in the manual transcripts. Omitting the non-speech event length from our models for the transcribed data causes substantial performance reduction. Combining the segmentation probabilities obtained from the ASR and manual transcripts improves the performance considerably.

1.7 Results and Conclusion

The results of our TDT-3 story segmentation experiments are summarized in Tables 7.3 and 7.4 and Figures 7.4 and 7.5. The results show the following two effects: first, the performance of the segmentation system improves by combining the decision tree and maximum entropy models. Second, in the English task, the performance of the maximum entropy model is stronger than that of the decision tree model, whereas in the Mandarin task the performance of the two models is quite similar, with the DT model being slightly stronger. This reversal effect can be partially explained by the fact that the probability threshold used in the Mandarin ME model was estimated using the TDT-2 corpus. This threshold determines an operating point of the model which doesn't match the TDT-3 data well. To validate this intution, we have redecoded using thresholds near the optimal TDT-3 values. As shown in Table 7.5, the ME model does outperform the DT model on the Mandarin task with tuned thresholds.

2. Topic Detection

The purpose of the topic detection task is to investigate the nearly-synchronous, unsupervised clustering of the stories by topic. By unsupervised, we mean that no topic descriptions, queries, or example stories are provided for any topic. By nearly-synchronous, we mean that several months of news stories are presented in chronological order, and cluster membership decisions about each document must be made shortly thereafter (less than 1 day "corpus" time.)

To measure the performance of a detection algorithm, misses and false alarms are defined with respect to a manually annotated labeling of the stories by topics. Since the correspondence between system clusters and the manual annotation is ambiguous, the scoring gives the benefit of the doubt to the system and chooses the best mapping of system clusters to annotated topics. The costs reported here are a topic-weighted, linear combination of miss and false-alarm probabalities, as in segmentatoin.

IBM's detection system is a *two-tiered* clustering system: each cluster is composed of several *microclusters*. In our system, stories are assigned to microclusters, and then – at the end of the deferral period – microclusters are assigned to clusters. Only the clusters (the upper tier, or coarser granularity) are reported for evaluation purposes. The underlying process for creating the microclusters is an incremental clustering process in which a story-microcluster similary measure is thresholded in order to control microcluster growth/formation. The similarity measure is both time-dependent and cluster-dependent, based on our TDT'98 system [3]. The primary theme of the topic detection section of this paper is the motivation for the two-tiered clustering, and how it differs from a more traditional system. In this section, we first describe our basic detection system (the assignment of stories to microclusters in more detail.) Then we address some issues associated with incorporating Mandarin into the system. We motivate the additional stage of merging microclusters into clusters with a performance study of the basic detection system on monolingual subsets of the corpus. To remedy the defects this exposes, we introduce a second tier of clustering and comment on important differences between the first and second tiers of clustering. A secondary theme throughout the section is the difference in the choice of word weights for various aspects of the task.

2.1 Microclusters

At the core of our system is a symmetrized version of the Okapi formula [5] to score the similarity of two stories.

$$\mathrm{Ok}\left(d^1, d^2\right) = \sum_{w \in d_1 \cap d_2} t_w^1 t_w^2 \lambda(w, \mu) \tag{7.3}$$

where the story representation component t_w^i for each word w in story d^i is obtained by length-normalizing and warping the term count c_w^i,

$$t_w^i = \frac{\hat{t}_w^i}{a + \hat{t}_w^i} \text{ where } \hat{t}_w^i = \frac{c_w^i}{\sum_w c_w^i}. \tag{7.4}$$

The words are weighted by a word-weight $\lambda(w, \mu)$, which, in general, is allowed to be microcluster-dependent, and also time-dependent through changes in the microcluster content. Our word-weight differs from traditional definitions of inverse story frequency in several respects. We have adapted our word weights to be both time-dependent and microcluster-dependent and we obtain significant improvements in performance because of this adaptability. We write $\lambda(w, \mu) = \text{idf}_0(w) + \Delta\lambda(w, \mu)$, where $\text{idf}_0(w)$ is the standard inverse story frequency of the word and $\Delta\lambda(w, \mu)$ is a measure of the similarity of two sets of stories: $\mathcal{D}_w$, the set of stories that contain the word w, and the set of stories in microcluster μ. In fact, we choose

$$\Delta\lambda(w, \mu) = \lambda_0 \frac{2n_{w,\mu}}{|\mathcal{D}_w| + |\mu|} \tag{7.5}$$

where $n_{w,\mu}$ is the number of stories in $\mathcal{D}_w \cap \mu$, and λ_0 is an adjustable scaling parameter. This quantity can be interpreted as a harmonic mean of a "recall" and a "precision." (If $\mathcal{D}_w$ is interpreted as a set of relevant stories, and μ as a set of retrieved stories, then, for important words, the "precision" indicates that most stories in the cluster contain the word, and the "recall" indicates that the cluster contains most occurrences of the word.) Note that $\Delta\lambda(w, \mu) = 0$ if and only if $\mathcal{D}_w \cap \mu$ is empty and $\Delta\lambda(w, \mu) = \lambda_0$ if and only if $\mathcal{D}_w = \mu$.

We represent each microcluster internally as the centroid of the story representation components t_w of the documents in the microcluster. The score of a story with a microcluster $\text{Ok}(d, \mu)$ becomes the *mean* of the scores of that story with the stories contained in the microcluster. This scoring function is ultimately compared against a threshold Θ: Generally the story is merged into the microcluster μ^* which maximizes $\text{Ok}(d, \mu)$; if $\text{Ok}(d, \mu^*) < \Theta$ then the story becomes the seed of a new cluster. In other circumstances, discussed previously [3], we label a story as belonging to μ without updating the statistical description. We will present our system performance across a range of Θ. We regard Θ as a control parameter that allows us to select a system operating point (manifested by mean microcluster size.) On the other hand, tunable parameters (such as λ_0) were trained on the TDT-2 corpus and frozen for this series of experiments.

2.2 Mandarin

The incorporation of Mandarin stories into the TDT'99 task brings many new issues. We believe that it is important to contrast the difference between

measuring the performance of two separate monolingual systems (source language (SL) = English, and source language = Mandarin) and measuring the performance of a single multilingual system. We present experiments on both, because different issues can be addressed by each experiment. We also present some experiments varying clustering language (CL). We present our experiments across a range of operating points, and make comparisons at constant Θ rather than constant mean cluster size because of our interest in merging issues: this is the parameter that can easily be controlled in multilingual runs.

In Fig. 7.6 we contrast four approaches to topic detection on the Mandarin subset of the TDT-3 stories. The first three approaches use the Systran translation of the Chinese stories; the first two of these mix English and (originally) Chinese stories. In the first approach we show the score of the *subset* of Mandarin stories, with respect to a set of multilingual (SL=mul, CL=eng) topic detection runs. In the second, we *extract* the Mandarin stories from the above multilingual runs and score only on them. Note that the subset score will always have higher cost than the extract score, because the scoring alignment between topics and system clusters will be chosen optimally. Nevertheless, the difference between these two runs reflects a tendency for the Systran translated stories to cluster together because of idiosyncrasies introduced in translation, rather than on a topical basis, a phenomenon we call *cohesion.* We also show a set of detection runs, labeled *Systran*, on the (SL=man, CL=eng) stories (that is, no English stories in the detection run) which performs considerably worse than the *extract* run. Finally, for comparison purposes we also evaluate a baseline *native* Chinese (SL=man, CL=nat) topic detection system which is a direct adaptation of English system. (Here our "vocabulary" consists of Chinese character bigrams.) We have also observed that at very high thresholds (well larger than optimal with respect to the TDT'99 metric,) the pure Systran runs continue to improve and eventually have lower C_{det} than even the native runs. The significance of this observation, (for example, with respect to granularity of "word" size in Chinese) is still under investigation. A similar set of experiments, shown in Fig. 7.7 shows a much smaller difference in the performance of the three corresponding approaches (English to Chinese MT not available) to English topic detection. This observation is not surprising because of the large number of English sources and stories in the corpus.

Although it appears advantageous to work with the Mandarin stories in the native representation, it is unclear how to merge Mandarin clusters in the native representation into the multilingual clusters required by the TDT'99 task. Because the primary task is the multilingual system, we have focused our research on how best to use the translated stories. A principal challenge is how to mitigate the effects of cohesion.

Figure 7.6. Normalized cost of detection as a function of primary system threshold for four Mandarin detection systems (see text for explanation.)

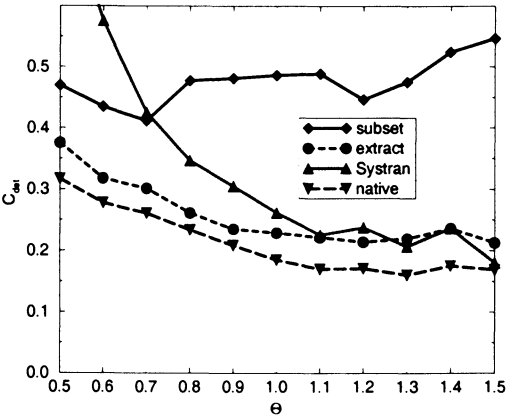
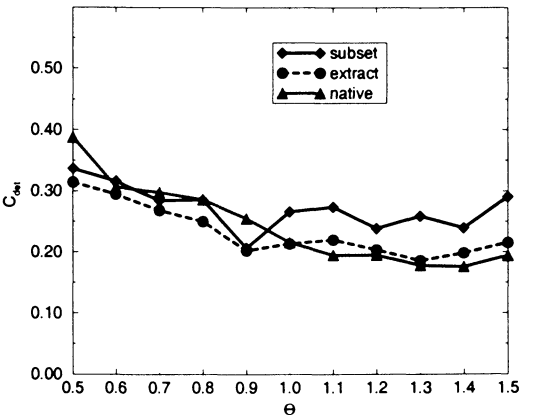


Figure 7.7. Normalized cost of detection as a function of primary system threshold for three English detection systems (see text for explanation.)



2.3 Two-tiered clustering

Our strategy to counteract cohesion is to form a two-tiered clustering. Stories are initially assigned to microclusters as described above. After the microclusters accumulate more stories (to the extent allowed by the deferral period), the microclusters are then grouped into the actual clusters which we report. Because our microclusters are represented by a centroid, our story-story and story-

microcluster versions of Okapi score are easily generalized into a microcluster-microcluster Okapi score. A microcluster is assigned to a cluster by assigning it to the same cluster as the most similar microcluster, or starting a new cluster if none of the scores are sufficiently high. We note an important contrast between the two tiers of clustering : a story-microcluster score represents a mean over story-story scores,

$$\mathrm{Ok}(d,\mu) = \frac{1}{|\mu|}\sum_{d'\in\mu}\mathrm{Ok}(d,d') \tag{7.6}$$

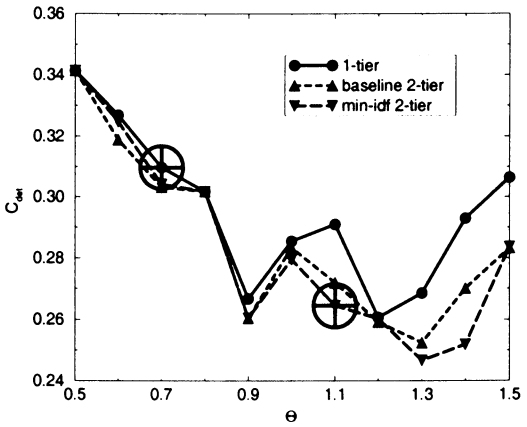
whereas a microcluster-cluster score represents a maximization over microcluster-microcluster scores. Thus

$$\mathrm{Ok}(\mu,cl) = \max_{\mu'\in cl}\mathrm{Ok}(\mu,\mu'). \tag{7.7}$$

The decision to either merge a microcluster into a cluster or to start a new cluster is analagous to the decision to merge a story into a microcluster or to start a new cluster: we compare $\mathrm{Ok}(\mu, cl)$ against a second threshold Θ_2. We note that in the limit $\Theta_2 \to \infty$, we recover the basic 1-tier clustering system. We further note that it is efficient to cluster across many values of Θ_2 simultaneously because no cluster-level statistics need ever be computed. The intent of the two-tiered clustering is to reduce cohesion and make cluster assignments more dependent upon topic and less dependent upon source. However, the cause of cohesion ultimately lies in different word-use statistics associated with each source. Lists of words that are "disproportionately common" for each source clearly show both topical effects (e.g. sports reports occur in only some sources) and source-based effects (e.g. reporter names, speech recognition and machine translation errors.) These observations suggest that $\mathrm{Ok}(\mu,\mu')$ scoring formula should somehow attempt to compensate for the source- dependent variations by lowering the weight of words that are disproportionately common in a source. A simple approach here is to compute source-specific idf's, and then, for scoring, choose the minimum, across all sources, of the source-specific idf's. We will denote this set of word-weights as $min-$ idf.

In Fig. 7.8, we summarize the performance our two-tier clustering system across a range of thresholds appropriate to the TDT'99 definition of C_{det}. We remind the reader that these results are for a mixture of native English and Systran-translated Chinese broadcast news and newswire. The upper curve is the baseline system obtained at $\Theta_2 \to \infty$. The lower two curves represent *lower bounds* (minimization with respect to Θ_2) on C_{det} for baseline idf's and $min-$ idf's, respectively. Our submission system, a $min-$ idf system, with Θ_2 as trained from the TDT-2 corpus is highlighted. We see that the gains are quite erratic, sometimes large and sometimes nearly negligible. However, the gains seem to increase as the Θ increases, in other words, merging microclusters

Figure 7.8. multilingual detection: baseline (1-tier system) vs. 2-tier system for two sets of idf's. We have plotted lower bounds on performance. Submission systems are marked by $\oplus$.



becomes more important as the mean microcluster size decreases. On the other hand, the optimal Θ_2 parameter transferred well from the training data. The min – idf's appear to outperform the baseline idf's by a small margin. This observation could not have been made on the training set because too few multilingual topics were annotated.

3. Acknowledgements

This work is by DARPA under SPAWAR contract number N66001-99-2-8916. We would like to thank Salim Roukos for valuable discussions, and Kishore Papineni for valuable discussions and for designing and implementing the maximum entropy code.

References

[1] D. Beeferman, A. Berger, and J. Lafferty, "Statistical Models for Text Segmentation", *Machine Learning*, vol. 34, pp. 1-34, 1999.

[2] S. Della Pietra, V. Della Pietra, J. Lafferty, "Inducing Features of Random Fields", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, p.380 1997.

[3] S. Dharanipragada, M. Franz, J.S. McCarley, S.Roukos, T.Ward, "Story segmentation and Topic detection in the Broadcast News Domain", *Proceedings of the DARPA Broadcast News Workshop*, pp. 65-68, 1999.

[4] S. Dharanipragada, M. Franz, J.S. McCarley, S. Roukos, T. Ward, "Story Segmentation and Topic Detection for Recognized Speech", *Proceedings of Eurospeech*, pp. 2435-2438, Budapest, Hungary, September 1999.

[5] S.E. Robertson, S. Walker, S. Jones, M.M. Hancock-Beaulieu, M. Gatford, "Okapi at TREC-3", *Proceedings of the Third Text REtrieval Conference (TREC-3)* ed. by D.K. Harman, NIST Special Publication 500-225, 1995.

[6] "The Topic Detection and Tracking Phase 3 (TDT-3) Evaluation Plan", Version 2.7, Aug. 10, 1999, http://www.itl.nist.gov/iaui/894.01/tdt3/tdt3.htm

Chapter 8

A Cluster-Based Approach to Broadcast News

David Eichmann and Padmini Srinivasan
School of Library and Information Science
The University of Iowa
Iowa City, Iowa 52242

Abstract We present an approach to detection and tracking of topics in multilingual broadcast news based upon a dynamic clustering scheme. Our approach derives from a system used to filter Web searches from multiple sources, with extensions for pipelining document clusters, part-of-speech tagging and extraction of named entities for use in an extended similarity measure.

1. Introduction

The task of information analysis is awash in a sea of material: hundreds of television channels, thousands of newspapers, millions of Web servers serving billions of Web pages. Traditional information retrieval systems were constructed with an assumption that the material that they indexed exhibited a significant degree of homogeneity with respect to format and location. This is no longer the case. We are developing an integrated architecture of extensible components supporting a broad spectrum of 'document' types, including newswire, broadcast news (through both speech recognition and closed captioning) and the Web, allowing analysis of a rich mix of language, media and format in a single user interface. Our participation in the Topic Detection and Tracking (TDT) effort arose out of a desire to quantitatively evaluate system performance.

1.1 The TDT Scenario

Consider the manipulation requirements of an information analyst. The traditional requirements of an intelligence analyst – large volumes of relatively unstructured data (frequently in text form), multiple concurrent sources of infor-

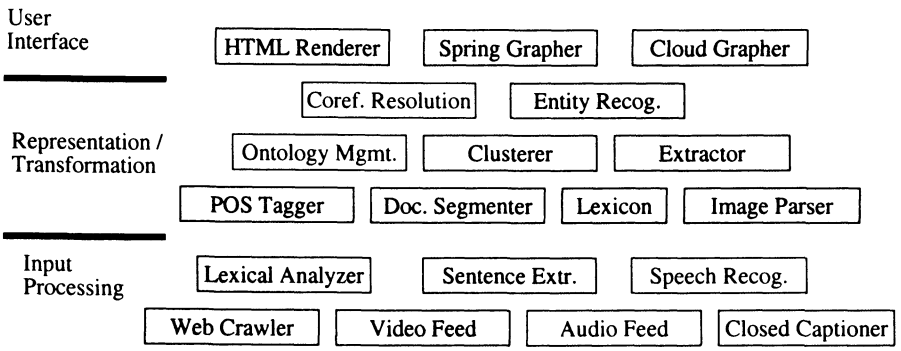


Figure 8.1. Our evolving architecture. Existing components are shaded.

mation, multiple source languages, support for tuning of the trade-off between precision and recall, etc. – have become common requirements for today's corporate competitive intelligence analyst.

1.2 System Architecture

Cowie and Lenhert have noted [3] that "[t]he idea that good software development always requires good software engineering is worth repeating in the extraction context, because natural language processing (NLP) researchers do not always think of software engineering as a critical NLP research weapon. As NLP researchers become involved in corpus-driven research, speed and efficiency suddenly matter when a large text corpus has a central role in a research program." Our experience in rapid development of systems using established software engineering principles is a key factor in the uniqueness and strength of our approach.

Our system architecture readily supported extension and enhancement of the components which we have built for Sulla[4, 5] and TREC [6, 7, 8]. The architecture supports a variety of task-specific configurations through observer/observable design patterns [9]. The boundaries between the architectural layers shown in Figure 8.1 are implemented as a variation of an Observable object, where the Observer registers not against a specific object of the class of interest, but rather against a class sentinel. The sentinel notifies observers of the creation of a new class instance, rather than of a state change in an already existing object (the usual semantics of a notification to an observer). This allows us to construct new components that function as producers or consumers of information with no knowledge of the consumer(s) or producer(s) of that information. The boundaries in Figure 8.1 are hence API classes that insulate the components the respective layers not only from the implementation details

of components in the other layer, but also any knowledge of the existence of any given component in the other layer. A task-specific application program is then just a wrapper class that acquires runtime parameters and instantiates the components required for the task.

This approach has a further beneficial ability to scale readily. By incorporating lightweight producer and consumer surrogates that use Linda-like tuple spaces [13], we have the capability to parallelize tasks across loosely-coupled networks of processors. Our TDT2000 tracking runs were done using three memory-rich workstations for lexical analysis, part-of-speech tagging and entity extraction and twenty-eight medium-scale workstations each serially processing a single topic. Task allocation was accomplished simply by having an initiation process populate the tuple space with the list of topic definitions and source files to be processed for the run.

1.3 TDT Tasks

Turning back to our scenario, there are a set of naturally occurring tasks that a system supporting an analyst must implement:

Segmentation – Newswires and Web pages have naturally occurring boundaries delimiting 'documents,' the typical unit of retrieval. This is not the case for broadcast news, where anchorpersons slide relatively seamlessly from story to story and commercials occur in rapid-fire succession. For our purposes then, segmentation is the task of declaring boundaries between broadcast news stories (and potentially, commercials), so that downstream tasks can focus on the stories as the unit of analysis. In a more abstract sense, segmentation is a filter that consumes a polymorphic input stream (a mix of story streams from newswire and word streams from broadcast news) and produces a monomorphic output stream (a mix of newswire and broadcast news stories).

Detection – Given a stream of stories, detection is the identification of stories that are 'new' and the subsequent stories that discuss the same topic.[1] It can be naturally viewed as a clustering problem, where the clustering must be incremental (you don't want to wait forever for 'all' stories to arrive just to be able to pick the best match to cluster with the first story). A deferral period is sometimes used to allow for the notion of grouping the news for the day or week prior to merging it with the preceding material. Detection is inherently an unsupervised activity, requiring no guidance from the user other than potentially setting threshold parameters.

Tracking – Unlike detection, tracking presumes that the user has already identified a topic by means of one or more exemplar stories, and any subsequent stories should be retrieved for the user. TDT tracking differs from the TREC

[1] There is a long (for TDT) history to the question of "what is a topic" – see [1] for further information.

adaptive filtering task [7, 8] through the restriction that feedback from the user is not provided during a tracking run.

While there are additional tasks defined in the TDT framework, we chose to focus on these as the most generally occurring in an information discovery and monitoring environment.

1.4 Measuring Similarity

We use a vector space model to assess document-cluster and cluster-cluster similarity using a straight-forward vector cosine measure:

$$sim(d,c) = \frac{\sum_{i=1}^{N_d}\sum_{j=1}^{N_c}(TF(W_i)\cdot TF(W_j))}{\sqrt{\sum_{i=1}^{N_d} TF(W_i)\cdot\sum_{j=1}^{N_c} TF(W_j)}}$$

where $TF(W_i)$ is the (current) $TF{\cdot}IDF$ weight for term W_i in a given document or cluster. Term frequencies are built up incrementally as a given run progresses and cluster term weights are adjusted every ten input files. This approach is therefore somewhat inaccurate in the initial phases of a run, but quickly reaches a point of reasonable stability with respect to term frequencies and has the added benefit of requiring no fore-knowledge of the vocabulary. Weights are incrementally updated every 10 input files. All vocabulary is stemmed using Porter's algorithm [11] and filtered through a stoplist. We prune document term vectors to the 100 most weighty terms (i.e., those with the highest $TF{\cdot}IDF$ values) and cluster vectors to the 200 most weighty terms. This proves to have no significant effect on the accuracy of our results, but a significant effect on both memory requirements and execution time, the latter due to a corresponding reduction in the cost of dot product calculations.

2. Segmentation

As the lead task in the TDT framework, we felt it useful to attempt segmentation at least once. Text segmentation was performed with an agglomerative clustering approach. Clusters were built iteratively from the word level up, combining neighboring clusters as long as sufficiently similar neighboring clusters appeared in the deferral window. The result is a very fast and flexible algorithm that will be extended in several natural ways to increase its accuracy. We note that in this section, the word "cluster" refers to a logical construct consisting of a block of consecutive words and pauses.

The algorithm begins as follows. Source text is read until the deferral window is filled. Initially each sentence (note: for ASR text, "sentence" refers to a group of words between pauses) is considered to be a cluster. A similarity score (described below) is then computed for all pairs of neighboring clusters. If the most similar pair of neighbors meets a minimum similarity threshold, the

two clusters are combined to form a new cluster, which is then compared to its neighbors. The process repeats until no pair of neighbors meets the similarity threshold, or until all the sentences in the window have been combined into one cluster.

A general step of the algorithm proceeds similarly. The left end of the deferral window is placed at the first inter-cluster gap; this is the earliest potential segment gap. The window is again filled with new words, and the clustering algorithm is performed until no further combinations are possible. If the leftmost cluster in the deferral window was combined with the cluster on its left, that means that the potential gap was in fact just part of a larger segment. If not, then the potential gap was in fact a segment break, and the new segment is declared. This step is then repeated until the end of the file is reached.

We use a max-heap to access the cluster similarity scores. This means that the most similar cluster pair can always be found in logarithmic time. The algorithmic complexity of the clustering method is therefore O(k log k), where k is the number of sentences in the file. In practice we found that the clustering runs could be performed in under an hour on a top-end Linux PC.

A simple combination criterion of similarity was used for the TDT2 runs, depending on the duration of the inter-cluster pause and a lexical similarity score. If the pause duration was smaller than a given time threshold (0.5 seconds in the test runs) the similarity score was set to the maximum, insuring that the clusters would be combined. If the duration was greater than a second threshold (4 seconds), the score was set to the minimum, ensuring that a gap would be declared. These two rules were applied irrespective of lexical similarity. Otherwise, the pause duration was considered to be of no value, and similarity was computed as a dot product of $TF{\cdot}IDF$ weighted cluster representation vectors. Stemming and stop words were not employed. Thus, only three operational parameters controlled the performance of the system. Figure 8.2 shows the combined results for all window sizes. Our false alarm rate is reasonable compared to other systems, but we have a rather high miss rate.

With the framework in place, we have now turned our attention to fine-tuning the algorithm with some natural extensions, of which we mention three. First, the values of the three operational parameters will be optimized. We are currently implementing a stepwise gradient descent method for learning the optimal values within the current combination objective. Second, the criterion will be extended to incorporate more information, such as the number of words in a cluster and the presence of stop words. Finally, the algorithm itself can be made more general by examining more than neighboring pairs of clusters. For instance, it is possible that a cluster could match its neighbor poorly, but match its neighbor's neighbor very well, indicating that all 3 clusters belong to the same story. We will experiment with varying the width of this search.

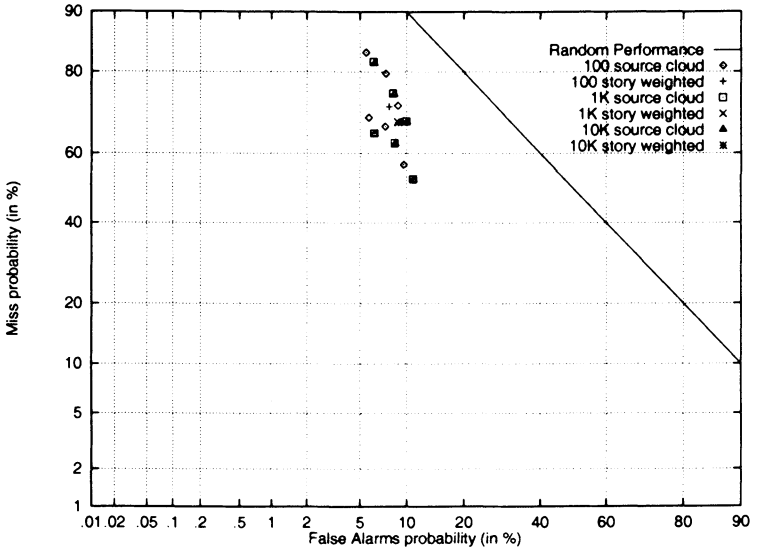


Figure 8.2. TDT 1998 Segmentation Results

3. Detection

This task takes the same general approach as that done for our Web search agent work, where stories (Web pages) are incrementally clustered as they arrive [4, 5]. The stories in each input file are clustered using a specified membership threshold (α). To support experimentation with deferral periods, we then 'pipeline' these cluster sets for the specified period and then base decisions first on whether a given cluster is sufficiently close to a previously declared topic cluster based upon a inter-cluster threshold (currently also α). If it is we merge the new cluster with the declared topic cluster. Otherwise we look forward in the pipeline to see if any future cluster is sufficiently close (same threshold α) as to warrant declaring the current cluster as a new topic cluster. Clusters failing both tests and containing a single story were discarded as noise for our official runs. Non-singleton clusters are declared as a new topic.

A deferral period of 1 is handled as a special case by retaining the 10 most recent file cluster sets for use in the second stage decision making process. This leads to a lag in the identification of new topics, but avoids the discarding of stories with a low appearance frequency. Story vectors and hence cluster vectors are generated after excluding stopwords and stemming the rest. Terms weights are computed using $TF{\cdot}IDF$ scores after normalizing for length of story. Term weights are incrementally updated every ten files.

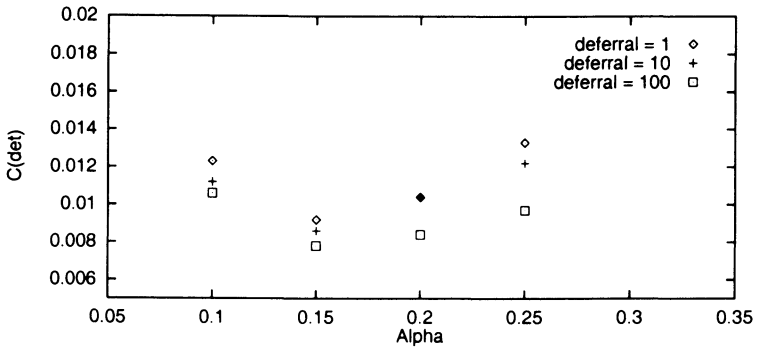


Figure 8.3. TDT 1998 Detection Threshold Tuning Runs

3.1 Developmental Runs

Our initial approach to detection did not include the pipeline concept, focussing instead on tuning α over a range of 0.10 – 0.25 in increments of 0.05 for the specified deferral periods. As shown in Table 8.1 for one month of ASR data, providing a retrospective pipeline for a deferral period of 1 has significant effect upon performance.

Table 8.2 shows the scoring for our development runs. In all ASR cases, α = 0.15 generates the minimal C_{det} value, which we subsequently used for our official runs. The tuning runs from Table 8.2 are more clearly visualized in Figure 8.3, where C_{det} is plotted against α. There is a clear (local) minimum for all deferral periods at $\alpha = 0.15$, with all deferral periods performing roughly the same.

Table 8.1. Pipeline Effects on Detection Performance

03/98 ASR		Story Weighted			Topic Weighted		
Deferral	α	P(Miss)	P(Fa)	C_{det}	P(Miss)	P(Fa)	C_{det}
10	.15	.9502	.0013	.0203	.3664	.0013	.0086
1 w/ retro	.15	.9546	.0012	.0203	.4268	.0012	.0097
1 w/o retro	.15	.9585	.0018	.0209	.5624	.0018	.0130

3.2 Analysis of TDT1998 Developmental Results

Comparing cluster score clouds across runs, we found that raising the threshold does improve our false alarm rate proportionately. Unfortunately, we did not see a corresponding improvement in P(Miss). Rather than the entire cloud shifting down, we are finding that the cloud is instead elongating, with some topics improving well and others hardly at all. Figures 8.4 and 8.5 illustrate this effect for α = 0.10 and 0.25 for the development runs.

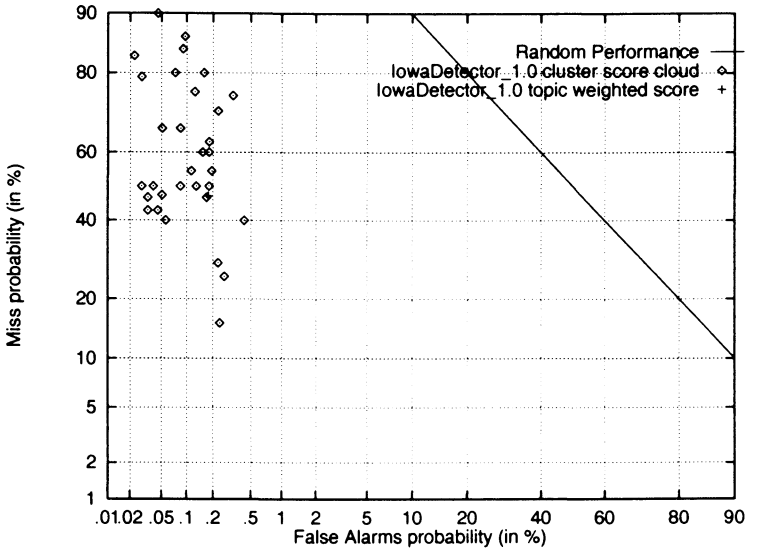


Figure 8.4. TDT 1998 ASR Detection Development Results, deferral = 10, α = .10

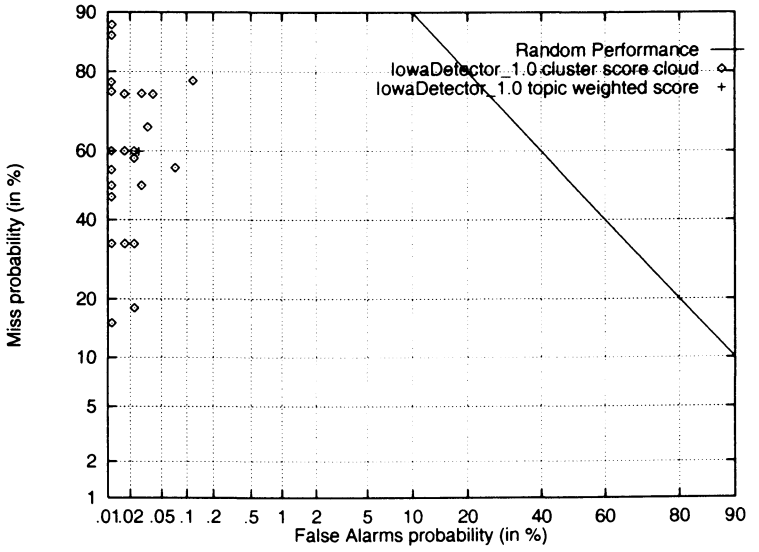


Figure 8.5. TDT1998 ASR Detection Development Results, deferral = 10, α = .25

Table 8.2. Detection Development Results

ASR		Story Weighted			Topic Weighted		
Deferral	α	P(Miss)	P(Fa)	C_{det}	P(Miss)	P(Fa)	C_{det}
1	.10	.9742	.0017	.0212	.5255	.0017	.0123
	.15	.9612	.0009	.0201	.4181	.0008	.0092
	.20	.9678	.0005	.0198	.4974	.0005	.0104
	.25	.9770	.0004	.0199	.6450	.0004	.0133
10	.10	.9735	.0018	.0212	.4706	.0018	.0112
	.15	.9632	.0007	.0200	.3944	.0007	.0086
	.20	.9637	.0005	.0198	.4954	.0005	.0104
	.25	.9730	.0003	.0197	.5987	.0003	.0122
100	.10	.9740	.0018	.0212	.4478	.0017	.0106
	.15	.9590	.0007	.0199	.3528	.0007	.0078
	.20	.9681	.0003	.0197	.4065	.0003	.0084
	.25	.9733	.0002	.0197	.4751	.0002	.0097

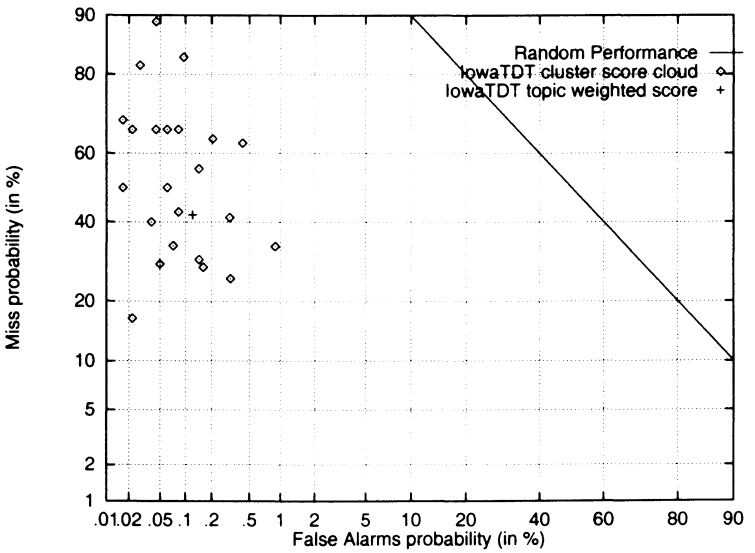


Figure 8.6. TDT 1998 Detection, Deferral = 1, NWT + ASR

3.3 Official TDT1998 Results

Figures 8.6 through 8.10 show our official TDT1998 runs. Cross comparing Figures 8.6 and 8.8 (NWT and ASR) and Figures 8.7 and 8.9 (NWT and CCAP) for deferrals of 1 and 10 respectively shows clear performance improvement for the cleaner sources (NWT and CCAP). Cross comparing Figures 8.6, 8.8 and 8.10 for deferrals of 1, 10 and 100 respectively also shows a clear pattern of improved performance for longer periods of deferral. This is a clear indi-

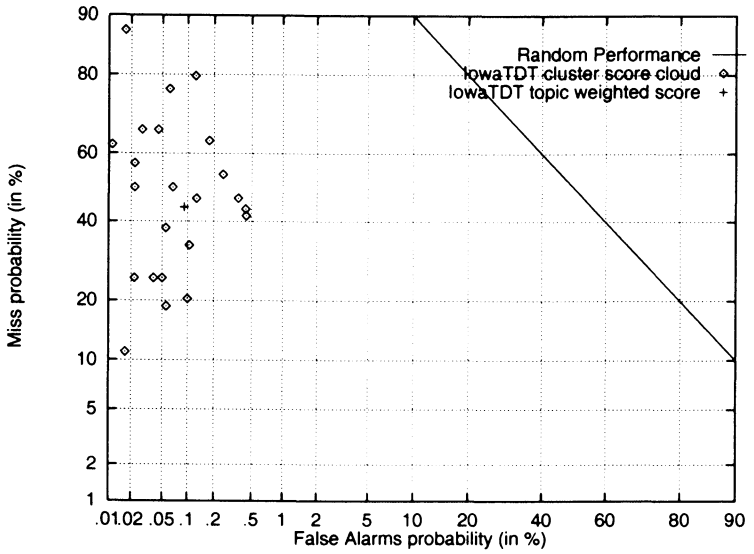


Figure 8.7. TDT 1998 Detection, Deferral = 1, NWT + CCAP

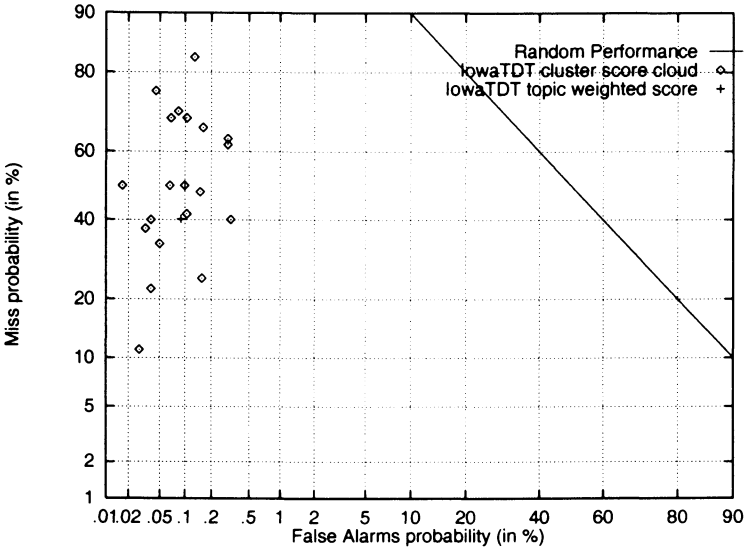


Figure 8.8. TDT 1998 Detection, Deferral = 10, NWT + ASR

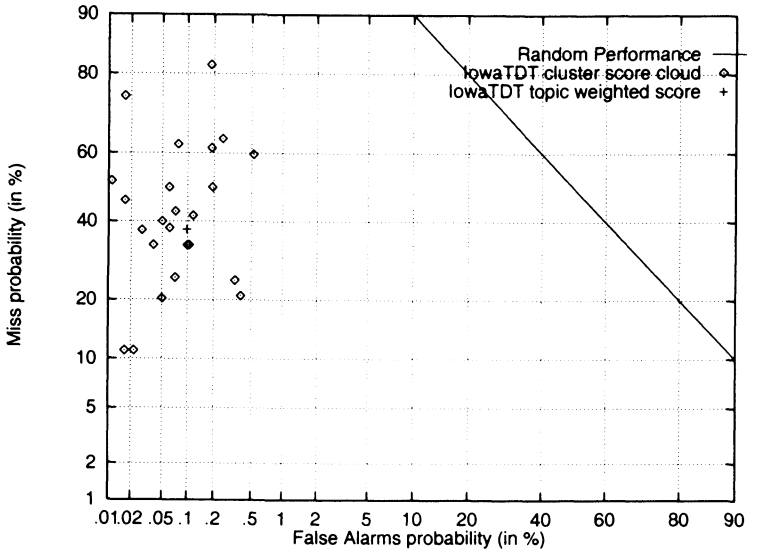


Figure 8.9. TDT 1998 Detection, Deferral = 10, NWT + CCAP

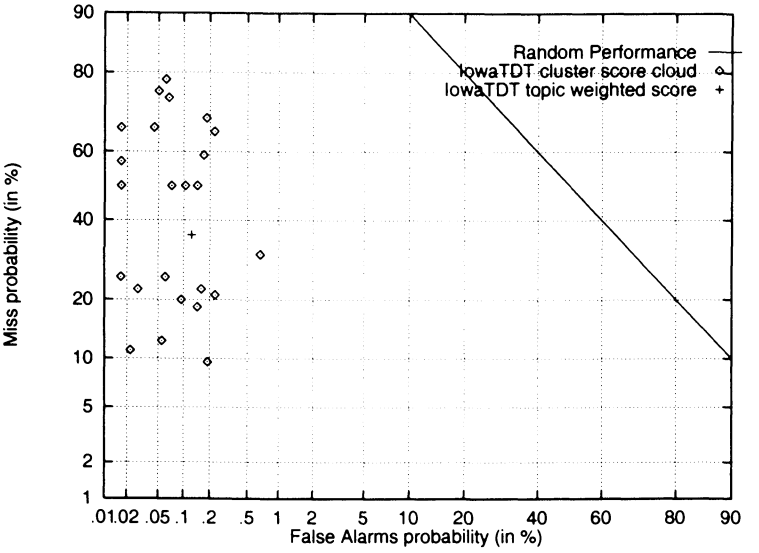


Figure 8.10. TDT 1998 Detection, Deferral = 100, NWT + ASR

cation that our pipeline clustering scheme is successfully grouping temporally proximate stories.

False alarm probabilities for all runs were good. All runs (with the exception of a single topic cluster in the deferral = 100 run) had topic cluster P(FA) values of 0.5% or less, with the topic weighted score somewhere around 0.1%. Our P(miss) performance was somewhere around 40% for all submitted runs. We suspect that this is due in a great extent to the fact that we are generating roughly 2000 – 3000 declared topic clusters during a given run, and that the documents relevant to a given topic are splitting across two or more clusters. The evaluation scheme then chooses only one of these for scoring. We are currently examining intercluster similarities to see if some form of cluster fusion could reduce the number of declared clusters.

3.4 Unofficial TDT1999 Results

TDT1999 marked our first foray into deeper lexical analysis.[2] Our work up to this point had focused solely on document/story representations that were term frequency vectors, where the terms were single words, stemmed using Porter's algorithm [11] and where stop words were removed. Rather than approaching the set of tasks from the perspective of incrementally refining system performance, we decided to take an approach of increasing both the semantics and the dimensionality of a document/story by first tagging the text with parts-of-speech and then selecting various features from the tagged text for distinct representation.

"'We developed an new implementation of Brill's rule-driven tagger [2] that was able to handle a document token stream of arbitrary length and that conformed to our architecture. We separated stories into sentences and applied the tagger to each sentence. The tagger acts as an input stream filter, adding separate vectors for all tagged terms, verb phrases and noun phrases to the 'traditional' stemmed term vector for each story. The observing clusterer process can then select which feature space in which to assess similarity.

Figure 8.11 shows the DET cloud using the stemmed term vector on TDT1999 data. This is basically the same system as used for our TDT1998 runs, and was used for contrastive purposes. Figure 8.12 shows the DET cloud using the part-of-speech tagged term vector. Note that the only clear effect using this feature space is to degrade P(Miss) without seriously damaging P(FA). Figure 8.13 shows the results using the noun phrase vector. Note that while there is a degradation in P(Miss), there is a substantial improvement in P(FA), with a number of

[2]This work was done very late inthe TDT evaluation cycle to the effort expended on supporting Mandarin for the tracking task, so the results reported here are unofficial, since the evaluation runs were done by us, rather than NIST.

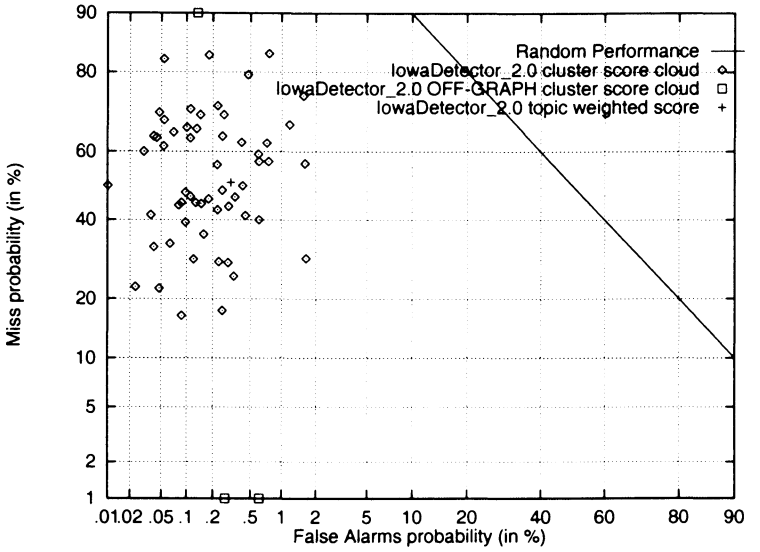


Figure 8.11. TDT 1999 Detection, Simple Term Vector Similarity

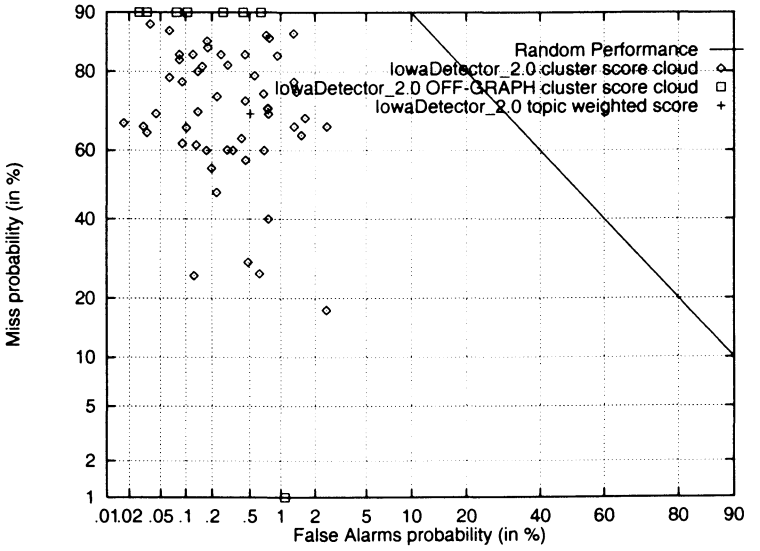


Figure 8.12. TDT 1999 Detection, Part-of-Speech Tagged Vector Similarity

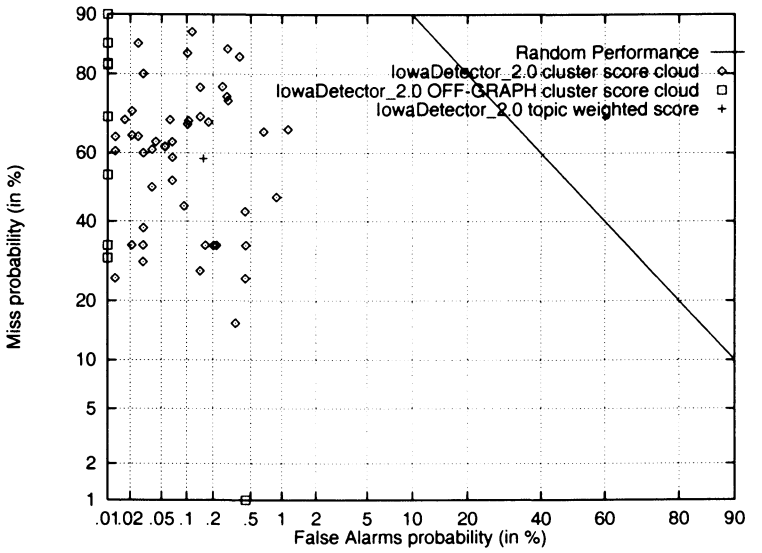


Figure 8.13. TDT 1999 Detection, Noun Phrase Vector Similarity

topic clusters having extremely low P(FA), $\sim 0.1\%$, even when having (relative to other topic clusters in the same run) very good P(Miss) performance, $\sim 30\%$.

We found these results very intriguing in their potential. Unlike the results from TDT1998, where reducing false alarms caused an elongation of the DET cloud in the P(Miss) dimension, the overall coherency of the cloud was still present when using the noun phrase vectors to assess similarity. This formed the basis for much of the tracking work we did for TDT2000, discussed in the next section.

We also did a retrospective analysis of topic cluster activity on the TDT1999 stemmed term vector run. We were primarily curious if there were any discernible patterns in topic cluster activity, defined as either a pipeline cluster forming a new permanent, declared cluster or joining an already declared cluster. Given the number of clusters that we were declaring, we suspected that a large number of clusters exiting the deferral pipeline were declaring, but failing to attract subsequent clusters to merge with them. An examination showed that there is a definite wavefront of activity around the time a cluster first forms, and that there is a drop-off in activity over time, but not a particularly large one. Large numbers of declared clusters continue to attract pipeline clusters, and hence their stories over extended periods of time. Given our low false alarm rates, this seems to imply that fusion of declared clusters might be the only viable approach to improving our miss rates.

4. Tracking

The tracking system begins by generating two sets of clusters from the training data: one set from the on-topic training stories and the other from the off-topic training stories (when provided). A membership similarity threshold (a) controls cluster formation and extension. In order to cut down on the time and space for computations, we introduced a second threshold (β). An off-topic training story must be within β similarity of at least one positive cluster in order to be clustered during this training phase. Any off-topic stories failing this criterion are discarded as insufficiently similar.

During testing, a new story must first qualify for consideration by having a maximal on-topic similarity that exceeds its maximal off-topic similarity. Any story failing this criteria is declared a non-match for the topic with a confidence equal to its maximal on-topic similarity. If the maximal match is on-topic a second level criteria is applied. If the maximal on-topic similarity is above α, the story is declared relevant, at or below α, the story is declared non-relevant, in each case with a confidence equal to its maximal similarity. The clusters generated with the training data remain unchanged throughout the test phase.

4.1 Analysis of TDT1998 Results

Our initial shakedown runs involved only a thresholds in effect at a very low setting (0.1). We then post-processed the result files to toggle yes/no decisions at a variety of α values, with optimal results in the range 0.20 – 0.25. We then tested the following parameter combinations using the development data:

1 $\alpha = 0.25$ and $\beta = 0.20$;

2 $\alpha = 0.20$ and $\beta = 0.15$.

Table 8.3 shows both the official results and the results of a run correcting a declaration error. Figure 8.14 shows the corrected DET curves for ASR, Nt = 4.

Table 8.3. TDT1998 Tracking Results, $\alpha = 0.25$, $\beta = 0.20$

	Story Weighted			Topic Weighted		
	P(Miss)	P(Fa)	C_{det}	P(Miss)	P(Fa)	C_{det}
Official asr, Nt=4	.0821	.0493	.0500	.1460	.0425	.0446
Official man_ccap, Nt=4	.2335	.0018	.0064	.2531	.0018	.0068
Corrected asr, Nt=4	.2476	.0020	.0069	.2639	.0020	.0072

We subsequently completed a full set of tracking runs for the 'boundaries given' case, with results for Nt =1 and Nt = 4 shown in Table 8.4. Comparing the corrected results with the results from the other teams leads us to believe

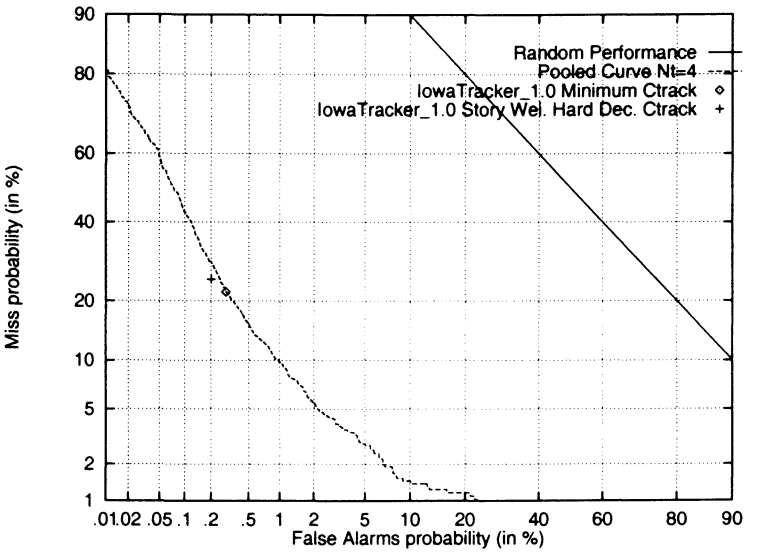


Figure 8.14. TDT1998 Tracking

that even simple algorithmic approaches can perform well when customized for the specific task. Performance across different input file types tends to be rather similar. In general our results, although not the best, are very reasonable for a first round. The DET curves for all the runs are consistently smooth and concave suggesting no sudden changes in expectation required from the user.

As expected, reducing the amount of relevant information used, from 4, to 2 to 1 relevant document worsens C_{track}. However comparison of the DET curves is interesting. For all three types of sources the highest P(Miss) value on the curve drops while the lowest P(Fa) value rises as fewer relevant documents are used. For example, Figure 8.15 shows FDCH runs with 4, 2 and 1 training examples, $\alpha = 0.25$ and $\beta = 0.20$. The highest P(Miss) score on the curve drops from 0.8 to 0.65, while the lowest P(Fa) score rises from 0.08 to 0.15 as one moves from 4 relevant documents to 1.

4.2 TDT1999 – Going Multilingual

The addition of Mandarin as a source language with TDT1999 was our first attempt at non-Roman languages. A great deal of our effort expended for TDT1999 was devoted to implementation and testing of two candidate word segmenters, derived from those done by Peterson [10] and Wu [12]. Lacking an objective framework to evaluate relative performance, we picked the Peterson segmenter for our runs based upon a qualitative assessment of English-Chinese dictionary look-up results.

Table 8.4. TDT1998 Tracking Retrospective Results

Source # Ex.	Run	Story Weighted P(Miss)	P(Fa)	C_{track}	Topic Weighted P(Miss)	P(Fa)	C_{track}
ASR 4	Best (BBN)	.1415	.0035	.0063	.1185	.0033	.0056
	$\alpha = .25, \beta = .20$	.2476	.0020	.0069	.2639	.0020	.0072
	$\alpha = .20, \beta = .15$	.1758	.0040	.0075	.1765	.0037	.0071
ASR 1	Best (UPenn)	.2586	.0017	.0068	.3644	.0021	.0094
	$\alpha = .25, \beta = .20$	.4176	.0010	.0093	.4618	.0009	.0101
	$\alpha = .20, \beta = .15$	.2586	.0021	.0072	.3967	.0020	.0099
CCAP 4	Best (Penn1)	.0725	.0043	.0056	.0904	.0046	.0063
	$\alpha = .25, \beta = .20$	.2474	.0018	.0067	.2637	.0018	.0070
	$\alpha = .20, \beta = .15$	.1618	.0034	.0065	.1844	.0031	.0067
CCAP 1	Best (UPenn)	.2297	.0020	.0065	.2897	.0023	.0080
	$\alpha = .25, \beta = .20$	.3785	.0009	.0084	.4319	.0008	.0094
	$\alpha = .20, \beta = .15$	.2225	.0018	.0062	.3222	.0018	.0082
FDCH 4	Best (CMU1)	.2103	.0032	.0073	.2589	.0021	.0072
	$\alpha = .25, \beta = .20$	.2248	.0018	.0063	.2483	.0018	.0067
	$\alpha = .20, \beta = .15$	.1606	.0034	.0065	.1859	.0031	.0068
FDCH 1	Best (UPenn)	.2296	.0020	.0065	.2898	.0023	.0081
	$\alpha = .25, \beta = .20$	.3759	.0009	.0084	.4296	.0008	.0094
	$\alpha = .20, \beta = .15$	.2226	.0018	.0063	.3231	.0018	.0082

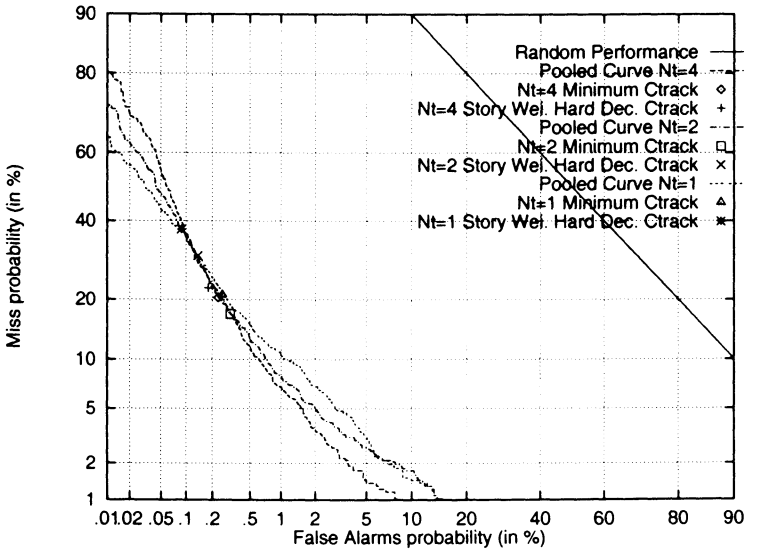


Figure 8.15. TDT1998 Tracking, Effects of Training Set Size

We submitted a single official run over multilingual source and training documents with Nt = 4, with our results shown in Figure 8.16. At first, the curve seems somewhat odd, given our relatively well-behaved curves for monolingual

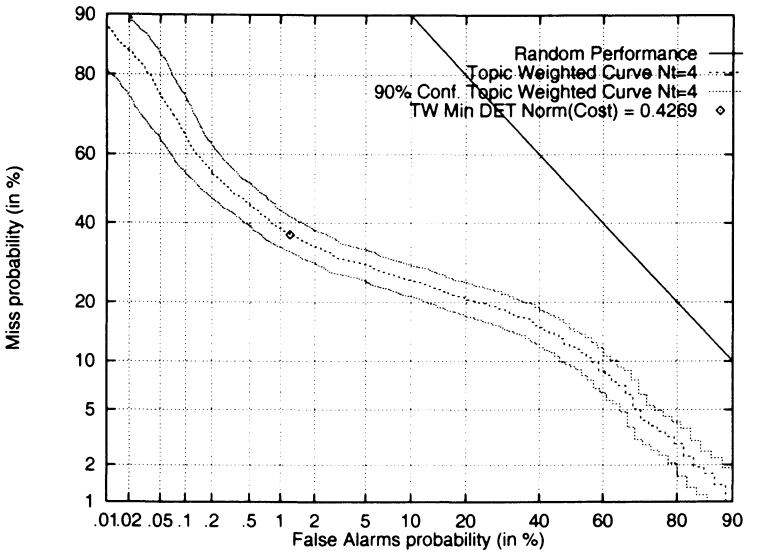


Figure 8.16. TDT1999 Tracking Results, overall

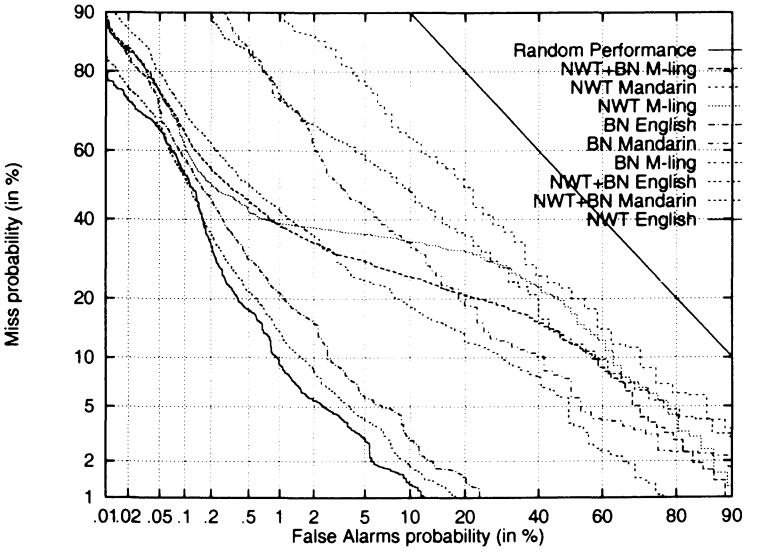


Figure 8.17. TDT1999 Tracking Results, by source and language

data. Plotting performance conditioned on source, as shown in Figure 8.17, yields the cause. The wide separation between the more closely grouped English-only and Mandarin-only curves are blended on the composite curves and this blending apparently occurs in the multilingual version in the ranges of

40% – 60% P(Miss), based upon our confidence scores. This effect is particularly pronounced for NWT data.

4.3 TDT2000 – Entity Tracking

Our work for the TDT2000 evaluation capitalized on the detection work from TDT1999 and work on named entity recognition mechanisms for the question-answering track for TREC-8 [7]. We added an English named entity recognizer of our own design that acts as an input stream filter, consuming the noun phrases generated by the part-of-speech tagger and generating four additional representational dimensions: persons, organizations, locations and events. Each entity type has a set of known entities drawn from external resources and a set of patterns used to match longer strings.

For example, organizations are initialized with a list of international organizations mined from the CIA Fact Book and lists of the Fortune 500 and Global 500 companies. This is then enriched with a set of pattern expressions, e.g.:

- <proper name> "&" "Sons"
- <proper name> "Incorporated"

This category of entity is further supported by various 'glue' words such as "of" being treated as part of an open, non-empty noun phrase in order to improve recognition of entities such as the Organization of American States. Multi-letter words comprised of all capital letters are assumed to be acronyms for organizations, with support for matching already recognized organizations against later occurring acronyms for that organization.

Since TDT topics are defined by exemplar stories, an entity-based tracker will succeed (and fail!) on the nature of the exemplars. Here are two examples and our Ctrack(norm) for each (numbers in parentheses are occurrence counts:

Topic 30001 – Ctrack(norm) = 0.86

- Persons: Hun Sen (1)
- Places: Cambodia (4)

Topic 31029 – Ctrack(norm) = 0.00 (2 Matches, 0 Misses, 0 False Alarms)

- Persons: Geidar Aliev (4)
- Organizations: National Independence Party (2), Baku State University (1)
- Places: capital Baku (1), Baku (1), Azerbaijan (1), Soviet Union (1)
- Events: October (1)

The first has a clear focus, but such a limited number of entries that vector space calculations fail rather ungracefully. The second, with a larger number of entries, succeeds even in the face of a sparse positive document population. These effects are particularly pronounced when each entity category is measured separately. For our official runs, we used the following weights, with the overall similarity calculated as a simple sum:

$$\begin{aligned} sim(t,d) &= 0.3 \cdot sim(persons) + 0.3 \cdot sim(organizations) \\ &+ 0.2 \cdot sim(places) + 0.1 \cdot sim(events) \end{aligned}$$

over the multilingual sources and Nt = 1. Our first interesting result was that we were unable to generate the usual DET curves using the evaluation program due to a number of topics having a $C_{track}(norm)$ of zero. Log scales in plotting packages require non-zero data. Even DET curves for single topics appear odd, due to the fact that the limited vocabulary of entities for both cluster and stories yields a point on the curve where all remaining documents are judged to have zero similarity (they share no entities with the topic). The evaluation program currently plots this area of the curve with slope = 0, a horizontal line.

For purposes of presentation, we will forego log scales on the plots for this section and present data against linear scales. This is of little impact for P(FA), since all scores fall below 0.01, but has significant visual impact for P(Miss) due to the wide range of values. Each plotted data point represents the performance of our system for a single topic. Figures 8.18 and 8.19 show the DET clouds, multilingual and English respectively.

Note the wide disparity in P(Miss) performance, with numerous scores of 1.00 and a smaller number of 0.00 scores. Factoring out the Mandarin (Figure 8.19) further accentuates the P(Miss) spread and further increases the number of topics with P(FA) of 0.00. Note also that while the English outliers in the range of 0.04% – 0.07% P(FA) are broadcast transcription-based topics, there are a substantial number of newswire-based topics in the range of 0.02% – 0.03% P(FA).

Plotting the number of distinct entities recognized for topic stories against the total entities recognized for topic stories for both languages results in Figure 8.20. The large predominance of stories have ten or fewer distinct entities recognized and twenty or fewer total entities.

Plotting $C_{track}(norm)$ vs. distinct (Figure 8.21) & total (Figure 8.22) entities for the multilingual data appears to show that recognizing numerous entities is not in general a good thing.

Indeed, it would appear that a large value for total entities recognized can predict when the system is likely to perform poorly. This turns out to be a spurious conclusion, generated primarily by the Mandarin entity recognizer generating numerous entity entries that are in fact not.

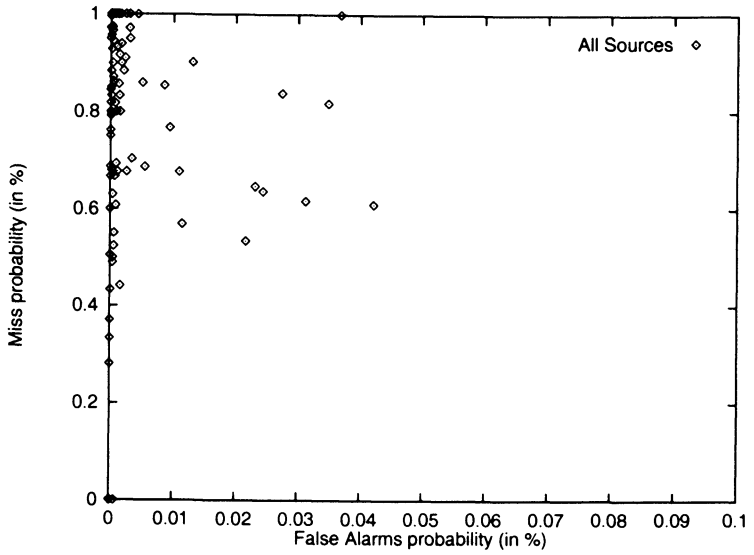


Figure 8.18. TDT2000 Tracking, Multilingual Results

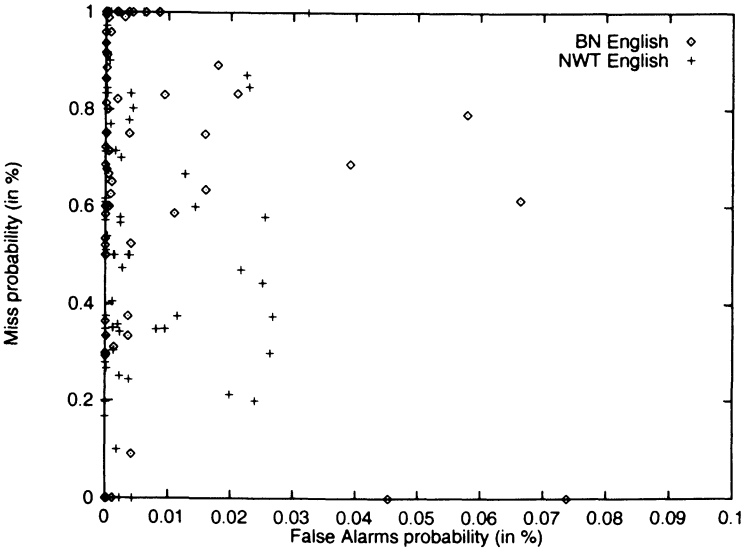


Figure 8.19. TDT2000 Tracking, English Results

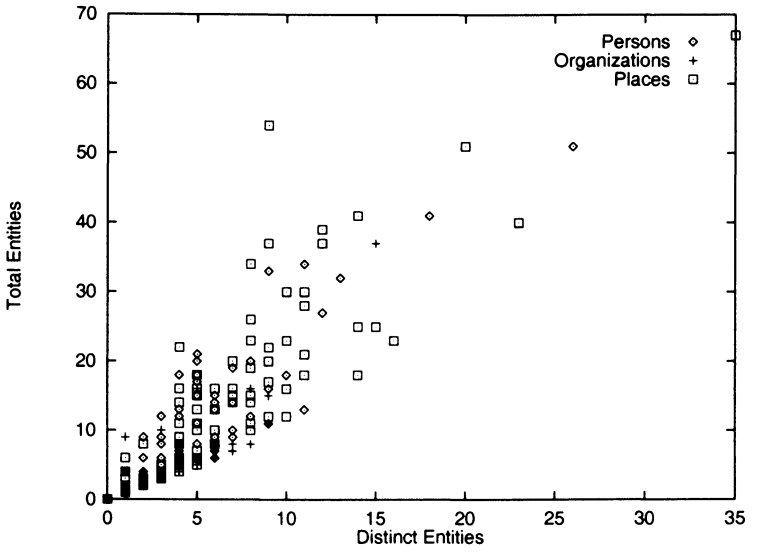


Figure 8.20. TDT2000 Tracking (Multilingual), Distinct vs. Total Entities in Training Story

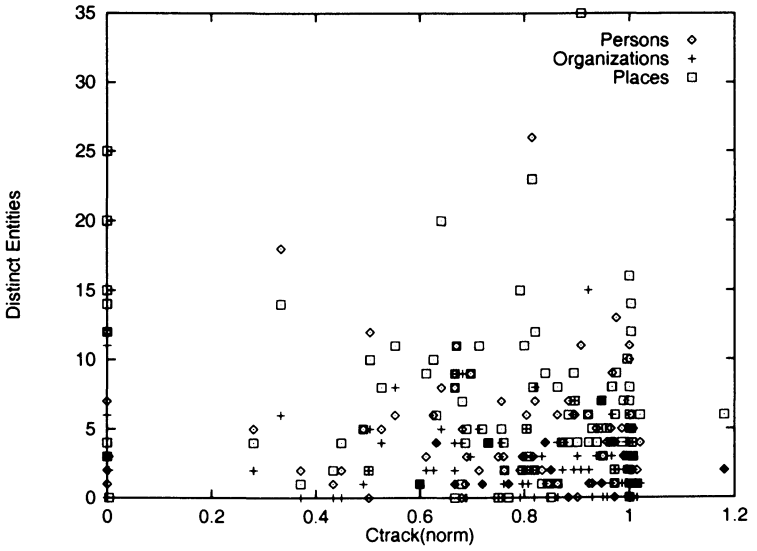


Figure 8.21. TDT2000 Tracking (Multilingual), $C_{track}(norm)$ vs. Distinct Entities in Training Story

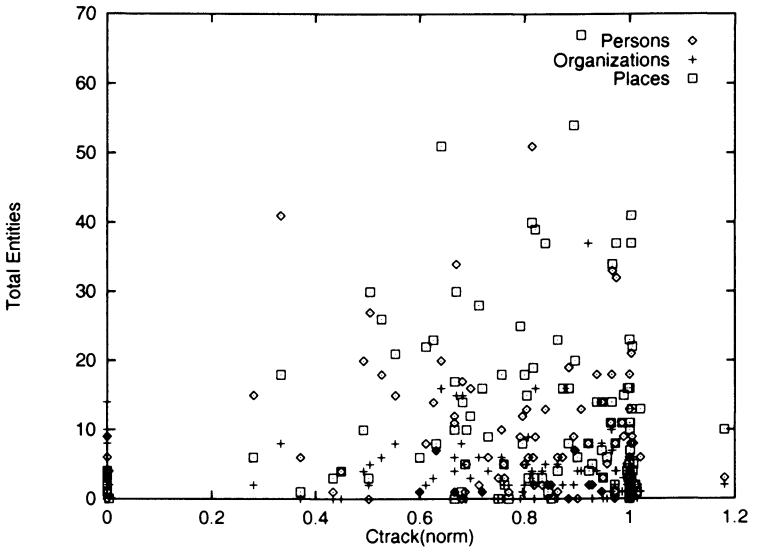


Figure 8.22. TDT2000 Tracking (Multilingual), $C_{track}(norm)$ vs. Total Entities in Training Story

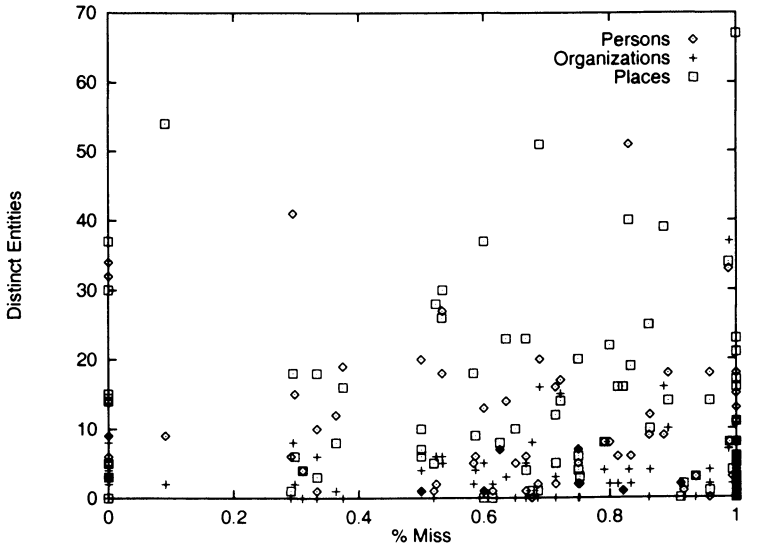


Figure 8.23. TDT2000 Tracking (BN English), % Miss vs. Distinct Entities in Training Story

Considering only the English data yields a very different perspective. Figures 8.23 and 8.24 plot distinct entity counts against P(Miss) for BN and NWT, respectively.

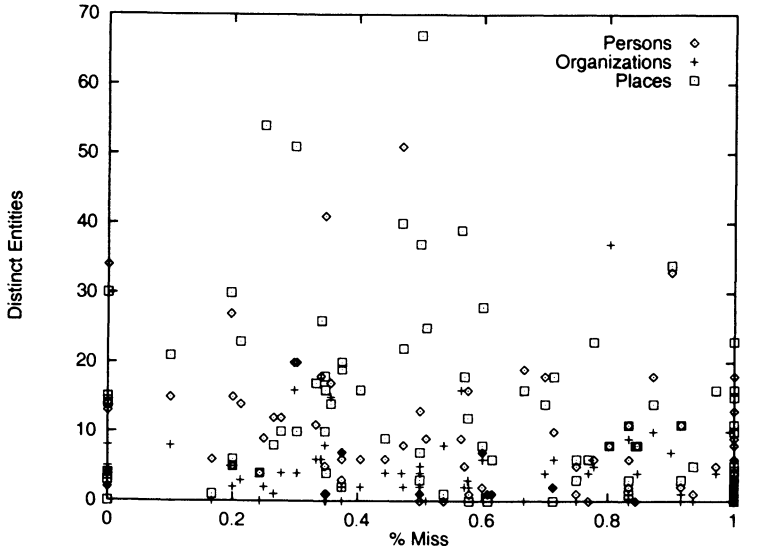


Figure 8.24. TDT2000 Tracking (NWT English), % Miss vs. Distinct Entities in Training Story

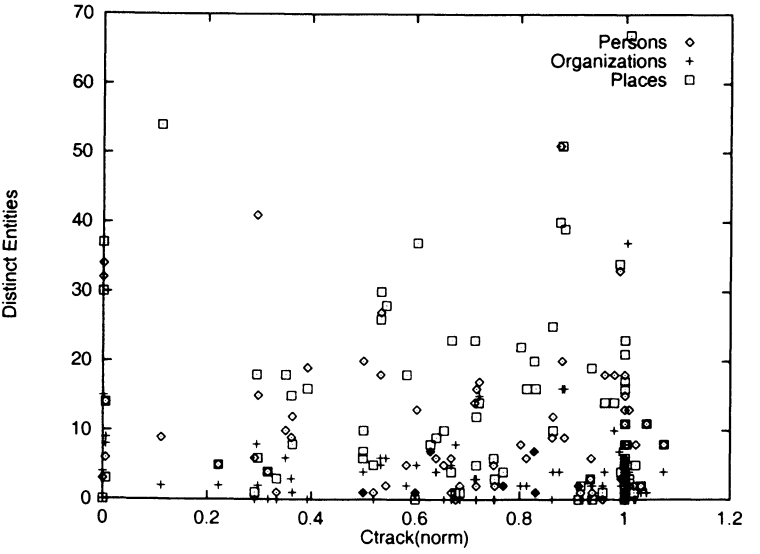


Figure 8.25. TDT2000 Tracking (BN English), $C_{track}(norm)$ vs. Distinct Entities in Training Story

Cleaner data (i.e. NWT) results in a more even spread, in a manner similar to that observed in our TDT1998 detection runs. Shifting from P(Miss) to $C_{track}(norm)$ bears out this observation, as shown in Figures 8.25 and 8.26.

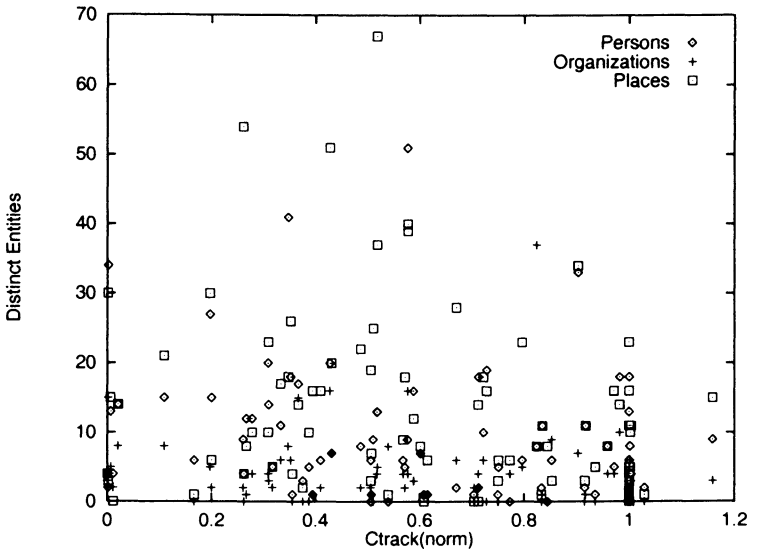


Figure 8.26. TDT2000 Tracking (NWT Eng.), $C_{track}(norm)$ vs. Distinct Entities in Training Story

4.4 Future Directions

There are a number of ways in which we could improve upon these results. For instance we could refine the initial criteria by considering the magnitude of the difference between the new story's similarity with the closest relevant cluster and the closest non-relevant cluster. This is likely to reduce our false alarm rate. Other refinements that modify the training clusters with "high confidence" topic stories are also possible. These are likely to impact both the misses and false alarms.

Clearly, improvements in the entity recognizers will have significant impact. Our current approach operates in a domain-independent manner, with no training required. Extending this to include non-named entities (e.g., physical artifacts such as cars and computers) will alleviate our problems with topics that have no binding to entities per se, such as those relating to economic news or fashion trends.

5. Acknowledgements

Over the course of our work in this area, a number of people have provided valuable support and feedback. Nick Street was instrumental in formulating our approach to segmentation and carried out our TDT1998 experiments in that area. Miguel Ruiz, Fillipo Menczer and Chris Culy participated in our initial concept planning and brainstorming sessions.

References

[1] Allan, J., "Introduction to TDT," in this volume.

[2] Brill, E., "A Simple Rule-Based Part-of-Speech Tagger," *Proc. of the Third Conference on Applied Natural Language Processing,* Trento, Italy, pp. 152-155.

[3] Cowie, J. and W. Lehnert, "Information Extraction," *Communications of the ACM,* v. 39, no. 1, January 1996, pp. 80-91.

[4] Eichmann, D., "Sulla - A User Agent for the Web," poster paper, *Fifth International WWW Conference,* Paris, France, May 6-10, 1996, poster proceedings p. 1-9.

[5] Eichmann, D., "Ontology-Based Information Fusion," *Workshop on Real-Time Intelligent User Interfaces for Decision Support and Information Visualization, 1998 International Conference on Intelligent User Interfaces,* San Francisco, CA, January 6-9, 1998.

[6] Eichmann, D., M. E. Ruiz and P. Srinivasan, "Cluster-Based Filtering for Adaptive and Batch Tasks," *Seventh Conference on Text Retrieval,* NIST, Washington, D.C., November 11 - 13, 1998.

[7] Eichmann, D. and P. Srinivasan, "Filters, Webs and Answers: The University of Iowa TREC-8 Results," *Eighth Conference on Text Retrieval,* NIST, Washington, D.C., November 16 - 19, 1999.

[8] Eichmann, D. and P. Srinivasan, "Adaptive Filtering and Question Answering: The University of Iowa TREC-9 Results," *Ninth Conference on Text Retrieval,* NIST, Washington, D.C., November 13 - 16, 2000

[9] Gamma, E., R. Helm, R. Johnson, J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software,* Addison Wesley, 1995.

[10] Peterson, E., On-line Chinese Tools, http://www.mandarintools.com/.

[11] Porter, M. F., "An Algorithm for Suffix Stripping," *Program,* v. 14, no. 3, 1980, p. 130-137.

[12] Wu, Z. Chinese Segmenter, http://morph.ldc.upenn.edu/Projects/Chinese/.

[13] Wyckoff, P,, S. W. McLaughry, T. J. Lehman and D. A. Font, "T Spaces," *IBM Systems Journal,* v. 37, no. 3, 1998, pp. 454-474.

Chapter 9

Signal Boosting for Translingual Topic Tracking

Document Expansion and n-best Translation

Gina-Anne Levow*
Institute for Advanced Computer Studies,
University of Maryland,
College Park, MD 20742

Douglas W. Oard
College of Information Studies and Institute for Advanced Computer Studies,
University of Maryland,
College Park, MD 20742

Abstract The University of Maryland participated in the TDT-1999 topic tracking task. This chapter describes the system architecture, including source-dependent normalization, and then focuses on the cross-language case in which English training stories were used to find Mandarin stories on the same topic. Processes that may introduce noise, including errorful translation and transcription, are described and five techniques for minimizing the impact of a reduced signal-to-noise ratio are identified. Three techniques focus on signal boosting: augmenting story representations with topically related terminology through "document expansion," exploiting knowledge of alternative translations using balanced n-best term translation, and enriching the bilingual term list to improve translation coverage. The remaining two techniques focus on noise reduction: removing common "stopwords" before translation and using corpus statistics to guide translation selection. Two of the signal boosting strategies yielded substantial gains using techniques that can be ported to other languages fairly easily, while outperforming state-of-the-art general-purpose machine translation. By contrast, neither of the noise reduction strategies produced significant improvements.

*Now at Department of Computer Science, University of Chicago.

1. Introduction

The University of Maryland participated in the Topic Detection and Tracking (TDT) evaluation's topic tracking task. In TDT, a topic is defined as a seminal event and directly related events that follow from it. In the topic tracking task, the goal is as follows: given a number N_t of known on-topic training example stories in English or Mandarin, identify all subsequent stories in English and Mandarin in the incoming stream of newswire text or broadcast news as on- or off-topic. In the required condition there are four example training stories ($N_t = 4$) in English, and tracking is performed on both English and Mandarin stories. Unlike retrospective retrieval, in topic tracking only information from the stories temporally earlier in the stream may be used in making the hard on-topic/off-topic decisions.

The University of Maryland submitted runs for the required condition.[1] As in TDT-1998, our TDT-1999 system was built around the freely available PRISE text retrieval system, using scripts that we will gladly share with other teams [Oard, 1999]. One goal of our work was to provide an easy entry path for new participants by maximizing the use of existing freely available (and supported) resources. In addition to adding the translingual capabilities reported below, we improved our system for TDT-1999 through a better choice of term weighting functions, through more sophisticated selection of query terms, and by tuning a source-specific score normalization strategy using the TDT-2 collection.

The TDT topic tracking task provided a unique opportunity for translingual information retrieval experiments. In translingual information retrieval, the goal is to retrieve relevant documents regardless of natural language (e.g., English or Mandarin Chinese) in which they are written. Prior translingual retrieval evaluations have addressed retrieval of character-coded electronic text among European languages (in the Text Retrieval Conference (TREC) Cross-Language Information Retrieval (CLIR) track) and between English and Japanese (in the NACSIS Test Collection Information Retrieval (NTCIR) evaluation). The TDT-1999 evaluation offered the first translingual evaluation to include:

- Mandarin Chinese,
- automatically transcribed speech,
- exhaustive relevance judgments,
- an event-oriented concept of relevance,
- time-ordered retrieval,

[1] The results presented in this paper were scored *post hoc* using relevance judgments provided by LDC and software provided by NIST. Official TDT-1999 results can be found in [Levow and Oard, 2000].

- a similarly-structured training collection, and
- a common set of baseline language resources.

The principal goal of the work reported here was to exploit this resource to improve our understanding of techniques for translingual information retrieval by evaluating extensions to the dictionary-based translation strategy that we have reported on previously (cf. [Oard et al., 1999]). The topic tracking task afforded an excellent opportunity to compare the effectiveness of our techniques on closely aligned source materials that differ in source type—broadcast news versus newswire text—and language—English and Mandarin Chinese. In the sections that follow we explain the challenges of translingual topic tracking using a signal-to-noise perspective, describe our core system architecture, present experiment results for several contrastive conditions, and suggest some future research directions.

2. The Signal-to-Noise Perspective

Translingual topic tracking in TDT involves several stages of story processing that can introduce errors. Mandarin stories must first undergo automatic segmentation or automatic transcription and then automatic translation. Written Mandarin does not use white space to separate words, so term-based translation of Mandarin newswire stories depends upon automatic segmentation of Mandarin character sequences into terms for which at least one translation is known. Automatic segmentation is imperfect because the optimal granularity for a term (e.g., morpheme, word, or phrase) is sometimes unclear, the semantic knowledge needed to reject implausible segmentations is difficult to represent, and the lexical knowledge encoded in monolingual Mandarin term lists is invariably incomplete. Automatic transcription of speech is also imperfect because acoustically confusable terms may be mistranscribed, unknown words cannot be generated, and the speaking or recording characteristics sometimes fail to match the conditions for which the transcription system was trained. Finally, translation can produce cascading errors that result from inadequate lexical coverage of the source language, a vocabulary mismatch between the translation resource (e.g., translation lexicon or bilingual term list) and the terms that can be generated by the segmenter or transcription system, or incorrect selection among translation alternatives.

Our initial work with Mandarin Chinese suggested that the effect of these cascading errors can be quite severe [Oard and Wang, 1999]. If we view the translated Mandarin stories as containing both signal (terms that help to match the story with our representation of a topic) and noise (spurious terms), then we can view the effect of the cascading errors described above as both reducing the signal (e.g., failure to generate unknown terms) and increasing the noise (e.g., incorrect translation selection). One broad approach to improving translingual

topic tracking performance is thus to improve the signal-to-noise ratio, either by boosting the signal (including more on-topic terms) or by reducing the noise (e.g., by choosing better translations). We have applied several approaches toward this end. To enhance the signal, we improved translation coverage by enriching the baseline bilingual term list that was provided by the Linguistic Data Consortium (LDC) with additional information from twenty general coverage and domain-specific bilingual dictionaries. We also enriched our indexing vocabulary for each document by adding related terms drawn from highly relevant documents in a comparable collection, in the process of document expansion. Finally, we retained multiple translations when more than one candidate was known, balancing the assignment of weights by replicating the same translation when necessary. For noise reduction, we made use of statistical evidence from comparable corpora to exclude very infrequent or misspelled translations and to promote translations that were found often in the comparable collection. We also removed extremely common Mandarin Chinese terms (which typically have many translations) before translation by using a "stopword" list. Finally, one can view state-of-the-art general-coverage machine translation as a careful approach to noise reduction in which the goal is to produce the best *single* translation for each term, so we performed a contrastive run using the Systran Chinese-to-English machine translation system. Since different sources and differential processing both produce differential effects on score assignment, we performed source-dependent score normalization using parameters trained on the TDT-2 collection.

Our experiments demonstrate that a simple focus on noise reduction is insufficient, but that signal boosting can provide substantial improvements in translingual topic tracking effectiveness. Specifically, we found substantial beneficial effects from:

- source-dependent normalization,
- post-translation document expansion, and
- balanced 2-best translation selection.

3. Topic Tracking System Architecture

Our topic tracking system is built around the freely available PRISE information retrieval system from the National Institute of Standards and Technology (NIST) [Dimmick et al., 1998]. PRISE implements a vector space information retrieval paradigm, which we have extended and specialized for the constraints of the TDT topic tracking task through automatic query formulation, offline estimation of collection statistics, and implementation of a source-dependent normalization strategy.

The topic tracking task design requires that all *a priori* statistics be computed from stories prior to the decision point. We implemented that by choosing a set of stories prior to *any* decision point. We used a topic-dependent set of 1,000 stories for this purpose,[2] working backwards from the last known relevant English story, to compute frozen Inverse Document Frequency (IDF) weights. This approach is designed to ensure that both topic-related terminology and a representative "background" vocabulary will be present in the collection from which IDF weights are learned. NIST added a capability to learn frozen IDF weights from a side collection to PRISE to support these experiments.

For query formulation, we constructed a vector of the 180 terms that best distinguish the four known relevant training stories from 996 contemporaneous (and hopefully not relevant) stories. We used a χ^2 test in a manner similar to that used by Schütze et al [Schütze et al., 1995] to select these terms. The χ^2 statistic is symmetric, assigning equal value to terms that help to recognize known relevant stories and those that help to reject the other contemporaneous stories. Because PRISE does not support negation in query formulation, we limited our choice of terms to those that were *positively* associated with the known relevant training stories. We formed the set of 996 contemporaneous stories for each topic by removing the four known relevant stories from the collection used to compute the frozen IDF weights.

We configured PRISE to compute term weights using the Okapi BM25 formula for combining evidence from the frequency of the term in each document, the number of documents in which the term appeared, and the total number of terms in each document. The Okapi BM25 formula is widely used in information retrieval [Robertson et al., 1994], and in a side experiment with the TREC-8 collection we found that it produced much better results than the term weights that we had used for our TDT-1998 experiments.

Source-dependent and topic-dependent normalization. The vector space information retrieval algorithm implemented by PRISE produces a score-ranked list of documents for each query, but those scores are not comparable across queries (because they are not normalized for query length) or across sources (because term usage seems to vary systematically by source). The systematic variation by source that we observed led us to consider source-dependent score normalization, and the topic tracking evaluation metrics (which are based on score rather than rank) required that we include a topic-dependent normalization component as well.

We adopted a two-pass approach to score normalization, first applying a source-specific normalization factor and then using the normalized scores of

[2]The earliest story used to compute collection statistics was never earlier than the first story in the English TDT-3 collection. Sometimes that resulted in fewer than 1,000 stories being used.

the known relevant stories to compute a topic-specific normalization factor. The TDT-3 evaluation collection includes stories drawn from four types of sources: English newswire text, English broadcast news, Mandarin newswire text, and Mandarin broadcast news. In examining the performance of our system on the TDT-2 collection, we observed that the scores assigned to relevant stories by PRISE varied in a manner that depended systematically on their source. Specifically, we found that English stories scored consistently higher than Mandarin stories, that within these categories, text stories scored higher than speech, and that within English text New York Times (NYT) stories scored higher than Associated Press (APW) stories. We therefore computed source-specific multiplicative normalization factors for five source classes (Mandarin speech, Mandarin text, English speech, APW, and NYT) based on the observed scores of relevant stories in the TDT-2 collection. The topic-specific multiplicative normalization factor was then computed by separately computing the source-normalized score for each of the four known relevant stories and taking the average of those scores as the topic normalization factor.

For each topic, we performed a single batch-mode PRISE run in which the score for each story in the evaluation collection was computed as the inner product of the query and document vectors. The resulting scores were automatically normalized for story length by PRISE, which divided the inner product value by the length of the story vector. The appropriate source and topic normalization factors were then applied, and the resulting normalized scores were reported. For contrast, we disabled source normalization and separately examined monolingual English and cross-language (English training stories, Mandarin evaluation stories) results. As Figure 9.1 shows, source-dependent normalization is clearly helpful in both cases.

In this chapter we focus on the contrast between pairs of topic-weighted Detection Error Tradeoff (DET) curves in order to characterize the effect of our techniques [Martin et al., 1997]. When interpreting DET curves, lower curves indicate improved tracking effectiveness. We selected a fairly *ad hoc* score threshold as a basis for the required hard decisions (on-topic/off-topic) after a brief examination of the performance of our system on the TDT-2 collection. The threshold we selected turned out to be far from optimal, so the reported single-value detection cost (C_{det}) values for our runs provides little basis for comparison between conditions.

3.1 Translingual Techniques

We implemented translingual topic tracking by using a dictionary-based translation strategy, consistently translating from Mandarin to English as a preprocessing step. This simplified the design of our system by allowing us to perform all subsequent processing in English, perhaps at some cost in tracking

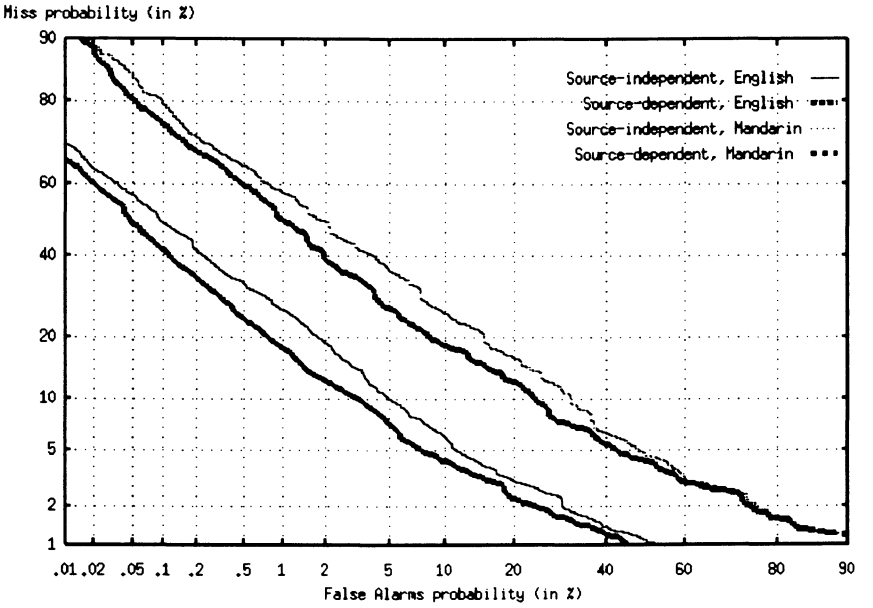


Figure 9.1. Source-dependent normalization produces a beneficial effect. [Combination of broadcast news and newswire text. Source-dependent (thick) vs source-independent (thin) normalization. Upper pair: Mandarin, Lower pair: English.]

effectiveness. In this section, we focus on the cross-language condition in which the training stories are in English and evaluation stories are in Mandarin Chinese in order to characterize the effect of alternative translingual techniques. We first introduce a straightforward topic tracking architecture based on dictionary-based term-by-term translation of each Mandarin story into English, and then describe the effect of augmenting that baseline with signal-boosting and noise reduction techniques.

Term segmentation. Term-by-term translation requires some way of choosing the terms to be translated. In European languages, the white space between words provides a useful cue for this purpose. By contrast, Mandarin words are not normally separated using orthographic delimiters such as white space in written text. We used the New Mexico State University (NMSU) `ch_seg` segmenter to identify individual words in Mandarin newswire text sources.[3] The NMSU segmenter employs a Mandarin term list and a set of rules for recognizing features such as Chinese names, dates and numbers. We based our choice of the NMSU segmenter on two side experiments. In the first experiment, we

[3]The NMSU `ch_seg` segmenter is available at http://crl.nmsu.edu/software.

compared the NMSU segmenter with the segmenter provided by the LDC by using each for query segmentation with Mandarin versions of TREC *ad hoc* queries. In that experiment we found no significant difference between the two segmenters (by the average precision measure) [Oard and Wang, 1999]. In the second experiment, we compared the output of each segmenter with text that was hand-segmented by a native speaker of Mandarin. The NMSU segmenter was assessed by inspection to more closely approximate the hand-segmented text due to better handling of named entities, dates and numbers. For the Mandarin broadcast news source (Voice of America) we used word boundaries provided in the baseline recognizer transcripts as a basis for term selection, so no separate segmentation step was required.

Bilingual term list. We enhanced the second release of the LDC Mandarin-English bilingual term list by automatically extracting translations from twenty dictionaries in the Chinese-English Translation Assistance (CETA) file. The CETA file contains over 230,000 entries compiled from 250 general purpose and domain-specific dictionaries.[4] The twenty dictionaries that we used included contemporary general purpose dictionaries and dictionaries with good coverage of economic and political terminology. Because the CETA dictionaries were originally designed for manual use, they often contain explanatory definitions and examples of usage in addition to the translation-equivalent terms. We extracted translation equivalents from the CETA dictionary using hand-crafted rules, converted both term lists into a uniform format, deleted English entries that were descriptions of function (e.g., "question particle" or "exclamation indicating surprise or disgust") where automatically identifiable as such, and removed all parenthetical clauses. When merging bilingual term lists, we deleted duplicate translation pairs. As Table 9.1 shows, the resulting combined bilingual term list contains 195,078 unique Mandarin terms, with an average of 1.9 English translations per Mandarin term. Remarkably, only 24,448 Mandarin terms (about 27% of the smaller list) were common to both lists. Additional coverage measures for these term lists are described in [Levow and Oard, 1999].

Corpus-based translation selection. Neither the LDC bilingual term list nor the bilingual term list that we extracted from the CETA file contained translation preference information, so we needed some basis on which to select appropriate translation(s) for each term. For our baseline system, we chose the single most likely translation for each term based on corpus statistics. We felt that the only available translation-equivalent parallel texts (Hong Kong laws)

[4]The commercial machine-readable version of the CETA dictionary (also known as "Optilex") is available from the MRM corporation, Kensington, MD.

Term List	Mandarin Terms	English Translations
Combined	195,078	341,187
CETA	91,602	169,067
LDC	127,924	187,130

Table 9.1. Bilingual term list coverage.

might exhibit characteristics very different from those of TDT-3 news stories, so we based our statistics on the observed usage of terms in a more closely comparable English collection. We accomplished this by sorting the English translations in an order that we expected to reflect the dominant usage in the TDT evaluation collection when more than one translation was known for a Mandarin term. Alternate translations were ranked as follows:

1 first, all single-word translations that occurred at least once in a side collection that we selected or in the Brown corpus (a balanced corpus of English), ordered by decreasing frequency in the collection;

2 second, all multi-word translations, in an arbitrary order; and

3 finally, any single-word translations that did not appear at all in the side collection, in an arbitrary order.

This approach was designed to minimize adverse effects from non-standard usage and misspelled translations, both of which are fairly common in our combined bilingual term list. For our side collection, we combined the TDT-2 English newswire text collection with all of the stories from the TDT-3 stories up through the day prior to the story being translated. This combination was designed to provide some degree of robustness for the handling of recently introduced terms, which are fairly common in time-ordered news stories.

Stopword removal. Very common words that would be expected to appear in almost every story are of little value because their presence does not help to distinguish on-topic and off-topic stories. We used the 23-word stopword list distributed with PRISE to remove common English words from the translated documents as an efficiency measure. In our side experiment with TREC query translation we had observed that efforts to translate common Mandarin terms can also be harmful because common Mandarin terms often have an exceptionally large number of possible translations, some of which are rarely used. In order to avoid the risk of selecting an inappropriate translation for a common Mandarin term, we used a Mandarin stopword list to suppress translation of common terms. Since we did not have a list of Mandarin stopwords available, we constructed one by hand. An initial list of candidates was formed by selecting terms from our combined term list with definitions that suggested their use

as function words and then adding the top 300 words from the LDC's Mandarin term frequency list. The resulting list of candidates was then hand-filtered by two speakers of Mandarin.

4. Contrastive Conditions

In this section we compare the results of several contrastive runs with results for the baseline condition described above.

4.1 Document Expansion

We implemented post-translation document expansion for the Mandarin stories in an effort to partially recover terms that may have been mistranscribed (in the case or broadcast news) or missegmented (in the case of newswire text), absent from our bilingual term list, or mistranslated. Singhal et al. used document expansion for monolingual speech retrieval [Singhal and Pereira, 1999], and Ballesteros and Croft applied a similar approach to query translation [Ballesteros and Croft, 1997]. We are not aware of any prior application of the technique to selection of indexing vocabulary for translated documents.

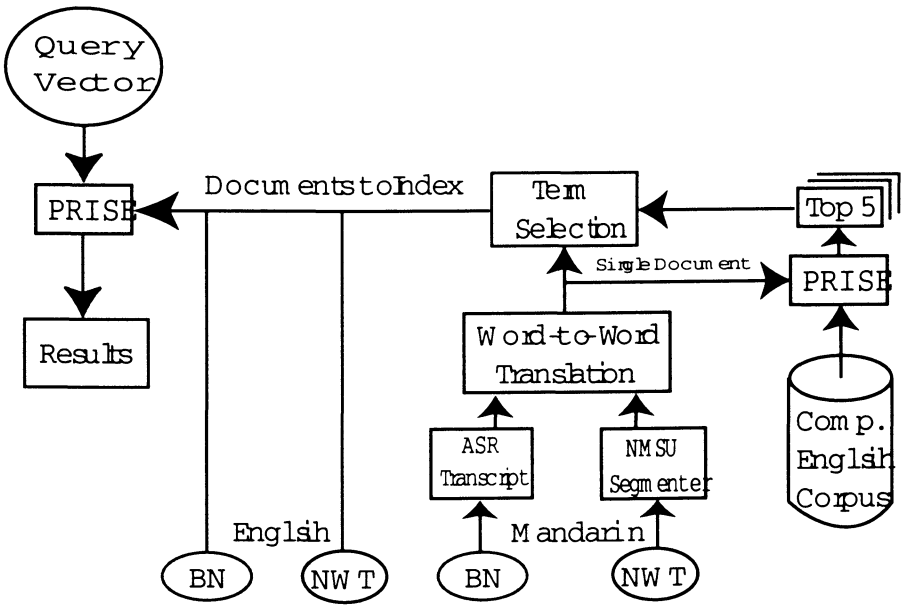


Figure 9.2. The post-translation document expansion process.

Document expansion is a signal boosting technique. Figure 9.2 depicts the document expansion process. Four source classes appear at the bottom of the

figure: English broadcast news (BN), English newswire text (NWT), Mandarin BN, and Mandarin NWT. The English stories were indexed directly—for this contrastive condition we applied document expansion only to the Mandarin stories. Mandarin NWT stories were segmented, and the standard Automatic Speech Recognition (ASR) transcripts were used for the Mandarin BN stories. Term-by-term translation was then used to produce a set of English terms that served as a noisy representation of the Mandarin story. These terms were then treated as a query to a comparable English collection (the English newswire text from the TDT-2 collection), from which PRISE retrieved the five highest ranked documents. From those five documents, we extracted the most selective terms and used them to enrich the original translations of the stories. For this expansion process we selected one instance of every term with an IDF value above an *ad hoc* threshold that was tuned to yield approximately 50 new terms. The resulting augmented translations were then indexed by PRISE, and topic-specific scores were computed in the usual way. As Figure 9.3 shows, document expansion improved topic tracking effectiveness on both Mandarin newswire text and Mandarin broadcast news, with the effect on broadcast news being somewhat larger.

The intuition behind document expansion is that terms that are correctly transcribed or segmented and then correctly translated will tend to be topically coherent, while mistranscription, missegmentation, and mistranslation will introduce spurious terms that lack topical coherence. In other words, although some "noise" terms are randomly introduced, some "signal" terms will survive. The introduction of spurious terms degrades ranked retrieval somewhat, but the adverse effect is limited by the design of ranking algorithms that give high scores to documents that contain many query terms. Because topically related terms are far more likely to appear together in documents than are spurious terms, the correctly transcribed, segmented and translated terms will have a disproportionately large impact on the ranking process. The highest ranked documents are thus likely to be topically related to the correctly transcribed, segmented and translated terms, and to contain additional topically related terms.

These experiments marked our first use of document expansion. Since our expansion parameters (five documents and a fixed IDF threshold) were chosen in an *ad hoc* manner, we felt it important to compare our results with what others have seen under similar conditions. Following Singhal, we applied the same document expansion strategy to the English broadcast news stories in a monolingual condition [Singhal and Pereira, 1999]. As shown in Figure 9.4, we found only a relatively small improvement from document expansion in that case. This suggests that our parameters may not yet be optimally tuned, and that even greater improvements may be possible in the cross-language condition.

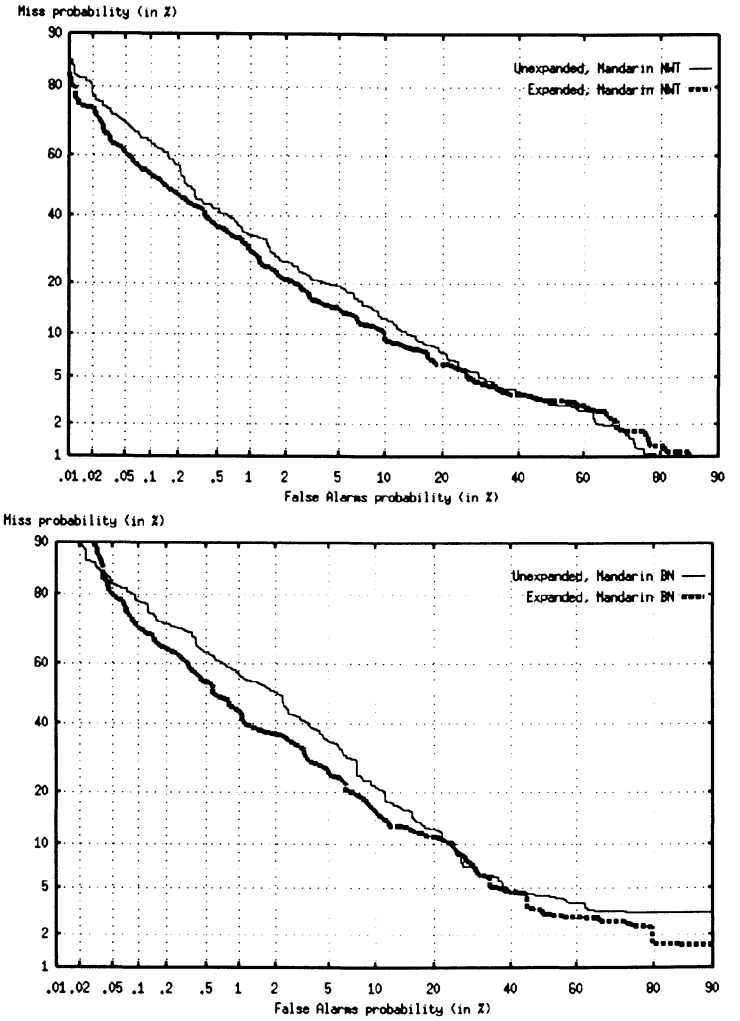


Figure 9.3. Post-translation document expansion produces a beneficial effect. [1-best translation, domain-tuned translation preference, combined term list, Mandarin stopwords removed. Expanded (thick) vs. unexpanded (thin) documents. Top: Mandarin newswire text, bottom: Mandarin broadcast news.]

4.2 Balanced n-best Translation

In prior experiments on portions of the TREC collection we had found that selecting a single English translation is generally better than adding all known translations of each term to the query [Oard and Wang, 1999]. As Leek, et al. have observed, including all known translations has the effect of giving greater weight to terms with more translations [Leek et al., 2000]. But Man-

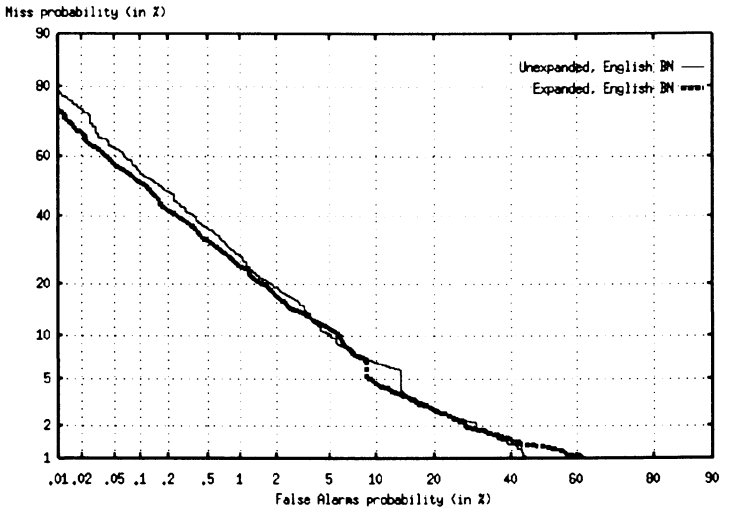


Figure 9.4. Monolingual document expansion produces a beneficial effect. [English broadcast news. Expanded (thick) vs. unexpanded (thin) documents.]

darin terms that have many English translations are almost invariably common terms—terms that a monolingual Mandarin system would suppress by assigning them low IDF values. Motivated by the same insight, we developed an n-best translation strategy in which the contribution from each Mandarin term remains balanced. To maintain this balance in the 2-best case, we duplicated the translation of any term for which only a single translation was known. We treated the 3-best case as follows:

- For terms with a single translation, replace the term with six instances of its translation.
- For terms with exactly two known translations, replace the term with three instances each of the two known translations.
- For terms with three or more known translations, replace the term with two instances each of the three top ranked translations.

We obtained a noticeable improvement from 2-best translation over 1-best translation. As Figure 9.5 shows, the improvement is relatively small for for Mandarin newswire text, but larger improvement is evident for Mandarin broadcast news. This improvement likely confounds two effects: an effect akin to query expansion that results from adding a second translation with similar meaning, or a greater chance of including at least one appropriate translation (when the translations have different meanings). We observed no further improvement from 3-best translation (Figure 9.6). It is interesting to note that our

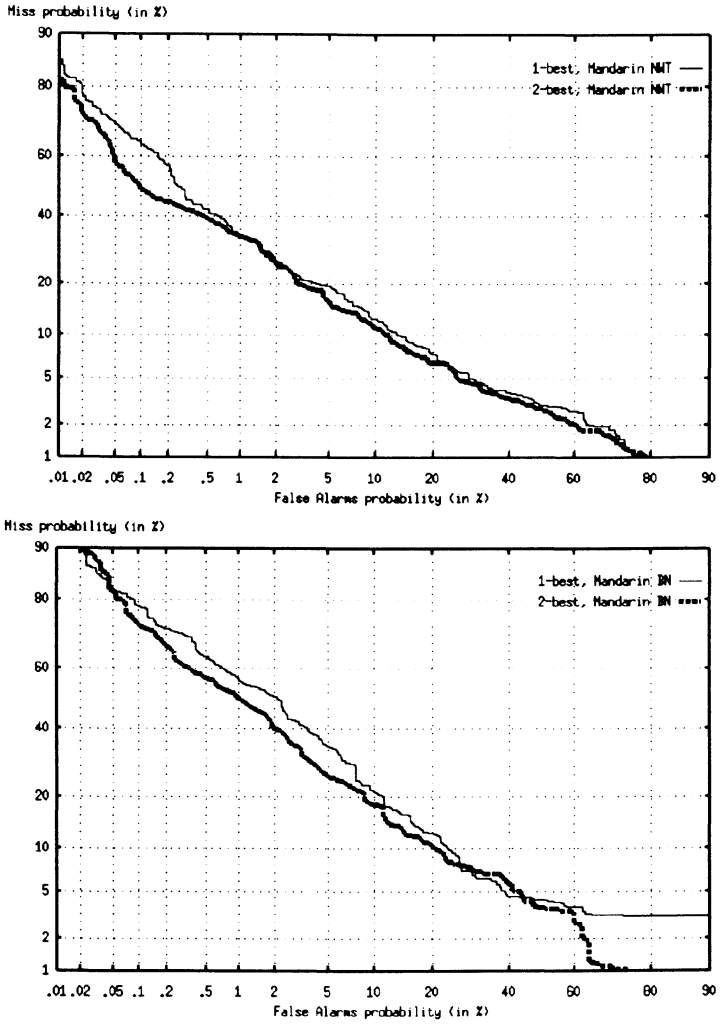


Figure 9.5. n-best translation produces a beneficial effect. [Domain-tuned translation preference, combined term list, Mandarin stopwords removed, no document expansion. 2-best (thick) vs. 1-best (thin) translation. Top: Mandarin newswire text, bottom: Mandarin broadcast news.]

bilingual term list contains an average of 1.9 translations for each Mandarin term—perhaps that value is a good predictor for the number of translations that should be retained when a balanced n-best translation technique is applied.

4.3 Mandarin Stopword Removal

As Figure 9.7 illustrates, we observed no noticeable effect on topic tracking effectiveness from our use of a Mandarin stopword list to suppress translation of

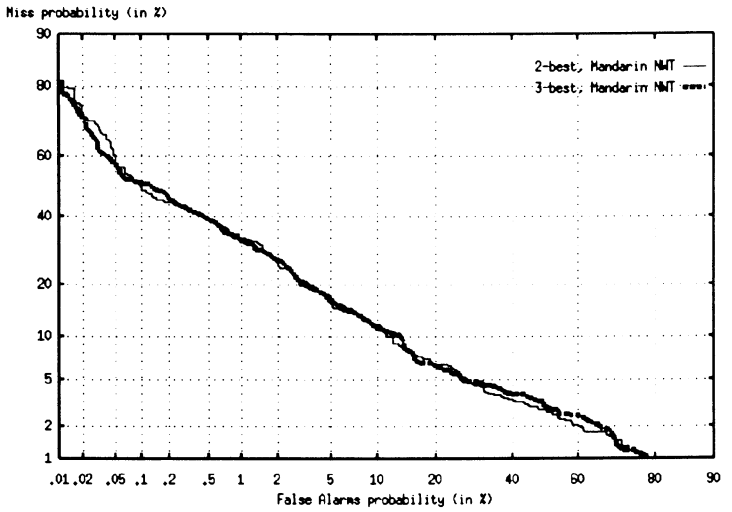


Figure 9.6. No further improvement results from 3-best translation. [Mandarin newswire text, domain-tuned translation preference, combined term list, Mandarin stopwords removed, no document expansion. 3-best (thick) vs. 2-best (thin) translation.]

common terms. Apparently our use of corpus statistics as a basis for translation preference inhibited the selection of uncommon translations for common terms sufficiently well, obviating the need for Mandarin stopword removal. The Mandarin stopword list does, however, avoid some translation effort, and it can reduce the size of the resulting index.

4.4 Translation Preference

In some earlier experiments we had based our translation preference technique solely on the balanced Brown Corpus [Levow and Oard, 1999], so we were interested in characterizing the effect of using a side corpus that was more similar to the stories being translated. As Figure 9.8 illustrates, we observed only a very small beneficial effect from sorting translations based on the domain-tuned statistics of incrementally updated English news over sorting translations based on statistics from the balanced Brown corpus alone.

4.5 Bilingual Term List Enrichment

As Figure 9.9 illustrates, our combined term list performs no better than the LDC term list alone on this task. This suggests that the additional 67,154 Mandarin terms that we added from the twenty CETA dictionaries may not have been well chosen for this task. For example, the CETA file contains 989

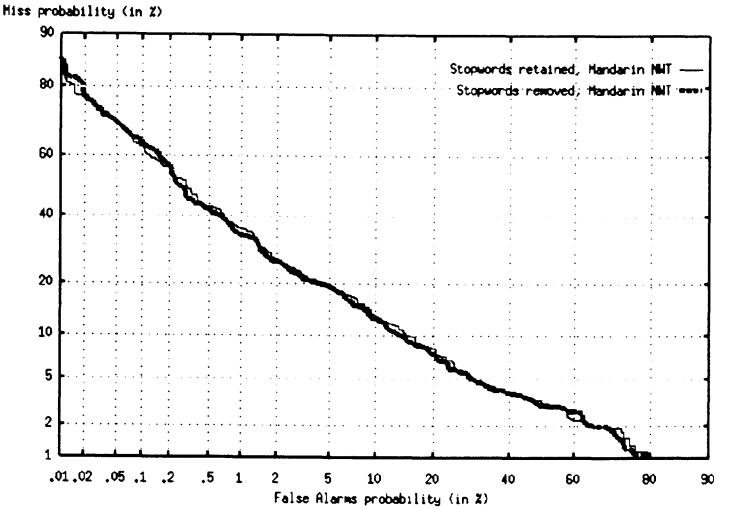


Figure 9.7. Little effect is observed from removal of Mandarin stopwords. [Mandarin newswire text, 1-best translation, domain-tuned translation preference, combined term list, no document expansion. Mandarin stopwords removed (thick) vs. retained (thin).]

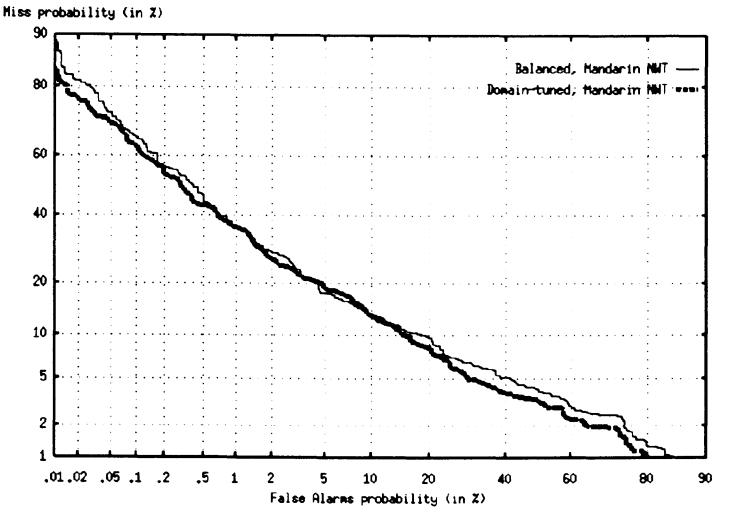


Figure 9.8. A small beneficial effect is observed from domain-tuned translation preference. [Mandarin newswire text, 1-best translation, combined term list, Mandarin stopwords removed, no document expansion. Domain-tuned (thick) vs. balanced (thin) translation preference.]

transliterated foreign names that might have been helpful, but the dictionaries that we selected did not contain those names.

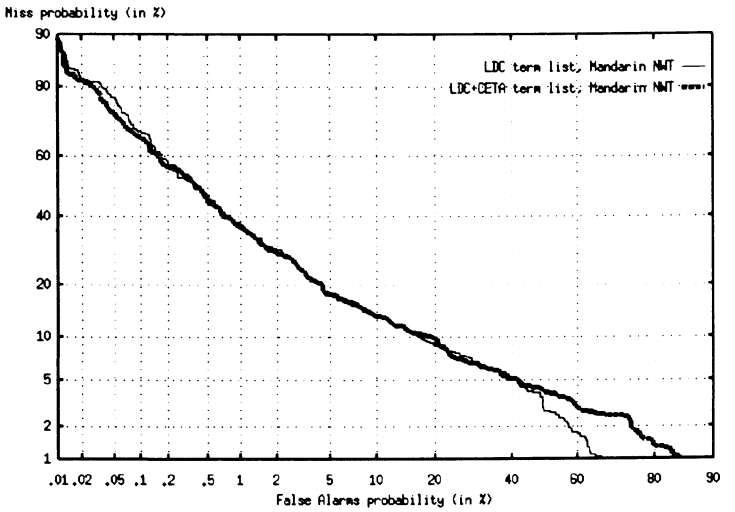


Figure 9.9. Little effect is observed from combining the LDC and CETA term lists. [Mandarin newswire text, 1-best translation, balanced translation preference, Mandarin stopwords removed, no document expansion. Combined (thick) vs. LDC (thin) term list.]

4.6 Comparison with Systran

To provide a baseline for comparison with other participants in the topic tracking task, we performed one run using the standard Systran machine translations that were provided with the TDT-3 collection. We preprocessed the Systran translations by transliterating all remaining Chinese characters (which Systran represents as GB-2312 character codes) into pinyin (with tones), since PRISE is not configured to handle two-byte characters. That approach was originally designed for use when known relevant stories in both English and Mandarin are available, in which case consistent pinyin transliteration could facilitate within-language matching. Since we submitted results only for the English-only training condition, we could equally well have simply removed all instances of GB-2312 characters. As Figure 9.10 shows, our balanced 2-best translation technique outperformed Systran (which produces a carefully tuned 1-best translation). Our (1-best, term-by-term) document expansion results also outperformed the straightforward use of Systran translations, but that is not a fair comparison since document expansion could equally well be used to enhance Systran translations.

5. Conclusions and Future Work

We explored a range of extensions to basic dictionary-based translation techniques for the TDT-1999 topic tracking task—demonstrating two techniques

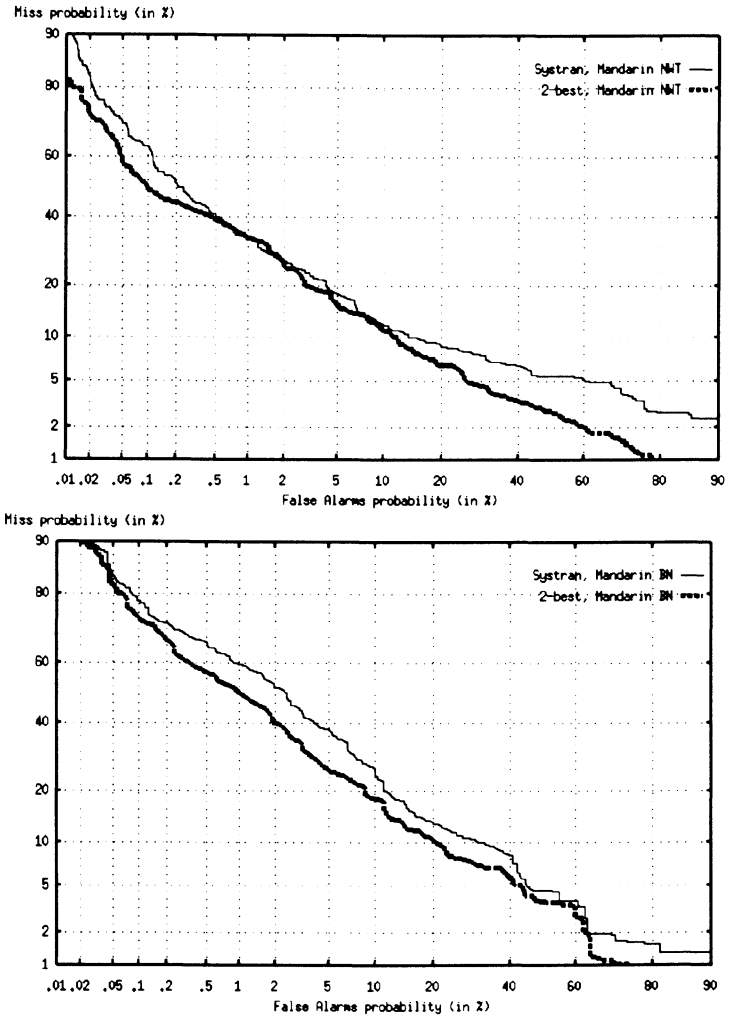


Figure 9.10. n-best translation based on a term list can outperform the straightforward use of Systran. [Thick: 2-best translation, combined term list, domain-tuned translation preference, Mandarin stopwords removed, no document expansion. Thin: Systran. Top: Mandarin newswire text, bottom: Mandarin broadcast news.]

(document expansion and balanced n-best translation) that can improve translingual topic tracking performance. Furthermore, we have shown that using only fairly simple resources it is possible to outperform the straightforward use of state-of-the-art machine translation. Working with Mandarin initially proved to be challenging because segmentation errors can have a cascading effect that results in inappropriate term weights, but we have successfully mitigated that problem by guiding translation selection using statistics from a side collection.

Similar challenges are present to some degree in any translingual information retrieval task, however. For example, the problem of identifying the correct term granularity for translation and indexing arises with English phrases and German compounds. So the results we have obtained should be broadly applicable.

There are two limitations to our results that will need to be addressed in future work. The first is that our present architecture - in particular the use of PRISE as an off-the-shelf component - limits the richness with which we can represent what we know about the likelihood of selecting a particular translation. Vector space systems are capable of capturing translation probability in a natural way (cf., [Oard, 1997]), but implementing such a closely coupled approach in PRISE would require some recoding. The second major limitation is that our results were obtained using a single topic tracking system. We expect that what we have learned will transfer well to any dictionary-based translingual topic tracking system, but firm conclusions in that regard cannot be drawn until these techniques are integrated with systems that achieved the best monolingual topic tracking performance.

The TDT-3 collection provides a remarkably rich basis for exploring translingual information access techniques, and our initial use of that collection has proved to be quite fruitful. Perhaps the most important immediate direction for future work is refining our implementation of document expansion. An obvious first step is to explore the parameter space, varying the number of top documents used and the way in which enrichment terms are selected from those documents. Thinking more broadly, Ballesteros and Croft found that a combination of pre-translation and post-translation query expansion performed better than either technique alone [Ballesteros and Croft, 1997], and we believe that this combination could be a productive approach to explore with document translation as well. Of course, implementing pre-translation expansion will require that we search a comparable Chinese collection. Once we have configured a retrieval system to do that, we will also gain the ability to perform parallel retrieval in English and Chinese. In cross-language information retrieval experiments between French and English, McCarley has found that merged results can outperform the use of either query-language matching or document-language matching in isolation [McCarley, 1999]. The close relationship between information retrieval techniques and the techniques presently being applied to topic tracking leads us to believe that a similar effect might be possible in topic tracking as well.

By creating the first Mandarin/English evaluation collection, the Topic Detection and Tracking evaluation has added an important new dimension to research on translingual information access. In the twelve months following the TDT-1999 workshop, three major evaluation efforts (the TREC-9 CLIR track, NTCIR-2, and TDT-2000) have chosen the same language pair. The relatively modest investment to add Mandarin to the TDT-2 and TDT-3 collections will

thus be very highly leveraged. The research results, the resources that have been assembled, and the test collections that are being created will likely facilitate innovative work in this area for years to come.

6. Acknowledgments

The authors are grateful to Ruth Sperer, Clara Cabezas and Hu Yali for their assistance with the experiments, to Darrin Dimmick and Will Rogers of NIST for making the needed modifications to PRISE, and to Philip Resnik and the reviewers for their helpful feedback on an earlier draft of this chapter. This work has been supported in part by DARPA contract N6600197C8540.

References

[Ballesteros and Croft, 1997] Ballesteros, L. and Croft, W. B. (1997). Phrasal translation and query expansion techniques for cross-language information retrieval. In *Proceedings of the 20th International ACM SIGIR Conference on Research and Development in Information Retrieval.*

[Dimmick et al., 1998] Dimmick, D., O'Brien, G., Over, P., and Rodgers, W. (1998). Guide to Z39.50/PRISE 2.0: Its installation, use, & modification. http://www.itl.nist.gov/iaui/894.02/.

[Leek et al., 2000] Leek, T., Jin, H., Sista, S., and Schwartz, R. (2000). The BBN crosslingual topic detection and tracking system. In *Working Notes of the Third Topic Detection and Tracking Workshop.* http://www.itl.nist.gov/iaui/894.01/tests/tdt/tdt99/papers/index.htm.

[Levow and Oard, 1999] Levow, G.-A. and Oard, D. W. (1999). Evaluating lexicon coverage for cross-language information retrieval. In *Workshop on Multilingual Information Processing and Asian Language Processing*, pages 69–74. http://www.umiacs.umd.edu/~gina/cv/.

[Levow and Oard, 2000] Levow, G.-A. and Oard, D. W. (2000). Translingual topic tracking with PRISE. In *Working Notes of the Third Topic Detection and Tracking Workshop.* http://www.glue.umd.edu/~oard/research.html.

[Martin et al., 1997] Martin, A., Doddington, G., Kamm, T., Ordowski, M., and Przybocki, M. (1997). The DET curve in assessment of detection task performance. In *Proceedings of the 5th European Conference on Speech Communication and Technology*, volume 4, pages 1895–1898.

[McCarley, 1999] McCarley, J. S. (1999). Should we translate the documents or the queries in cross-language information retrieval. In *37th Annual Meeting of the Association for Computational Linguistics*, pages 208–214.

[Oard, 1997] Oard, D. W. (1997). Adaptive filtering of multilingual document streams. In *Fifth RIAO Conference on Computer Assisted Information Searching on the Internet*. http://www.glue.umd.edu/~oard/research.html.

[Oard, 1999] Oard, D. W. (1999). Topic tracking with the PRISE information retrieval system. In *Proceedings of the DARPA Broadcast News Workshop*, pages 209–211. http://www.glue.umd.edu/~oard/research.html.

[Oard and Wang, 1999] Oard, D. W. and Wang, J. (1999). Effects of term segmentation on Chinese/English cross-language information retrieval. In *Proceedings of the Symposium on String Processing and Information Retrieval*. http://www.glue.umd.edu/~oard/research.html.

[Oard et al., 1999] Oard, D. W., Wang, J., Lin, D., and Soboroff, I. (1999). TREC-8 experiments at Maryland: CLIR, QA, and routing. In *The Eighth Text Retrieval Conference (TREC-8)*. http://trec.nist.gov.

[Robertson et al., 1994] Robertson, S. E., Walker, S., Jones, S., Hancock-Beaulieu, M. M., and Gatford, M. (1994). Okapi at TREC-3. In Harman, D. K., editor, *Overview of the Third Text REtrieval Conference (TREC-3)*, pages 109–126. U.S. National Institute of Standards and Technology. http://trec.nist.gov.

[Schütze et al., 1995] Schütze, H., Hull, D. A., and Pedersen, J. O. (1995). A comparison of classifiers and document representations for the routing problem. In Fox, E. A., Ingwersen, P., and Fidel, R., editors, *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 229–237. ftp://parcftp.xerox.com/pub/qca/schuetze.html.

[Singhal and Pereira, 1999] Singhal, A. and Pereira, F. (1999). Document expansion for speech retrieval. In *Proceedings of the 22nd International Conference on Research and Development in Information Retrieval*, pages 34–41.

Chapter 10

Explorations Within Topic Tracking and Detection

James Allan, Victor Lavrenko, and Russell Swan*
Center for Intelligent Information Retrieval
Department of Computer Science
University of Massachusetts
Amherst, MA 01003

Abstract This chapter presents the system used by the Center for Intelligent Information Retrieval (CIIR) at the University of Massachusetts for its participation in four of the five TDT tasks: tracking, detection, first story detection, and story link detection. For each task, we discuss the parameter setting approach that we used and the results of our system on the test data.

For the task of link detection, we look more carefully at score normalization across different languages and media types. We find that we can improve results noticeably though not substantially by normalizing scores differently depending upon the source language. We also consider smoothing the vocabulary in stories using a "query expansion" technique from Information Retrieval to add additional words from the corpus to each story. This results in substantial improvements.

In addition, we use TDT evaluation approaches to show that the tracking performance that sites are achieving is what is expected from Information Retrieval technology. We further show that any first story detection system based on a tracking approach is unlikely to be sufficiently accurate for most purposes. Finally, we present an overview of an automatic timeline generation system that we developed using TDT data.

1. Introduction

This chapter describes the Center for Intelligent Information Retrieval's (CIIR's) approaches to four of the five TDT tasks. We provide a brief overview of each task further in this section. The remainder of the chapter is structured as follows. Section 2 describes the general architecture of our system, and high-

* Russell Swan was the primary investigator and author for the automatic timeline construction work discussed in this chapter. He passed away unexpectedly after completing the work but before this chapter was published.

lights the variations in detection algorithms, similarity functions, and weighting schemes. Sections 3, 4, 5, and 6 report the performance of our system on each of the tasks below. Section 7 provides a theoretical analysis of detection errors and suggests upper and lower bounds on performance. Section 8 describes a novel approach for generating topical timelines. We summarize our contributions in Section 9.

1.1 Task Descriptions

Tracking and Filtering. In a TDT tracking task, we are given a small number, N_t, of stories that discuss a particular topic, and asked to find all the other stories that discuss the same topic. The TREC filtering task is similar, except that in filtering (i) a short topic description is available in the form of a query, and (ii) the system may receive additional feedback beyond N_t training stories. We highlight the similarities and differences between TDT tracking and TREC filtering in Section 7. A detailed analysis of our tracking performance is given in Section 3.

Cluster Detection. The task is to organize a stream of stories into clusters, where each cluster contains stories that discuss a single topic. Cluster detection is similar to tracking, except no example stories are given. Clusters have to be generated "on the fly", with limited lookahead. Experiments with cluster detection are detailed in Section 4.

First Story Detection. The objective is to identify the very first story that mentions any topic in the news stream. It is similar to Cluster Detection, except we only need to report the first story of each cluster. Section 5 describes our performance on this task, and Section 7 postulates that it will be very difficult to improve the performance with current approaches.

Link Detection. In this task we are given two random stories and asked to determine if they discuss the same topic. Link Detection is a fundamental task that underlies all other TDT tasks: given a perfect Link Detection system, it is easy to construct a perfect tracking or clustering system. Our approach to the Link Detection task, along with two novel directions of research, are described in Section 6.

2. Basic System

The core of our TDT system uses a vector model for representing stories—i.e., we represent each story as a vector in term-space, where coordinates represent the frequency of a particular term in a story. Terms (or features) of each vector are single words, reduced to their root form by a dictionary-based stem-

mer. This system is based on one that was originally developed for the 1999 summer workshop at Johns Hopkins University's Center for Language and Speech Processing.[Allan et al., 1999] It was substantially reworked to provide improved support for "language model" approaches to the TDT tasks, though that functionality was not deployed extensively in the TDT 2000 evaluation.

2.1 Detection algorithms

Our system supports two models of comparing a story to previously seen material: centroid (agglomerative clustering) and nearest neighbor comparison.

Centroid. In this approach, we group the arriving documents into clusters. The clusters represent topics that were discussed in the news stream in the past. Each cluster is represented by a *centroid*, which is an average of the vector representatives of the stories in that cluster.

Incoming stories are compared to the centroid of every cluster, and the closest cluster is selected. If the similarity of the story to the closest cluster exceeds a threshold, θ_{match}, we declare the story "on-topic" for the cluster; if the similarity exceeds a second threshold, $\theta_{certain}$, we add the new story to the topic and adjust the cluster centroid. If the similarity does not exceed θ_{match}, we declare the story new, and create a new singleton cluster with the story as its centroid. Both thresholds are set globally and apply to all clusters.

k-nearest neighbor. The second approach, k-NN, does not attempt to explicitly model a notion of a topic, but instead declares a story to be on the topic of the existing story most similar to it. That is, incoming stories are directly compared to all the stories we have seen before. The most similar k neighbors are found, and if the story's similarity to the neighbors exceeds a threshold, the story is declared to be on the same topic. Otherwise, if the story does not exceed that similarity with any existing story, the incoming story is declared the start of a new topic. In this work, we focused primarily on $k = 1$.

2.2 Similarity functions

One important issue in our approach is the problem of determining the right similarity function. We considered four functions: cosine, weighted sum, language models, and Kullback-Leibler divergence. The critical property of the similarity function is its ability to separate stories that discuss the same topic from stories that discuss different topics.

Cosine. The cosine similarity is a classic measure used in Information Retrieval, and is consistent with a vector-space representation of stories. The measure is simply an inner product of two vectors, where each vector is nor-

malized to unit length. It represents the cosine of the angle between the two vectors $\vec{d}$ and $\vec{q}$.

$$\left(\sum q_i d_i\right) / \sqrt{\left(\sum q_i^2\right)\left(\sum d_i^2\right)}$$

(Note that if $\vec{q}$ and $\vec{d}$ have unit length, the denominator is 1.0 and the angle is calculated by a simple dot product.) Cosine similarity tends to perform best at full dimensionality, as in the case of comparing two long stories. Performance degrades as one of the vectors becomes shorter. Because of the built-in length normalization, cosine similarity is less dependent on specific term weighting, and performs reasonably well when raw word counts are presented as weights.

Weighted sum. The weighted sum is an operator used in the *InQuery* retrieval engine developed at the Center for Intelligent Information Retrieval (CIIR) at the University of Massachusetts. *InQuery* is a Bayesian inference engine with transition matrices restricted to constant-space deterministic operators (e.g., AND, OR, SUM). Weighted sum represents a linear combination of evidence with weights representing confidences associated with various pieces of evidence:

$$\left(\sum q_i d_i\right) / \left(\sum q_i\right)$$

where q represents the *query* vector and d represents the *document* vector. For instance, in the centroid model, cluster centroids represent *query* vectors which are compared against incoming *document* vectors.

Weighted sum tends to perform best at lower dimensionality of the query vector q. In fact, it was devised specifically to provide an advantage with short user requests typical in IR. The performance degrades slightly as q grows. In addition, weighted sum performs considerably better when combined with traditional tf·idf weighting (discussed below).

Language model. Language models furnish a probabilistic approach to computing similarity between a document and a topic (as in centroid clustering) or two documents (nearest neighbor). In this approach, previously seen documents (or clusters) represent models of word usage, and we estimate which model M (if any) is the most likely source that could have generated the newly arrived document D. Specifically, we are estimating $P(D|M)/P(D)$, where $P(D)$ is estimated using the background model $P(D|GE)$ corresponding to word usage in General English.

By making an assumption of term independence (unigram model), we can rewrite $P(D|M) = \prod_i P(d_i|M)$, where d_i represent individual tokens in D. We start with a maximum likelihood estimator for $P(d_i|M)$, which is simply

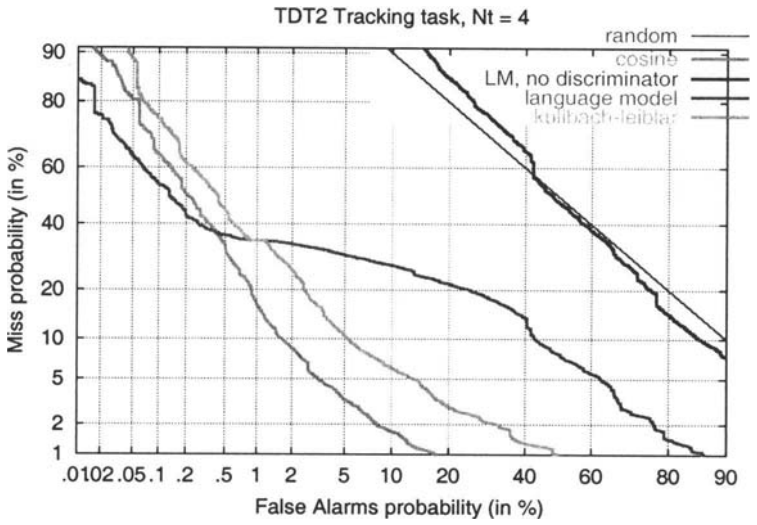


Figure 10.1. DET plot of performance on the tracking task for different similarity functions. "Cosine" intersects the x-axis at about 20, "Kullback-Leibler" at about 50, "language model" is the oddly shaped line, "LM no discriminator" hovers around "random," which is the thin line passing through 50/50.

the number of occurrences of d_i in M divided by the total number of tokens in M. Since our models may be sparse, some words in a given document D may have zero probability under any given model M, resulting in $P(D|M) = 0$. To alleviate this problem we use a smoothed estimate $P(d_i|M) = \lambda P_{ml}(d_i|M) + (1-\lambda)P(d_i|GE)$, which allocates a non-zero probability mass to the terms that do not occur in M. We set λ to the Witten-Bell[Witten and Bell, 1991] estimate $\lambda = N/(N+U)$ where N is the total number of tokens in the model and U is the number of unique tokens. (Note that since detection tasks are online tasks, we may encounter words not in GE, and so we smooth GE in a similar fashion using a uniform model for the unseen words.)

Kullback-Leibler divergence. Instead of treating a document D as a sample that came from one of the models, we could view D as a distribution as well, and compute an information-theoretic measure of divergence between two distributions. One measure we have experimented with is the Kullback-Leibler divergence, $KL(D,M) = -\sum_i d_i \log(m_i/d_i)$, where d_i and m_i represent relative frequencies of word i in D and M respectively (both smoothed appropriately).

Effect of similarity functions on performance. Figure 10.1 gives an example of the kind of effect that a similarity function can have on system performance in TDT. We used the tracking task as a convenient example, since we find that results observed in tracking usually generalize to other tasks. We measured performance of our tracking system on the TDT-2 multi-lingual corpus with four different similarity functions. Using *cosine* with *tf·idf* weighting scheme gives the best performance in the lower section of the graph. *Language modeling* system achieves the highest precision at low recall, but shows poor performance at high recall. *Kullback-Leibler* divergence is more stable, but uniformly worse than our *cosine* system. Finally, we observe the nearly-random performance for the variation of our language modeling system that does not use a *General English* model for topic discrimination.

2.3 Feature weighting

Another important issue is weighting of individual features (words) that occur in the stories. The traditional weighting employed in most IR systems is a form of tf·idf weighting. The *tf* component of the weighting – proportional to the number of times a term occurs in a document – represents the degree to which the term describes the contents of a document. The *idf* component – the inverse of the number of documents in which a term occurs – is intended to discount very common words in the collection (e.g., function words) since they have little discrimination power.

We discuss three different weighting schemes: *Inquery, tf* and *tf·idf*. We tested all three with the *cosine* and the *weighted sum* similarity metrics. Note that probabilistic similarity metrics (e.g., *LM, KL*) imply their own specific weighting schemes, and cannot be used with the schemes described below.

InQuery. The following tf·idf scheme was designed specifically for the InQuery engine, and has proven to be particularly effective in TREC evaluations:

$$tfcomp = \frac{tf}{tf + 0.5 + 1.5\frac{len_d}{len_{avg}}}$$

$$idfcomp = \frac{\log(N/df)}{\log(N+1)}$$

The *tf-comp* component has a general form of $tf/(tf+K)$, where *tf* is the raw count of term occurrences in the document, and K influences the significance we attach to seeing consecutive occurrences of the term in a particular document. The functional form of *tf-comp* is strictly increasing and asymptotic to 1.0 as *tf* grows without bounds. The effect is that we assign a lot of significance

to observing a single occurrence of a term, and less and less significance to consecutive occurrences. This is based on the observation that documents that contain an occurrence of a given word w are more likely to contain successive occurrences of w.

The parameter K influences how aggressively we discount successive occurrences, and in *InQuery* is set to be the document length (len_d) over average document length in the collection (len_{avg}). This means that shorter documents will have more aggressive discounting, while longer stories will not assign a lot of significance to a single occurrence of a term. This approach is adapted from Okapi [Robertson et al., 1995].

The *idf-comp* component is the logarithm of the inverse probability of the term in the collection, normalized to be between 0 and 1. N denotes the total number of documents in the collection, while *df* shows in how many of those documents the term occurs. This particular *idf* formulation arises naturally in the probabilistic derivation of document relevance under the assumption of binary occurrence and term independence.

tf. This weighting scheme is simply the actual *tf* value used in the *tfcomp* formula above—i.e., the number of times the term occurs in the story. The intuition behind omitting the *idf* component is that feature selection at other points in the process will choose only medium- and high-idf features with good discrimination value. As a result, the *tf*-only weighting scheme is less likely to work at high dimensionality when low-idf features will appear and need to be down-weighted.

tf·idf. This weighting scheme is simply the raw *tf* component times the *idf* component of the *InQuery* scheme. This weighting method boosts the importance of multiple occurrences of a feature over that given in the *InQuery* scheme.

Effects of feature weighting. Figure 10.2 shows a sample effect from varying weighting schemes when using *cosine* as a similarity measure. As an example, we use the tracking performance on TDT2 multi-lingual corpus with 4 training examples. We observe that *InQuery* weighting performs significantly worse with cosine. Performance of *tf* weighting scheme is consistently worse than either *InQuery* or *tf·idf*. A detailed analysis is provided in Section 3.

3. Tracking

Our research on tracking consisted mostly of making our parameter choices by sweeping a range of values. After doing a number of preliminary experiments, we settled on centroid representation of topics (i.e., average all N_t training stories together), and cosine comparison of stories to topics. The other

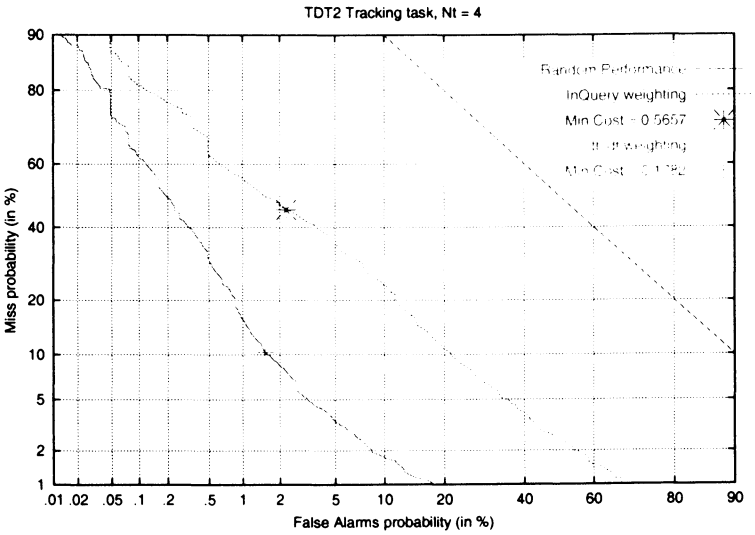


Figure 10.2. DET plot of performance on the tracking task for different weighting schemes. *tf·idf* weighting significantly outperforms *InQuery* weighting.

parameters (weighting, number of features, adapting thresholds) were chosen by a parameter sweep as shown in Table 10.1.

It is interesting to note the difference between effectiveness of InQuery's weighting function (Okapi *tf-comp*) compared to just using the *tf* count directly (*tf·idf* scheme). This difference is surprising because the Okapi *tf-comp* function has been widely adopted in IR—yet here it appears to be less useful. We posit this is because the Okapi *tf-comp* function is valuable for high-precision (low false alarm) tasks such as information retrieval. In the TDT tracking task, the optimum score is in a part of the error tradeoff curve that is less significant for IR. Section 7 provides further evidence that TDT evaluations require substantially different parameter settings than TREC evaluations.

We normalized the scores by comparing all N_t training stories to the centroid and then finding the average of those N_t similarities. During tracking, all subsequent story similarities were divided by that average score. So an "average on-topic story" would have a score of 1.0.

If the topic was adapted, the average was recalculated using the original N_t training stories as well as the stories that had been included in the topic. Adapting did not provide any significant reduction in the cost. This is consistent with results from TDT 1998, though continues to surprise us.

We selected using 1000 features (the full story), tf·idf weighting of those features, and no adapting. The threshold was selected depending on the task, as follows:

Weight	Terms	Adapt	Cost
Reference boundaries, $N_t = 4$			
tf·idf	**1000**	**no**	**0.2255**
tf·idf	100	no	0.2560
tf·idf	50	no	0.2992
tf·idf	20	no	0.3718
tf·idf	10	no	0.4082
Inquery	1000	no	0.6038
Inquery	100	no	0.2663
Inquery	50	no	0.3102
Inquery	20	no	0.3761
Inquery	10	no	0.5879
Automatic boundaries, $N_t = 4$			
tf·idf	1000	no	0.2586
tf·idf	1000	1.0	0.3146
tf·idf	100	no	0.2840
tf·idf	100	1.0	0.3451

Weight	Terms	Adapt	Cost
Reference boundaries, $N_t = 1$			
tf·idf	1000	no	0.2673
tf·idf	100	no	0.2906
tf·idf	50	no	0.3311
tf·idf	20	no	0.3751
tf·idf	10	no	0.4487
tf·idf	1000	1.0	0.2673
tf·idf	1000	0.9	0.2673
tf·idf	1000	0.8	0.2673
tf·idf	1000	0.7	0.3550
Automatic boundaries, $N_t = 1$			
tf·idf	1000	no	0.9533
Inquery	1000	no	0.9720
Inquery	1000	1.0	0.9816
Inquery	1000	0.9	0.9730

Table 10.1. Result of parameter sweep for tracking run on TDT-2 training data

$N_t = 1$	reference boundaries	0.07
$N_t = 1$	automatic boundaries	0.13
$N_t = 4$	reference boundaries	0.07
$N_t = 4$	automatic boundaries	0.13

The threshold was chosen by sweeping through the scores on the training data and finding the threshold that yielded the best normalized tracking cost. Figure 10.3 shows the performance of the final system on the TDT-2 multi-lingual dataset with reference boundaries.

4. Cluster Detection

As with tracking, we ran wide parameter sweeps on the six month TDT-2 corpus. We also checked our choice of parameters on different languages. For languages, we tried eng-nat (English-only corpus in its natural language—i.e., English), mul-eng (English and Mandarin, with the Mandarin translated into English by SYSTRAN), and mul-nat (English and Mandarin, each in their own "natural" language).

Our initial experiments suggested that 1-NN approach to detection was consistently outperforming the centroid approach. Furthermore, using k-NN approach with k =2, 4 or 8 did not improve performance over 1-NN. This is somewhat surprising, since we expect higher values of k to provide more stable comparisons. As a result of these experiments, our final system used 1-NN story comparison, so that a story was added into the topic that contained a *single* story to which it was very similar. Comparison was done using the cosine

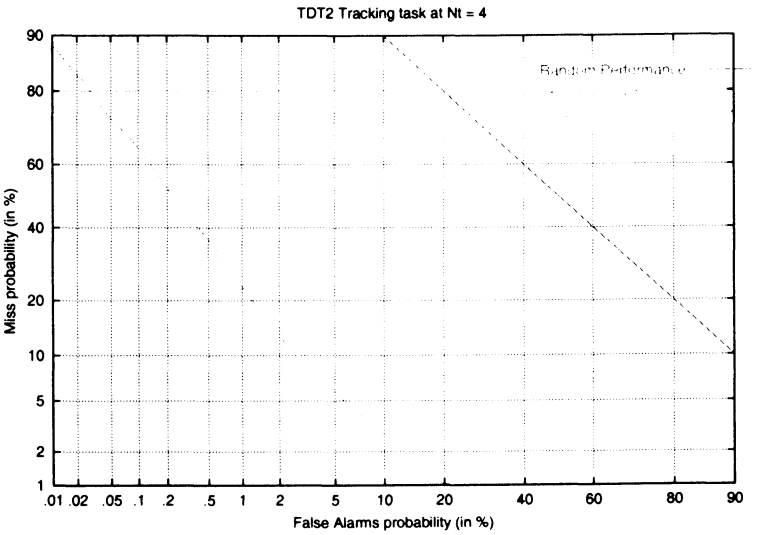


Figure 10.3. DET plot of performance on the Tracking Task on TDT-2 multi-lingual corpus using *cosine* with *tf·idf* weighting. Nt = 4 training examples were given, and no adaptation was performed.

measure; weighted sum resulted in consistently lower performance. Idf values were calculated using a retrospective corpus.

Table 10.2 shows the result of the parameter sweep for selecting the comparison function, the weighting, and the threshold θ_{match}. The optimal combination for the six-months of data was 1-NN, full dimensionality, idf weighting, and threshold of 0.20. We repeated the same parameter sweep for different language conditions, and observed that the optimal settings were relatively stable. Figure 10.4 shows the performance of our detection system with optimal parameter settings on the TDT-2 multi-lingual corpus.

As part of a cooperative project with BBN's Oasis system, we have begun looking at cluster detection on "real world" data and in a "real world" evaluation setting. It is obviously from the very first attempts that 1-NN cluster formation will not be appropriate. The created clusters have a property that is common among algorithms of the "single link" genre: they tend to be "stringy" with stories that are linked together in long chains, but that may not hold together as a group. Using the optimal settings trained on the TDT-2 corpus (i.e., our TDT 2000 parameters), we found clusterings containing 100s of at best marginally related stories.

The evaluation measure currently used in TDT cluster detection rewards a system for getting the bulk of a topic's stories together, and does not appear to penalize enough for mistakes. At a minimum that means that the cost values

Metric	Weight	Thresh	Cost
cosine	tf·idf	0.04	0.9253
cosine	tf·idf	0.06	0.7707
cosine	tf·idf	0.08	0.5981
cosine	tf·idf	0.10	0.4673
cosine	tf·idf	0.16	0.2604
cosine	tf·idf	0.18	0.2334
cosine	**tf·idf**	**0.20***	**0.2193**
cosine	tf·idf	0.22	0.2212
cosine	Inquery	0.02	1.0000
cosine	Inquery	0.04	1.0000
cosine	Inquery	0.06	0.9904
cosine	Inquery	0.14	0.6219
cosine	Inquery	0.16	0.5289
cosine	Inquery	0.18	0.4383

Metric	Weight	Thresh	Cost
wsum	tf·idf	0.04	0.9804
wsum	tf·idf	0.06	0.9569
wsum	tf·idf	0.08	0.9569
wsum	tf·idf	0.10	0.9569
wsum	tf·idf	0.16	0.9560
wsum	tf·idf	0.18	0.9560
wsum	tf·idf	0.20	0.9246
wsum	tf·idf	0.22	0.9035
wsum	Inquery	0.02	0.9245
wsum	Inquery	0.04	0.9245
wsum	Inquery	0.06	0.8393
wsum	Inquery	0.14	0.2713
wsum	Inquery	0.16	0.2832
wsum	Inquery	0.18	0.3101

Table 10.2. Result of parameter sweep for cluster detection run on TDT-2 training data.

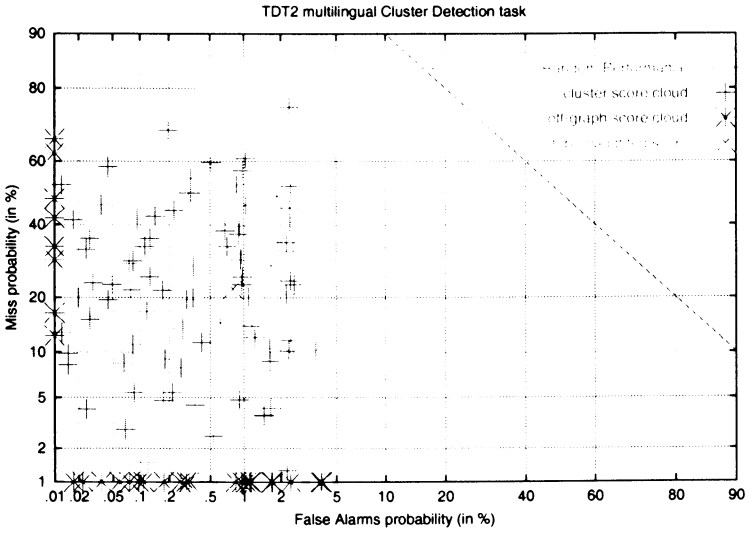


Figure 10.4. DET plot of performance on the Cluster Detection Task on TDT2 multi-lingual corpus using *cosine* with *tf·idf* weighting.

for detection need to be different for the Oasis task. At worst, it means that the detection cost function is inappropriate.

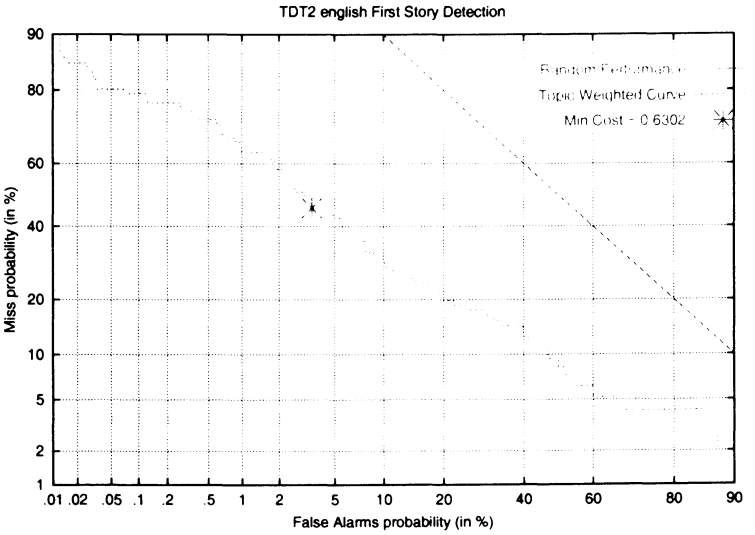


Figure 10.5. DET plot of performance on the First Story Detection Task on TDT2 multi-lingual corpus using *cosine* with *tf·idf* weighting.

5. First Story Detection

Our first story detection system was run identically to the cluster detection system, except that a separate parameter sweep was used to determine optimal parameter settings. The emitted score was one minus the detection score—i.e., the confidence that this story is new (rather than on a topic).

Idf was calculated from a retrospective corpus. By running a parameter sweep similar to the one done for cluster detection, we selected 1-NN topic representation, the cosine as a similarity measure, the *tf·idf* weighting scheme, and all features to represent each story. We selected 0.20 as the threshold—the same value as used in clustering, despite the different measures. We are somewhat surprised by this result, but have not yet investigated it. Figure 10.5 shows the performance achieved with optimal parameters on the TDT2 dataset.

Note that the performance of the first story detection system is considerably worse than similar tracking, detection or link detection systems (compare, for instance, Figure 10.5 to Figure 10.3). Section 7 advocates a theoretical justification of why first story detection exhibits such high error rates.

6. Link Detection

Link Detection can be seen as a fundamental task of TDT. It isolates and directly evaluates the component that is implicitly present in all other TDT tasks: comparing two pieces of text to determine whether they discuss the same

Weight	Thresh	Cost
tf·idf	0.02	1.6619
tf·idf	0.04	0.6322
tf·idf	0.05	0.4591
tf·idf	0.06	0.3769
tf·idf	0.065	0.3523
tf·idf	0.07	0.3412
tf·idf	0.075	0.3289
tf·idf	**0.08 ***	**0.3200**
tf·idf	0.085	0.3235
tf·idf	0.09	0.3216
tf·idf	0.10	0.3248
tf·idf	0.12	0.3583
tf·idf	0.14	0.4084
tf·idf	0.16	0.4641

Weight	Thresh	Cost
Inquery	0.02	4.2889
Inquery	0.04	3.3705
Inquery	0.05	2.8463
Inquery	0.06	2.3356
Inquery	0.065	2.1033
Inquery	0.07	1.8761
Inquery	0.075	1.6715
Inquery	0.08	1.4895
Inquery	0.085	1.3109
Inquery	0.09	1.1864
Inquery	0.10	0.9522
Inquery	0.12	0.6969
Inquery	0.14	0.5994
Inquery	0.16	0.6063

Table 10.3. Result of parameter sweep for link detection run on TDT-2 training data.

topic. We devoted a significant amount of research to link detection, and in this section will report on some preliminary results that seemed particularly promising. We explored how a query expansion technique from information retrieval could smooth the compared stories, and how score normalization depending on language mix can improve results.

6.1 Baseline Link Detection

We ran a parameter sweep to select the similarity measure, the weighting scheme and the threshold for comparison. Table 10.3 shows the cost function varying over a range of parameter values. We found that the cosine measure with *tf·idf* weighting consistently outperformed all other combinations. A threshold of 0.08 worked best. Idf scores were taken from a retrospective corpus. Figure 10.6 shows Link Detection performance with optimal baseline parameter settings.

6.2 LCA smoothing

In SIGIR 1996, the CIIR presented a query expansion technique that worked more reliably than previous "pseudo relevance feedback" methods.[Xu and Croft, 1996] That technique, Local Context Analysis (LCA), locates expansion terms in top-ranked passages, uses phrases as well as terms for expansion features, and weights the features in a way intended to boost the expected value of features that regularly occur near the query terms.

Because LCA has been so successful in IR tasks, we felt it was appropriate to explore it as a smoothing technique in TDT's story link detection task. That is,

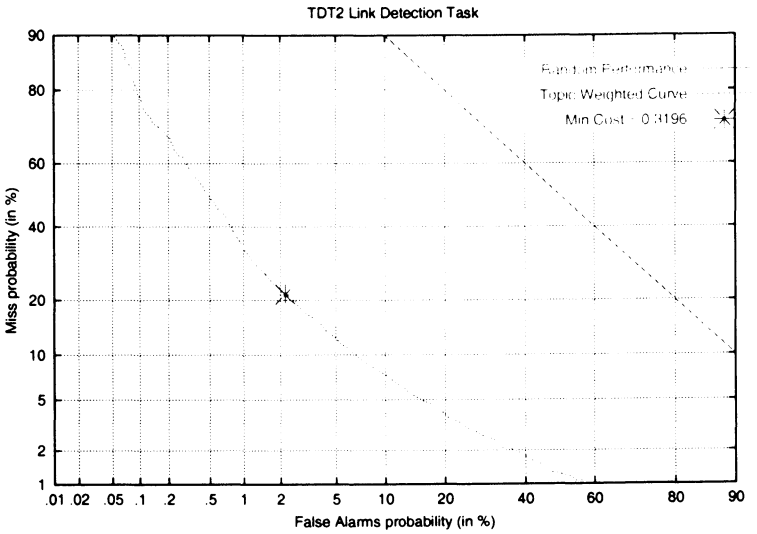


Figure 10.6. DET plot of performance on the Link Detection Task on TDT2 multi-lingual corpus using *cosine* with *tf·idf* weighting.

each story is treated as a "query" and expanded using LCA. Additional words that occur in the corpus very near the words in the story are added into each story and the resulting, larger, stories are compared as before.

We first provide some details about how LCA works, and then discuss its explicit use and results in TDT.

LCA used for Link Detection. We used LCA query expansion to replace the original story vector with a different, smoothed one. We first converted the story to a vector as before, selecting either Inquery or tf·idf as a weighting function. We then select the n most highly weighted features from that vector and discard all other features.

Those n features are used as a query to find the s stories from the *evaluation* corpus (TDT-3) that are most similar to features (as vectors). Except where noted otherwise below, we only allow those stories to come from stories that appeared *before* the story being expanded. (We could have used any stories up until the later of the two stories, but have not yet explored that adjustment.)

We extract all features from those s stories and weight them based upon their proximity to the original n "query" features. The LCA weighting function is a complex heuristic that gives higher weights to features that occur with many query words.[Xu and Croft, 1996] We select the top n LCA expansion features and add them to the vector. Note that it is possible for some of the *original* n

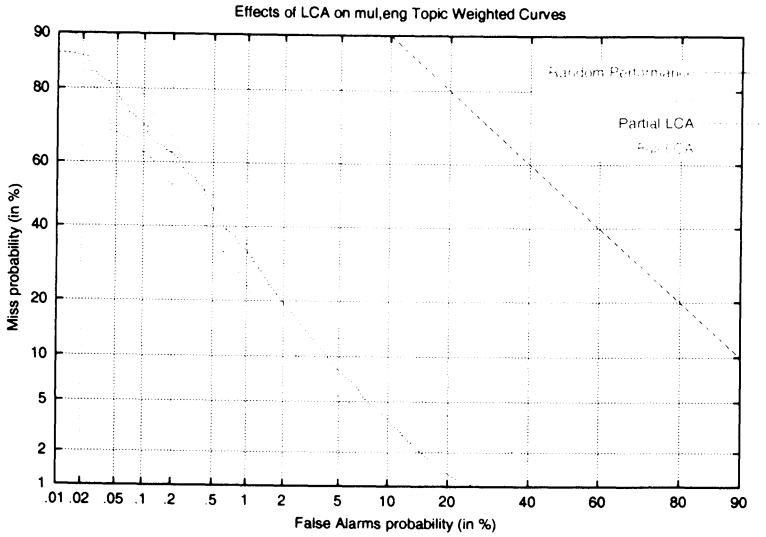


Figure 10.7. Results of LCA smoothing on Link Detection task. Experiments were done on the TDT-2 corpus. The lines hit the x-axis from left to right as "partial," "full," and "none."

features to re-appear as LCA features. The resulting vector has anywhere from n to $2n$ unique features.

The new features are added in with weights that start at 1.0 and smoothly drop down to $1.0 - (n-1)0.9/n$. This is the common weighting function for LCA features, and may not be the best choice for adding into the vector.

The result is that a story's vector is replaced by n to $2n$ features with weights that are a combination of Inquery or tf·idf weights, and LCA weights.

For this study we used $s = 20$ stories for expansion, used $n = 100$ features from each story, and added $n = 100$ expansion features.

LCA experiments. Figure 10.7 shows the impact of story smoothing using LCA on the link detection task. The curve that is consistently worst is the DET plot for no smoothing at all: our baseline. The next curve toward the original (it moves closest to the origin at both ends) is the result of using LCA as described above. The curve that comes closest to the origin is a "cheating" run that uses the *entire* TDT-3 corpus for expansion, meaning that a story could be expanded by stories that follow it and not just those in the past. Even without looking ahead, the value of LCA smoothing is apparent.

For our experiments, we used either the Inquery or the *tf·idf* weighting function both for determining the top n features of the story, and for finding the best-matching stories for expansion. Our best results in non-LCA Link

Detection were obtained with the *tf·idf* weighting function, but with LCA, *InQuery* weights performed better. Why?

We hypothesize that the reason is that query expansion requires highly accurate retrieval of the type that is typical in an IR system. The cost of expanding using non-relevant passages is very high: the query will be expanded in a direction that is not related to the original request. Our tf·idf weight is well known to be less effective in IR, so we expect it generates less relevant expansion terms. Since those terms account for up to half of the story's representation, it is very important that they be accurate.

As a future direction of research, we will consider using two separate weighting schemes, and perhaps two different similarity functions: one for fetching best-matching passages for each story, and the other for comparing expanded versions of the two stories to each other. The first could be tuned to high-precision performance (e.g., *weighted sum* with *InQuery* weighting), while the second would have to be more recall-oriented (e.g., *cosine* with *tf·idf* weighting).

6.3 Cross-language score normalization

Effects of SYSTRAN translations. During our experiments we stumbled upon an interesting effect of Mandarin documents on performance. We observed that the performance of our story-link detection system was noticeably worse on a multi-lingual dataset than it was on the English-only data. We hypothesized that the drop in performance could be due to lexical differences between the use of language in native English stories and in SYSTRAN translations of Chinese stories.

To test this hypothesis we performed the following post-hoc experiment. We partitioned our set of story pairs into three subsets: (1) pairs where both stories are native English stories, (2) pairs where both stories are SYSTRAN translations of Chinese, and (3) pairs where one story is a native English story and the other is the SYSTRAN translation. Then we analyzed the distributions of similarities of stories in the pair for each subset. Figure 10.8 presents distribution plots separately for on-target (both stories discuss the same topic) and off-target (stories discuss different topics) pairs in each subset.

It is evident that similarity distributions are very different for different subsets of pairs. On average, two SYSTRAN stories have a higher expected similarity than do two native English stories; the expected similarity of a SYSTRAN story to a native English story is even lower. Note that this observation holds for both on-target and off-target story pairs, but the effect is much more pronounced for on-target pairs.

We suspect the differences are due to the limited vocabulary of SYSTRAN translations. Any machine translation system, including SYSTRAN, has a

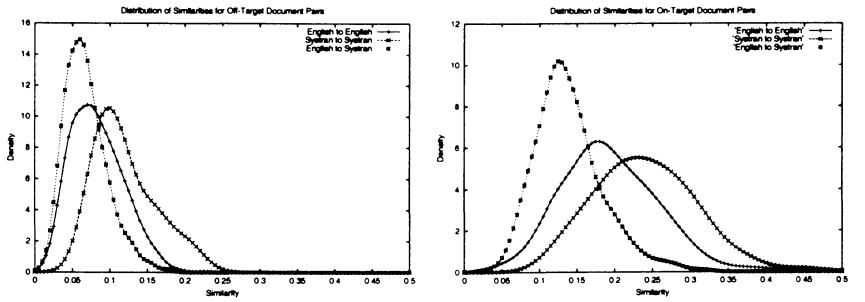


Figure 10.8. Effect of language on distributions of story similarities. Left: off-target story pairs. Right: on-target story pairs

relatively small vocabulary, whereas native English authors tend to use a much wider range of words. Also, SYSTRAN uses words consistently from story to story, whereas different human authors might use different words to describe the same idea. Inconsistent use of words leads to smaller expected word overlap between any two stories, which translates to lower expected similarity between two stories.

Whatever the cause, the differences in similarities present a serious challenge to effective cross-lingual story-linking. Suppose two given stories have a similarity of 0.1. If we know that both stories are SYSTRAN translations, the pair is most-likely off-target (from Figure 10.8 we see that probability of getting a 0.1 similarity in an on-target SYSTRAN pair is extremely low). However, if we know that one story is native English, and the other is a SYSTRAN translation, the pair is most-likely on-target, since the probability of getting 0.1 is higher for on-target pairs (Figure 10.8). This example implies that our similarity values are not directly comparable when pairs of stories involve multiple languages. To make them comparable, we need to normalize the similarities with respect to the source of stories in the pair.

Compensating translation effects. There exist a number of normalization techniques, ranging from simple range normalization and linear scaling (used in our tracking approach) to more elaborate techniques. We consider a probabilistic normalization technique where we replace the similarity x of a pair from subset S with the posterior probability that the pair is on-target $P(T|x, S)$, given the similarity x and subset S. If we have access to distributions of on-target similarities $P(x|T, S)$ and off-target similarities $P(x|N, S)$, we can use Bayes rule to derive the posterior:

$$P(T|x, S) = \frac{P(x|T, S)P(T, S)}{P(x|T, S)P(T, S) + P(x|N, S)P(N, S)}$$

Note that estimating the posterior requires knowledge of relevance judgments for each pair (to estimate $P(x|T,S)$ and $P(x|N,S)$). What we would do in practice is estimate the probabilities from the training data and then apply the transformation to the similarities in the testing data.

A number of parametric and non-parametric techniques could be used to estimate the conditional densities $P(x|T,S)$ and $P(x|N,S)$. In this work we chose non-parametric *kernel* density estimators because they can provide an arbitrarily close fit to the training data [Bowman and Azzalini, 1997]. The conditional probability of x is a function of every story pair in the training set S:

$$P_{\phi,h}(x|S) = \frac{1}{h|S|}\sum_{y\in S}\phi(\frac{x-y}{h})$$

Here ϕ is the *kernel*, which can be any probability density function, and h is the *bandwidth* parameter, representing the desired degree of smoothness. For kernel estimators the choice of ϕ has very little effect, as long as it is unimodal, symmetric and smooth. We selected Gaussian kernels:

$$\phi(x) = \frac{1}{\sqrt{2\pi}}e^{-x^2/2}$$

The bandwidth, h, on the other hand, has very strong effects on the final distribution. We used an automatic bandwidth selection technique:

$$h = \gamma E_M|\vec{X} - E_M(\vec{X})|$$

where $\vec{X}$ is the set of points for which we are estimating a density, E_M is a median operator, and γ is a normalization constant. To get an intuition for this formula, observe that it is similar to estimating a sample variance: $V = E(\vec{X}-E\vec{X})^2$. The differences are: (i) we use a median E_M rather than the mean E, and (ii) we use absolute-value deviations rather than squared deviations.[1]

Figure 10.9 shows the effects of applying our normalization to the training set of story-link pairs. The system that used normalized similarities shows a small but consistent improvement over no normalization. In this case we performed a cheating experiment, using the training data to normalize itself.

To better understand the effects of our normalization we plotted the overall densities of the original similarities (top half of Figure 10.10), and normalized similarities (bottom half). The main effect is in spreading the distributions apart. However, our normalization also introduces very "heavy" tails in both densities on the bottom half of Figure 10.10, and the tails are "bumpy", which

[1] For more details, refer to page 31 of [Bowman and Azzalini, 1997].

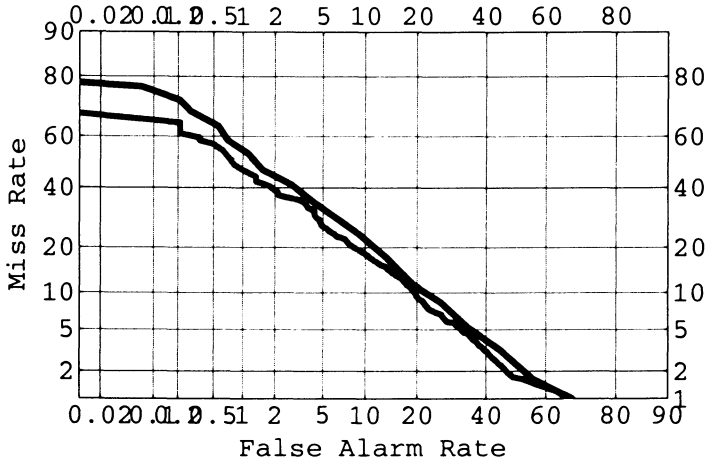


Figure 10.9. Improvement in performance resulting from normalization of similarities. Lower curve represents normalized system.

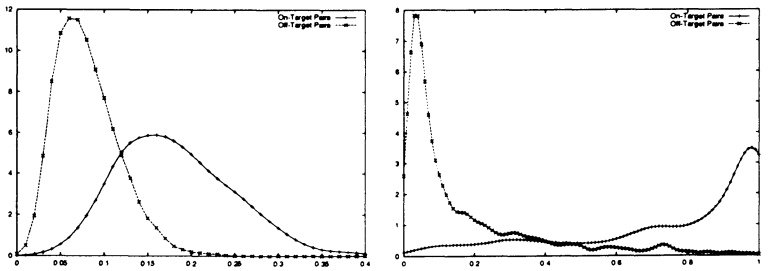


Figure 10.10. Effect of score normalization on similarity distributions. Left: distributions before normalization. Right: after normalization.

means that our normalization is non-monotonic (higher similarities don't always mean higher probability of being on-target). We suspect that bumpiness is the result of over-fitting the density. Possible ways to avoid this problem would be to increase the bandwidth h or use a parametric density estimator instead of kernel estimator described above.

7. Bounds on Effectiveness

In this section we show two things:

1 Tracking performance is approximately what we expect given state-of-the-art information filtering systems from TREC.

2 If an FSD system is built using a tracking system, it is extremely unlikely that FSD effectiveness can be satisfactory. We do not suggest that FSD is unsolvable, only that effective FSD is not a simple matter of improving tracking technology.

The work in this section is described in more detail elsewhere.[Allan et al., 1999, Allan et al., 2000]

7.1 Expected Tracking Performance

Information filtering has been extensively studied in the context of TREC conferences, and we can reasonably assume that current TREC filtering systems represent the state of the art. TDT tracking is very similar to filtering, the only differences being that in filtering: (i) a short topic description in the form of a query is available and (ii) the system may receive additional feedback beyond N_t training stories.

The DET curve of Figure 10.11 shows two tracking runs from the TDT-2 evaluation data. It also shows two runs from a TREC filtering task. We used the UMass filtering system, which has consistently ranked high in TREC evaluations. The system was modified to be more like TDT tracking (i.e. we started with no query and did not allow additional feedback).

One thing that the graph shows is that tracking performance at $N_t = 4$ is near the performance that filtering achieves with similar starting information. Although the tasks were run on completely different corpora, and had different definitions, tracking performance is approximately what filtering performance predicts. We hypothesize that the wildly different performance of the tasks for $N_t = 1$ is because news topics are more focused (e.g., "Oklahoma City bombing") than TREC filtering queries (e.g., "drug legalization benefits"). As a result, a single story is a good representative of a news topic, but it might take several documents to isolate the information pertinent to a hidden query.

We would like to postulate that the tracking technology developed in TDT is essentially the same as the filtering technology in TREC. Each is tuned to

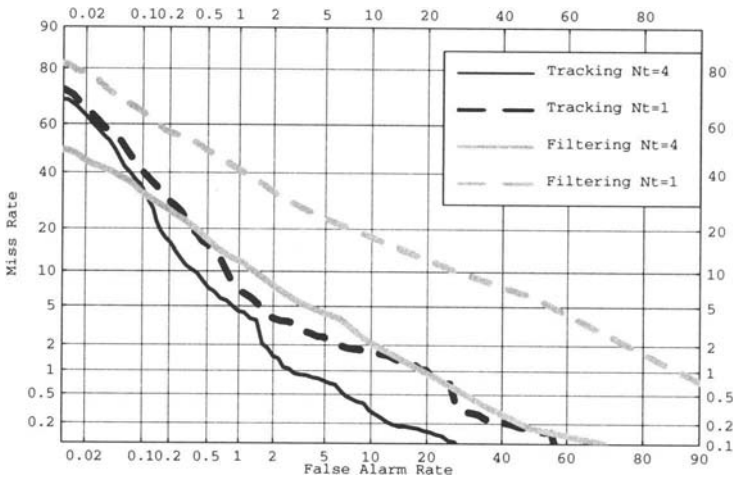


Figure 10.11. DET plot of two filtering and two tracking runs, each with the "query" generated from $N_t = 1$ or 4 stories.

its own evaluation measures: note how filtering runs are better in the upper portion of the graph. We suspect this is caused by TREC researchers tuning their systems to precision-oriented performance measures in TREC. However, small differences aside, we suggest that TDT tracking performance is essentially what one would expect by looking at TREC filtering systems.

7.2 Bounds on FSD

We now turn our attention to a less obvious relationship between tracking technology and first story detection (FSD). One possible solution to FSD is to apply tracking technology. Intuitively, the system marks the first story of the corpus with a very high score (it *must* be the first story on any topic in the corpus). It then begins tracking that story. If the second story tracks, it is assigned a low FSD score. If it does not track (is not on the same topic as the first story), it is assigned a high FSD score, and the system starts tracking that one, too. At any point, the system is tracking numerous topics—in fact, if the system makes an FSD false alarm, it will be tracking some topics in multiple ways.

It should be clear that a perfect tracking system (for $N_t = 1$) yields a perfect FSD system. However, tracking systems are far from perfect. What sort of FSD performance can we expect from a state-of-the-art tracking system?

It is possible to derive expected FSD error rates from average TDT error rates (omitted here, see [Allan et al., 2000]). The result will be lower- and upper-

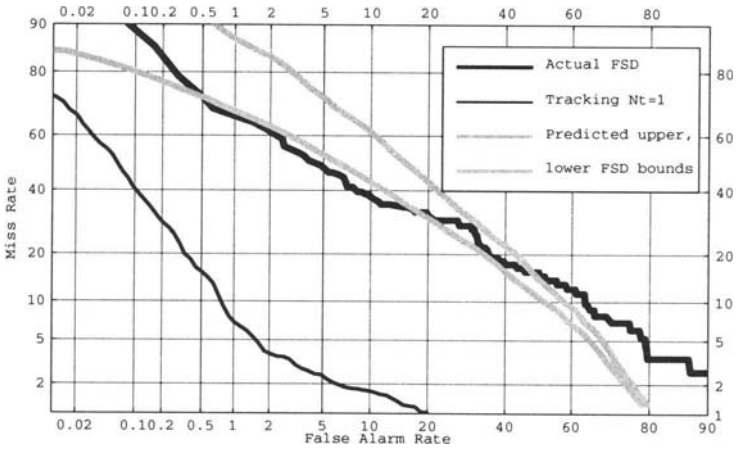


Figure 10.12. The lower-left curve is a tracking DET curve for $N_t = 1$. The upper part of the graph shows the lower- and upper-bound predicted performance for tracking-based FSD error rates in grey, as well as the actual system performance of an FSD system in black.

bounds on expected FSD performance. We emphasize that the predictions only make sense if we assume that the FSD system uses an approach that is based upon tracking. Figure 10.12 shows both the appropriate DET curves. Note that the FSD error rates fall nicely within the performance that is predicted by tracking. This result suggests that our FSD system is working about as well as we could expect.

7.3 Difficulty of improving FSD

The predicted and actual error rates of a tracking-based FSD system are in fact not very good: they are unacceptably high for all but a few applications, no matter what threshold on the DET curve is used.

We assume that "reasonable" FSD performance is approximately equal to the tracking DET curve shown in Figure 10.12 (the lower-left curve). A system that misses less than 10% of the first stories while generating only 0.5% false alarms is acceptable for many applications.

Figure 10.13 shows the desired FSD curve (it is really just the tracking curve again) and lower- and upper-bounds on errors that encompass it. In order to achieve those bounds, we had to improve tracking performance for $N_t = 1$ by a factor of 20. The resulting DET curve is a small line segment in the lower left of the figure.

None of the research in TDT 1997 through TDT 2000 has resulted in a tracking DET curve that is substantially better than the ones in Figure 10.12.

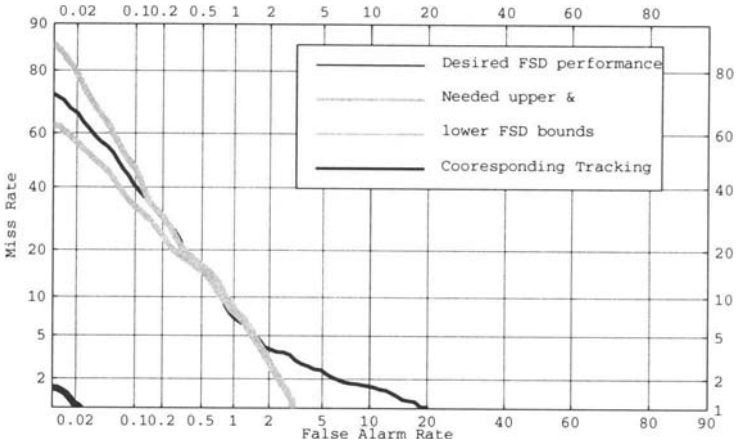


Figure 10.13. Shows desired FSD performance in black surrounded by reasonable confidence intervals. The extreme lower-left curve is the tracking performance that would be required to achieve desired bounds.

Further, as shown in Section 7.1, that level of effectiveness is comparable to that achieved by many years of filtering research at TREC. There is little reason to believe that tracking technology will improve 20-fold with current approaches.

We have shown how to reduce the FSD problem to a tracking task. We have also shown that a given error rate in tracking results in substantially worse error rates in a corresponding FSD system. Most importantly, we have shown that there is little reason to believe that tracking-based FSD effectiveness can be raised to the point that the technology is widely useful. To make first story detection usable, researchers must investigate novel methods for dramatically improving tracking effectiveness, or they must find new approaches to address FSD that do not depend on tracking work.

8. Automatic Timeline Generation

We have developed a technique for determining the relative importance of the occurrence of extracted features within text. Our technique requires an explicitly time tagged corpus, such as TDT with its stories that arrive at known times. With our technique we are able to analyze extracted features (named entities and noun phrases) and explicitly rank how likely these features are to be high content bearing. We are then able to group these features into clusters that correspond strongly with the notion of "topic" as defined in the Topic Detection and Tracking (TDT) study. Figures 10.14 and 10.15 show examples

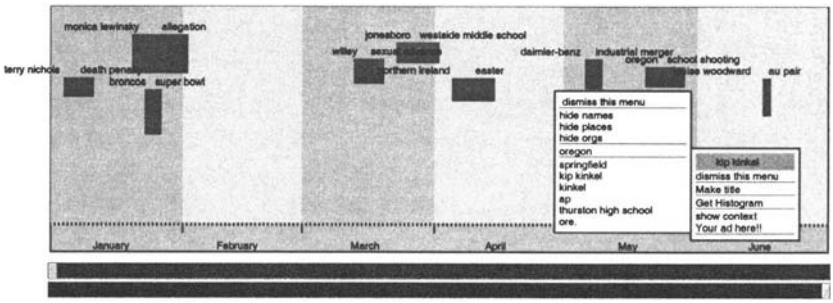


Figure 10.14. Overview of January - June, 1998. The topic labeled *monica lewinsky allegation* is the highest ranked topic, and the topic labeled *jonesboro westside middle school* is the second highest ranked. The pop-up on *oregon school shooting* shows significant named entities of *oregon*, *springfield*, *kip kinkel*, *kinkel*, *ap*, *thurston high school*, and *ore*. The other pop-up displays a submenu for obtaining more information on *kip kinkel*.

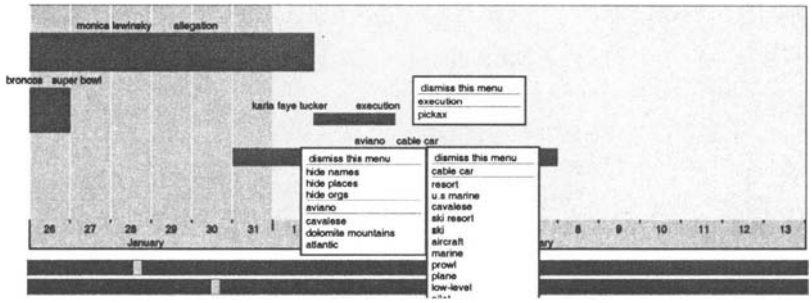


Figure 10.15. Detail, January 26 to March 13, 1998. Topics shown are *monica lewinsky allegation*, *broncos super bowl*, *karla faye tucker execution*, and *aviano cable car*. Additional phrases are displayed with the Karla Faye Tucker execution and the Aviano cable car crash.

of the system running. This work is described in more detail elsewhere.[Swan and Allan, 2000, Swan and Allan, 1999]

With the model that tokens are emitted by random processes, we assume two hypotheses as defaults. The assumptions are (1): the random processes generating tokens are stationary, meaning that they do not vary over time, and (2): the random processes for any pair of tokens are independent. We use the χ^2 measure to look for features that violate those hypotheses.

We built a system that constructed timelines such as those shown. We were curious whether it was finding "reasonable" events, so we ran a small evaluation. We used the entire TDT-2 corpus for our experiment, training on the 4-month development set, and evaluating on the held-out 2 months. The corpus was tagged by BBN using the Nymble tagger[Bikel et al., 1997], which identified 184,723 unique named entities. We also extracted noun phrases by running a shallow part of speech tagger[Xu et al., 1994], and labeling as a noun phrase any groups of words of length less than six which matched the regular expression (Noun|Adjective)*Noun. This led to a set of 1,188,907 unique noun phrases.

Our final run on the evaluation portion of TDT-2 (May and June 1998) produced 146 clusters of those features (based on pairing features by χ^2 value and time, and imposing a threshold on which could be paired). We believe that the clusters of features found are indicative of the major news stories that were covered by the news organizations during the time spanned by the corpus. We felt that the clusters were highly suggestive of the major news stories and provided as good a summation as could be obtained by an unordered collection of features. To test this, we hired four students (three undergraduates and one graduate student) to evaluate the clusters.

Of the 146 clusters 79 were judged three times, and 67 had four judgments. The four evaluators found that the great majority of groups were indicative of a single topic (71.2%, 79.4%, 82.2% and 90.2% of the groups judged), and the pairwise overlap on the judgments of how many topics were contained in a group was 73.6%. However the overlap expected by chance was nearly 70%, and the pairwise Kappa statistics ranged from 0.045 to 0.315, with a (weighted) average value of 0.223. The Kappa statistic is a measure of inter-evaluator reliability, and a value of 0.0 indicates an overlap that would be expected by chance and a value of 1.0 indicates perfect overlap. A Kappa value of 0.233 indicates poor agreement among evaluators and that the data are not reliable. This can also be seen by looking at the scores given individual groups. Only twenty of the 146 were not judged to be a single topic by the majority of assessors, and of these twenty there were only three where the assessors unanimously agreed.

We also asked the assessors to compare the generated groups with the TDT-2 topics and indicate if they agreed. Here the results were stronger. The (pairwise) overlap in topic/group matches was 86.7%, and the six pairwise Kappa statistics ranged from 0.600 to 0.785, with an average value of 0.699, indicating very good agreement. This indicates that if a topic is defined, the features our system selects are sufficient for recognizing the topic.

The groups of terms were automatically labeled and our assessors were asked to rate the usefulness of the label. Our assessors were asked to rank these on a six point Likert scale. In general our assessors felt that the labels were very poor, with an average rank of 2.8 (1 = poor, 6 = excellent). Our assessors

were in good agreement on the rankings, with the average standard deviation equaling 1.0.

We feel that the techniques presented in this study can make a significant contribution to the accessibility of information, as it allows the automatic generation of interactive overview timelines at modest cost. As archives of news, e-mails, historical newspapers, memos, and other such time based corpora become increasingly common in digital libraries we feel that this system, or one like it, will be a tremendous tool to allow broader access to electronic information.

9. Conclusions

The results that we have presented on the three detection tasks were acceptable, but not as high a quality as we would have liked. We believe that we have hit the limits of effectiveness that can be reached with simple IR-based approaches to story/topic comparison.

We spent considerable effort, including two months over the summer[Allan et al., 1999], working on FSD but were unable to achieve great improvements in the system. A major finding of that workshop, however, and one which we have extended since then[Allan et al., 2000], is the idea that tracking-based FSD systems cannot be effective enough. This result bolsters the idea that current approaches have hit their limits. We believe that event-based information organization as realized in TDT requires substantially different approaches and ideas.

We have some preliminary work that shows the value of smoothing stories by other, related stories in the corpus. We are simultaneously working on improved formal models for query expansion, and anticipate incorporating that approach into our language modeling ideas.

Score normalization is a key task within TDT that has not been important in areas such as information retrieval. We have been using distribution plots to recognize when normalization is likely to be helpful, and have shown that definitely helps within and across languages.

We have briefly presented our work on automatic timeline generation, work that we believe serves as an example of moving TDT ideas in new directions. We hope that a richer set of ideas and directions will yield new approaches and techniques for addressing the existing TDT tasks, as well as new tasks that arise.

Acknowledgments

The authors thank Daniella Malin, David Frey, and Vikas Khandelwal for their assistance in some of the experiments discussed in this chapter. The authors also would like to acknowledge the work of Ron Papka while he was a

student in the CIIR on the TDT research program, when he helped develop the CIIR's initial approaches to the TDT tasks.

This work was supported in part by the National Science Foundation, Library of Congress and Department of Commerce under cooperative agreement number EEC-9209623, in part by the National Science Foundation under grant number IRI-9619117, in part by SPAWARSYSCEN-SD grant number N66001-99-1-8912, and in part by the Air Force Office of Scientific Research under grant number F49620-99-1-0138. The opinions, views, findings, and conclusions contained in this material are those of the authors and do not necessarily reflect the position or policy of the Government and no official endorsement should be inferred.

References

[Allan et al., 1999] Allan, J., Jin, H., Rajman, M., Wayne, C., Gildea, D., Lavrenko, V., Hoberman, R., and Caputo, D. (1999). Topic-based novelty detection: 1999 summer workshop at CLSP, final report. Available at http://www.clsp.jhu.edu/ws99/tdt.

[Allan et al., 2000] Allan, J., Lavrenko, V., and Jin, H. (2000). First story detection in TDT is hard. In *Ninth International Conference on Information Knowledge Management (CIKM)*, pages 374–381. ACM Press.

[Bikel et al., 1997] Bikel, D., Miller, S., Schwartz, R., and Weischedel, R. (1997). Nymble: a high-performance learning name-finder. In *Fifth Conference on Applied Natural Language Processing*, pages 194–201. ACL.

[Bowman and Azzalini, 1997] Bowman, A.W., and Azzalini, A. (1997). *Applied Smoothing Techniques for Data Analysis*. Oxford Science Publications.

[Robertson et al., 1995] Robertson, S.E., Walker, S., Jones, S., Hancock-Beaulieu M.M, and Gatford, M. (1995). Okapi at TREC-3. In *Proceedings of the Text Retrieval Conference (TREC-3)*. NIST Special Publication.

[Swan and Allan, 1999] Swan, R. and Allan, J. (1999). Extracting significant time varying features from text. In *Eighth International Conference on Information Knowledge Management (CIKM'99)*, pages 38–45, Kansas City, Missouri. ACM.

[Swan and Allan, 2000] Swan, R. and Allan, J. (2000). Automatic generation of overview timelines. In *Proceedings of ACM SIGIR, Research and Development in Information Retrieval*, pages 49–56.

[Witten and Bell, 1991] Witten, I. and Bell, T. (1991). The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. *IEEE Transactions on Information Theory*, 37:1085–1094.

[Xu et al., 1994] Xu, J., Broglio, J., and Croft, W. B. (1994). The design and implementation of a part of speech tagger for english. Technical Report IR-52, Center for Intelligent Information Retrieval, University of Massachusetts, Amherst.

[Xu and Croft, 1996] Xu, J. and Croft, W. B. (1996). Query expansion using local and global document analysis. In *Proceedings of ACM SIGIR, Research and Development in Information Retrieval*, pages 4–11.

Chapter 11

Towards a "Universal Dictionary" for Multi-Language Information Retrieval Applications

J. Michael Schultz and Mark Y. Liberman
University of Pennsylvania

Abstract Multilingual information retrieval tasks such as Topic Tracking have yielded high-quality results simply using word-by-word translation approaches. However, the construction of translation dictionaries for new languages is expensive and time-consuming. We show that an appropriate metric for term selection in a monolingual English corpus allows us to define a fairly small list, containing about ten thousand inflected forms or about 7500 lemmas, which works essentially as well (for a particular monolingual document classification evaluation) as an unlimited vocabulary of more than 300,000 word forms does. We suggest that such a list can be taken to form the English axis of a sort of "universal dictionary" for document classification tasks, providing a much more efficient path to the addition of new languages.

1. Introduction

Recent formal evaluations suggest that simple techniques can accomplish high-quality automatic retrieval from multilingual document sets, at least for some tasks. These techniques are based on per-document word counts — what in the monolingual case are familiarly called "bag-of-words" models — generalized across languages using simple word-by-word translation. However, an important practical impediment to coverage of a large number of languages is the need for translation dictionaries. These are time-consuming and expensive to create by hand, and few language pairs have large enough parallel or comparable text corpora for statistical induction methods to be feasible. We show that an appropriate metric for term selection in a monolingual English corpus allows us to define a fairly small list, containing about ten thousand inflected forms or about 7500 lemmas, which works essentially as well (for a particular monolingual document classification evaluation) as an unlimited vocabulary of more than 300,000 word forms does. We suggest that such a list can be taken

to form the English axis of a sort of "universal dictionary" for document classification tasks, providing a much more efficient path to the addition of new languages.

If proper names can be treated separately, then the "universal dictionary" becomes even smaller — about 5 thousand terms. Given a new language for which no prior resources of any kind exist, an initial mapping for a term list of this size should require only a few person-weeks of work, even if done entirely by hand. Even smaller lists will still be much better than nothing, if terms are added to the translation dictionary in the order specified by a metric of the type we propose. Useful results should be achieved after only a few hours of work, with performance increasing to an asymptote as the translation dictionary reaches its full size.

Of course, term selection may have to be different for different subject areas. Health records can't be classified on the basis of a term list derived from a corpus of computer repair manuals. However, term selection is done on the English side only, so that a good general list — perhaps almost deserving the hyperbolic name of "universal dictionary" — can be derived from a very large topically balanced corpus, and more specific lists can be derived from easy-to-get corpora in specific domains.

In order to motivate some of its properties, we situate our "universal dictionary" experiment in the context of a description of our entrants in the TDT 1998 and 1999 tracking evaluations. Because of the extreme simplicity of our system, we feel that the results of the "universal dictionary" experiment ought to generalize to other approaches as well.

1.1 Multilingual Topic Detection and Tracking

The "tracking" task in the Topic Detection and Tracking (TDT) evaluation [2] begins with Nt seed stories from a news stream describing an event, where Nt is typically between 1 and 4. All subsequent stories in the news stream are then to be classified as to whether or not they are about the cited event. As in many other full-text information retrieval tasks, relatively simple techniques based on per-story word counts work quite well at TDT tracking. Such "bag of words" techniques can be set to operate in a way that combines a miss rate of 5% with a false alarm rate of 0.5% on this task, numbers that are nearly at the point where the inherent uncertainty of the task definition begins to make it hard to measure improvement.

When documents in different languages are added to the picture, simple word-by-word translation allows the same bag-of-words techniques to be applied with little or no change. The necessary "translation" can be produced by applying a conventional Machine Translation (MT) system to the document stream, or by simply selecting one or more target-language terms for each source

language term, either in document context or in isolation. TDT experiments with mixed Mandarin and English document streams have shown that simple implementations of such approaches work fairly well, coming within about 30% of monolingual performance on the cost metric for the TDT tracking task (a weighted sum of miss and false alarm rates). This difference is substantially smaller than the differences among algorithms for the monolingual task, and will doubtless be narrowed by on-going research into the improvement of translation dictionaries, the selection of translation equivalents, the treatment of proper names, and so forth.

These results from the TDT evaluations are consistent with the results from the rest of the literature, especially the various TREC evaluations. Bag-of-words measures of document similarity have been shown to work well for many tasks that can be built on top of document-to-document comparison, and simple word-by-word translation can in principle generalize these techniques to multilingual document sets with a modest performance cost.

However, the necessary word-by-word translation still requires a translation dictionary. In the simplest case, this is just a partial function from words in Language X and words in Language Y. Somewhat more complex translation dictionaries involve a relation, so that a given word in X may translate to more than one word in Y; in this case, some estimate of the frequency of different translations may be provided, and this estimate may be modified by the word's context in the Language-X document. There are many alternatives in detail: we might be dealing with word forms, stemmed or lemmatized words, or multi-word phrases; we may try to recognize proper names and transliterate them in a special way; we may try to provide special treatment for other categories such as dates, monetary amounts, and so forth. In the particular case of Mandarin and English, we also can take different approaches to the problem of word segmentation on the Mandarin side.

The results of the TDT tracking evaluations show that even very simple and unsophisticated approaches of this type can work surprisingly well, given a large bilingual dictionary to start with. This was true despite the fact that the coverage of the dictionary was not very good, and the quality of the dictionary was suspect in other ways, as it was derived by simple techniques from freely-available sources that were never intended for any such use.

1.2 Motivation for our experiment

The results of the TDT cross-language experiment, and similar results from the TREC cross-language track, are encouraging indications that solutions to some multi-lingual document retrieval or tracking tasks are within our grasp. However, if we face the problem of performing such a task on a document set that includes a new language for which we have no resources at all, several

daunting problems face us. If the new language is richly inflected, then some sort of stemmer or lemmatizer must probably be built – we will ignore this problem for our present purposes. But whatever the structure of the language, we need a translation dictionary.

The TDT experiments made use of Mandarin-Engish translation dictionaries involving on the order of 100K words, derived from a combination of "open" sources that permit free re-distribution. As far as we know, there is only one other language pair for which a free resource in this size range is available. For a dozen or so other languages, one might be able to buy large bilingual dictionaries in electronic form, from which such translation dictionaries might be created (semi-)automatically; or one might be able to buy an MT system of adequate quality for use in automated document comparison tasks. Beyond this point, we face keyboarding or scanning a paper dictionary, for the cases in which a suitable one exists; or constructing a translation dictionary from scratch. All of these options will be expensive and time-consuming. For example, creating a translation dictionary of 100K terms, assuming one minute spent on each translation, is roughly a person-year of work – and producing a translation a minute for a year, 9:00 to 5:00 with an hour for lunch, is a hard job at best.

Investment at this scale for tens or hundreds of languages is not unthinkable, given sufficient incentive, but one is certainly motivated to ask if there is an easier way. Do we really need such a large translation dictionary? Many words that occur in the document set are missing from the translation relation anyway – how damaging would even lower coverage be? Or to put it another way, if we only had the time or money to produce translations for N words, how well could we do for a given value of N?

In addressing such questions, we will get the clearest answers by looking first at the monolingual case. There are many detailed choices to be made in setting up a cross-language experiment, and many of these choices are likely to make a bigger difference in small-vocabulary systems than in large ones. A considerable amount of exploration of the algorithmic space would be necessary to find a local optimum for a small-vocabulary cross-language system, and it may well involve quite different choices than those that are optimal for a large-vocabulary system. Also, the translation dictionary we have used in our Mandarin-English experiments is not of especially high quality, with many missing words and many dubious translations. Thus in selecting vocabulary subsets, we typically find that a substantial fraction of the terms in a given selection are missing from the available translation dictionary, and that the fraction is by no means constant across all ways of selecting a subset of a given size. Thus these random flaws will add considerable noise to small-vocabulary tests.

For all these reasons, we think it is most informative to begin with a monolingual experiment: what is the performance of an English-only TDT tracking

system, if its vocabulary is artificially limited to a particular N words? Can we find a way of choosing N words so that the penalty for limiting the vocabulary is minimal?

Given a positive answer to this question, we can pose a second question: how should we configure a small-vocabulary cross-language system so that the cross-language penalty is as small as possible? We do not attempt to answer that question in this present paper.

2. Our TDT tracking algorithm

The similarity metric of our tracking system is based on the idf-weighted cosine coefficient described in [7] often referred to as $tf{\cdot}idf$. Using this metric, the tracking task becomes two-fold. The first stage is feature selection: we choose a set of features (words or stems) to represent a given topic. These features might be chosen from a single story or from multiple stories. The second stage is threshold determination: choosing a threshold on the $tf{\cdot}idf$ metric that optimizes the miss and false alarm rates for a particular cost function. In effect, the threshold selection normalizes the $tf{\cdot}idf$ similarity metric across topics.

The cosine coefficient as a document similarity metric has been investigated extensively. Here documents (and queries) are represented as vectors in an n-dimensional space, where n is the number of unique terms in the database. The coefficients of the vector for a given document are the term frequencies (tf) for that dimension. The resulting vectors are extremely sparse and typically high frequency words (mostly closed class) are ignored. The cosine of the angle between two vectors is an indication of vector similarity and is equal to the dot-product of the vectors normalized by the product of the vector lengths.

$$\cos(\theta) = \frac{\vec{A} \cdot \vec{B}}{\|\vec{A}\|\|\vec{B}\|}$$

$tf{\cdot}idf$ (term frequency times inverse document frequency) weighting is an ad-hoc modification to the cosine coefficient calculation which weights words according to their *usefulness* in discriminating documents. Words that appear in few documents are more useful than words that appear in many documents. This is captured in the equation for the inverse document frequency of a word:

$$idf(w) = \log_{10}\left(\frac{N}{df(w)}\right)$$

Where $df(w)$ is the number of documents in a collection which contain word w and N is the total number of documents in the collection.

For our implementation we weighted only the topic vector by idf and left the story vector under test unchanged. This allows us to calculate and fix an idf-scaled topic vector immediately after training on the last positive example story for a topic. The resulting calculation for the similarity measure becomes:

$$sim(a,b) = \frac{\sum_{w=1}^{n} tf_a(w) \cdot tf_b(w) \cdot idf(w)}{\sqrt{\sum_{w=1}^{n} tf_a^2(w)} \cdot \sqrt{\sum_{w=1}^{n} tf_b^2(w)}}$$

2.1 UPENN System Attributes

To facilitate testing, the evaluation stories were loaded into a simple document processing system. Once in the system, stories are processed in chronological order testing all topics simultaneously with a single pass over the data[1] at a rate of approximately 6000 stories per minute on a Pentium 266 MHz machine. The system tokenizer delimits on white space and punctuation (and discards it), collapses case, but provides no stemming. A list of 179 stop words consisting almost entirely of closed-class words was also employed. In order to improve word statistics, particularly for the beginning of the test set, we prepended a retrospective corpus (the TDT-1 corpus [4]) of approximately 16 thousand stories.

2.2 Feature Selection

The *choice* as well as *number* of features (here simply words) used to represent a topic has a direct effect on the trade-off between miss and false alarm probabilities. We investigated four methods of producing lists of features each sorted by their effectiveness in discriminating a topic. This then allowed us to easily vary the number of those features for the topic vectors[2].

1 Keep all features except for words on the stop list.

2 Relative to training stories, sort words by document count, keeping the n most frequent. This approach has the advantage of finding those words which are common across training stories, and therefore are more general to the topic area, but has the disadvantage of extending poorly from the $Nt = 16$ case to the $Nt = 1$ case.

3 For each story, sort by word count (tf), keeping the n most frequent. While this approach tends to ignore low count words which occur in

[1] In accordance with the evaluation specification for this project [2] no information is shared across topics.
[2] We did not employ feature selection on the story under test but used the text in entirety.

multiple training documents, it generalizes well from the $Nt = 16$ to the $Nt = 1$ case.

4 As a variant on the previous method we tried adding to the initial n features using a simple greedy algorithm. Against a database containing all stories up to and including the Nt-th training story, we queried the database with the n features plus the next most frequent term. If the separation of on-topic and off-topic stories increased, we kept the term, if not we ignored it and tested the next term in the list. We defined separation as the difference between the average on-topic scores and the average of the 20 highest scoring off-topic documents.

Of the feature selection methods we tried, the fourth one yielded the best results across varying values of Nt, although only slightly better than the much simpler third method. Occam's Razor prompted us to omit this complication from the algorithm. The DET curves[3] in Figure 11.1 show the effect of varying the number of features (obtained from method 3) on the miss and false alarm probabilities. The upper rightmost curve results from choosing the single most frequent feature. Downward to the left, in order are the curves for 5, 10, 50, 150 and 300 features. After examining similar plots from the pilot, training and development-test data sets, we set the number of features for our system to 50. It can be seen that there is limited benefit in adding features after this point.

2.3 Normalization / Threshold Selection

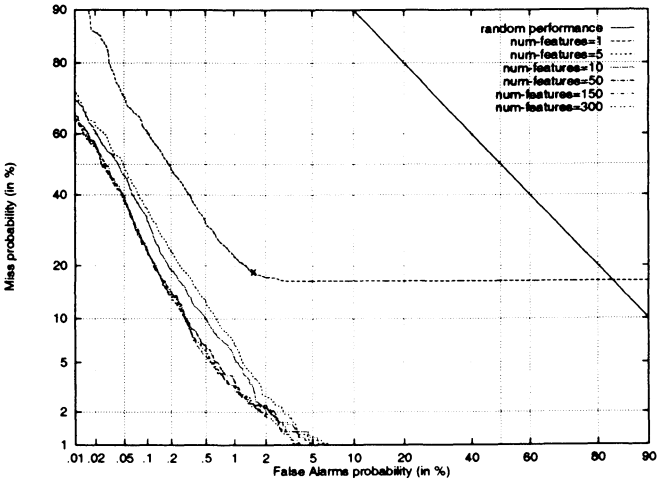
With a method of feature selection in place, a threshold for the similarity score must be determined above which stories will be deemed on-topic, and below which they will not. Since each topic is represented by its own unique vector it cannot be expected that the same threshold value will be optimal across all topics unless the scores are normalized. We tried two approaches for normalizing the topic similarity scores.

For the first approach we calculated the similarity of a random sample of several hundred off-topic stories in order to estimate an average off-topic score relative to the topic vector. The normalized score is then a function of the average on-topic scores of the training stories and the average and standard deviation of the off-topic samples[4]. The second approach looked at only the *highest* scoring *off-topic* stories returned from a query of the topic vector against a retrospective database with the score normalized in a similar fashion to the first approach.

[3] See [6] for detailed description of DET curves.

[4] σ(on-topic) is unreliable for small Nt but for larger Nt the σ(off-topic) was found to be a good approximation of σ(on-topic).

Figure 11.1. DET curve for varying number of features. (Nt=4, TDT2 evaluation data set, newswire and ASR transcripts)



Both attempts reduced the story-weighted miss probability by approximately 10% at low false alarm probability. However, this result was achieved at the expense of higher miss probability at higher false alarm rates, and a higher cost at the operating point defined by the cost function for the task defined in [2].

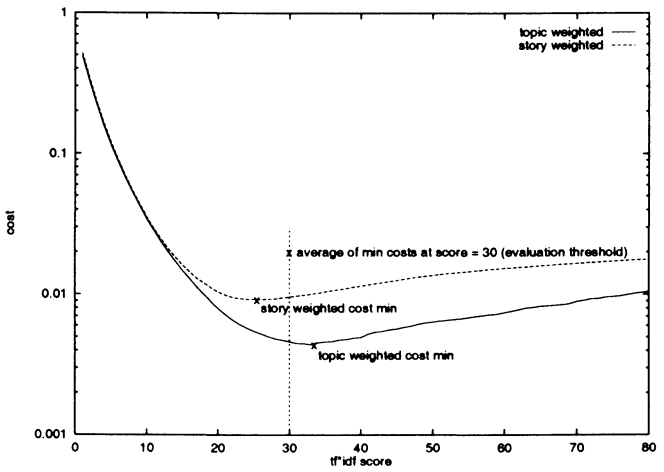
$$C_{track} = C_{miss} \cdot P(miss) \cdot P_{topic} + C_{fa} \cdot P(fa) \cdot (1 - P_{topic})$$

where

$C_{miss} = 1.0$ (the cost of a miss)
$C_{fa} = 1.0$ (the cost of a false alarm, changed to 0.1 in TDT3)
$P_{topic} = 0.02$ (the *a priori* on-topic probability)

Because of the less optimal trade-off between error probabilities at the point defined by the cost function, we choose to ignore normalization and look directly at cost as a function of a single threshold value across all topics. We plotted $tf{\cdot}idf$ score against story and topic-weighted cost for the training and development-test data sets. As our global threshold we averaged the scores at which story and topic-weighted cost were minimized. This is depicted in figure 11.2.

Figure 11.2. Story and topic-weighted cost as a function of $tf{\cdot}idf$ score. ($Nt = 4$, TDT2 training and development test data sets, newswire and ASR transcripts)



2.4 Tracking Results and Conclusions

We tried a number of approaches to optimize the $tf{\cdot}idf$ weighted cosine coefficient for the tracking task. In the end very simple feature selection with no normalization of topic scores performed as well or better than more sophisticated methods from other sites.

Table 11.1. Story weighted monolingual tracking results by site. ($Nt = 4$, TDT2 evaluation data set, newswire and ASR transcripts)

Site	$P(miss)$	$P(fa)$	C_{track}
UPenn1	0.0934	0.0040	0.0058
UMass1	0.0855	0.0043	0.0059
BBN1	0.1415	0.0035	0.0063
Dragon1	0.1408	0.0043	0.0070
CMU1	0.2105	0.0035	0.0077
GE1	0.1451	0.0191	0.0216
UMd1	0.8197	0.0062	0.0225
Ulowa1	0.0819	0.0492	0.0499

2.5 Generalization to mixed English/Mandarin document sets

For TDT3 we investigated a method of cross-lingual topic tracking built upon our cosine coefficient based monolingual approach. The system relies on

Table 11.2. Topic weighted monolingual tracking results by site. ($Nt = 4$, TDT2 evaluation data set, newswire and ASR transcripts)

Site	$P(miss)$	$P(fa)$	C_{track}
BBN1	0.1185	0.0033	0.0056
UPenn1	0.1092	0.0045	0.0066
Dragon1	0.1054	0.0049	0.0069
UMass1	0.1812	0.0038	0.0074
CMU1	0.2660	0.0023	0.0076
GE1	0.1448	0.0226	0.0251
UMd1	0.6130	0.0156	0.0275
Ulowa1	0.1461	0.0425	0.0445

a bilingual dictionary for translation as well as for word segmentation in the case of Mandarin. While the system performed above average of those participating in the true bilingual task, in the translated-monolingual[5] task it performed worse than expected. We attribute the poorer than expected results to the difficulty in determining the optimal system threshold but not to the metric's capacity to separate on-topic from off topic stories.

2.6 Topic Tracking in TDT3

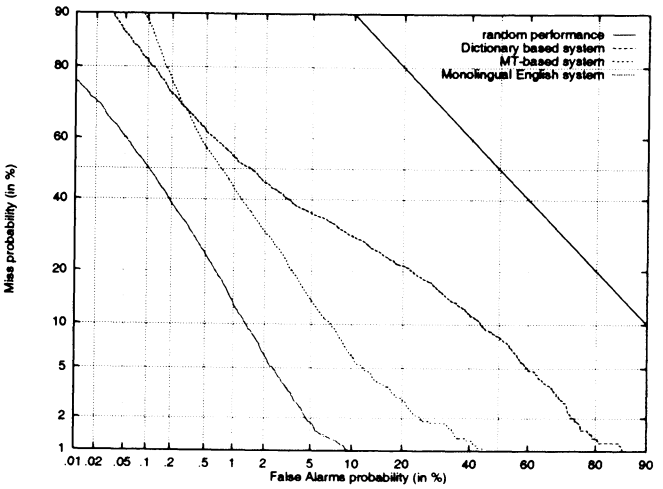
In addition to the cross-lingual nature of TDT3, there were a number of other changes in the task definition for tracking[6]. The most substantive change was that no list of off-topic training stories was provided. However, we had already decided for TDT2 to ignore the provided list and rely solely on a independent retrospective corpus for off-topic material. Other changes, which for the most part only affected the relative operating point of the systems, were the decision to the use the topic-weighted score exclusively as the benchmark for system performance (as opposed to story-weighted) and the change to the cost of a false alarm to 0.1 from 1.0 in the cost function. In addition, the cost function was normalized in TDT3 so that a normalized cost of less than one is achieved only when information is extracted from the source data.

$$C_{norm} = C_{track}/min(C_{miss} \cdot P_{target}, C_{fa} \cdot P_{non-target})$$

The objective of translingual tracking is to identify stories about a particular topic in a target language, given a set of training stories in a source language. As a baseline against which to measure system performance, a corpus was provided with target stories already translated into the source language. This

[5]Here the Mandarin text is first translated into English using an MT system.
[6]See [2] for a complete description of TDT3.

Figure 11.3. Comparison of two translingual tracking approaches to monolingual tracking. (trk-SR=nwt+bnasr)



makes it possible to run TDT2 systems over the TDT3 data without any modifications. Although the results from these baseline tests could be submitted as official results, we chose to concentrate on a self-contained system which incorporates the translation aspect of the task using only a bilingual dictionary. The advantage of this approach is that it is more easily applied to other languages than one dependent on a full-blown translation system. However, as the DET curves in Figure 11.3 show, it is difficult to approach the performance of the MT based method. The tracking tests represented by the three curves shown all used English as the training (source) language. The worst curve represents the performance of our dictionary based system where the text under test is native Mandarin. Next comes MT based approach where the Mandarin test stories are first translated into English using the MT system. Finally, the last curve represents the best we can probably expect of the translingual tracker. Here the test stories are native English text from the same time period (This is monolingual English tracking.)

2.7 Training Data in the Target Language

Given our decision to incorporate translation into the tracking task, an obvious approach to generating a topic vector in the target language is to simply translate each term in the source vector using a dictionary. We found that a much better approach was to *search* for training stories in the target language during the time period spanned by the training stories of the source language,

and then use those stories to generate a topic vector. The advantages of the latter approach are that terms not in the translation dictionary make it into the target vector and also the term counts reflect the native text. The search-based algorithm decreased the system cost by almost half as shown in Table 11.3.

We searched for training stories among the target language by first translating the source training stories, term for term, into a single large target query. We then $tf{\cdot}idf$ ranked a set of target stories from a time period corresponding to the first and last training story, our logic being that if there are stories to be found in the target language, they should appear during the same time period as those in the source language. From the sorted list, we arbitrarily chose the top ten stories and the query itself to be used as training. At this point we are now in the position to use the monolingual approach of TDT2 over the target language. However, since we track in the target language it is necessary to determine a optimal score threshold for that language. We used the training and development-test portion of the TDT2 corpus to determine a threshold for English and one for Mandarin using the method described in [8] this time optimizing only the topic-weighted cost.

2.8 Word Segmentation in Mandarin

Another aspect of the translingual task, this one particular to Mandarin is word segmentation. Since word boundaries are not explicit in Mandarin text, collecting term statistics based on words is not straight forward. However, on average, word size is approximately 2 characters so collecting overlapping bi-grams is a reasonable approximation to true segmentation. Our segmentation scheme looks for a dictionary entry beginning at the current character of the source text, if an entry is found we segment accordingly. If no entry is found, we create a bi-gram using this and the next character and advance one character in the text. We found using bi-grams where there was no coverage by the dictionary to be more effective than uni-grams in the training data but only slightly more effective in the evaluation data, as is shown in Table 11.3.

3. The "Universal Dictionary" experiment

In order to select the terms for a "Universal Dictionary", we designed an experiment to investigate the the tradeoff between tracking cost and vocabulary size for a given metric of term selection. Understanding the relationship between these two parameters will make it possible to build the smallest possible dictionary for a desired level of tracking performance.

We began by creating a large dictionary of general English terms using a corpus unrelated to our test corpus[7]. To insure that our approach is not biased

[7]Our test corpus was the TDT2 evaluation corpus [3] containing 24 test topics

Table 11.3. Comparison of algorithms over Mandarin only portion of SR=nwt+bnasr TR=eng,nat TE=mul,nat boundary Nt=4

Algorithm	cost
segmentation: dictionary/bi-grams training: translation-based	0.6145
segmentation: dictionary/uni-grams training: search-based	0.3772
segmentation: dictionary/bi-grams training: search-based [a]	0.3530

[a] Used in evaluation system

Table 11.4. Normalized tracking cost by site for *SR=nwt+bnasr, boundary Nt=4*

Site	TR=eng,eng TE=mul,eng	TR=eng,nat TE=mul,nat
BBN1	0.0922	0.1057
CMU1	0.1376	–
Dragon1	0.1596	–
GE1	0.3778	–
UIowa1	–	0.6051
UMd1	–	0.9662
UPenn1	0.2390 [a]	0.2575

[a] Unofficial

Table 11.5. Cost comparison for system threshold (predicted) vs. optimal threshold (post-hoc) for SR=nwt+bnasr, boundary Nt=4

Evaluation Condition	system threshold	optimal threshold
TR=eng,eng TE=mul,eng	0.2390	0.1539
TR=eng,nat TE=mul,nat	0.2575	0.1936
TR=man,nat TE=mul,nat	0.2149	0.1526
TR=mul,nat TE=mul,nat	0.1751	0.1191

toward the time period of the topics, spring of 1998, we chose one half of the 1997 Corpus of North American News [5]. This consists of approximately 250K news stories from two news sources. Using white-space tokenization and without stemming, we collected approximately 300K unique word-forms from these stories.

Next we modified our TDT2 monolingual tracker so that after collecting the terms from the training stories, we remove those not found in the candidate dictionary under test. A topic vector of the 50 most frequent remaining terms

was then used for tracking the topic in the same way described in [8]. Our system allowed us to modify the sort criterion and the size of the "universal dictionary" before each run over the test data.

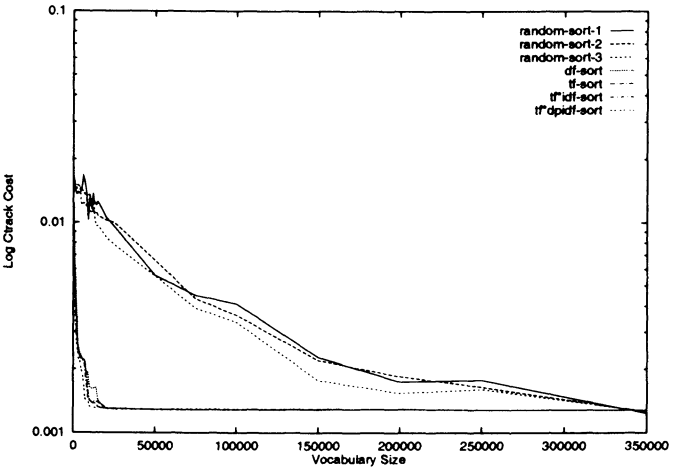
We varied dictionary size from approximately 300 thousand terms down to 100 terms for sorts based on tf, df, $tf{\cdot}idf$, $tf{\cdot}dpidf$ (to be explained shortly) and, to provide a frame of reference, three random sorts based on different seeds. Figure 11.4 shows the results of these experiments for the entire range of vocabulary sizes and Figure 11.5 shows detail for less than 20 thousand terms. In contrast to the actual tracking task where the output is a yes-no decision for each story based on its score relative to a predetermined threshold, here we are interested in only the score itself. The topic-weighted Ctrack cost plotted in these curves represents the theoretical minimum cost of our tracker for a given vocabulary size. Thus Ctrack cost serves our purposes for producing a single value that expresses the performance of vocabulary.

All of the statistics on which the sorts were based come from the North American News Corpus. The first three are well known and obvious first choices for feature selection: term frequency (how many times the word-form occurred), document frequency (how many documents the word-form occurred in), and term frequency weighted by inverse document frequency. The forth and most effective of the sorts is based on term-frequency weighted by the *difference* between a Poisson prediction of idf (based on tf) and the actual inverse document frequency. Church and Gale showed in [1] that good keywords for the purposes of IR and categorization tasks often have distributions which differ more from a Poisson-based expectation than poor ones. Effective keywords tend to "bunch up" in fewer documents than would be expected by a random distribution based on term frequency and the total number of documents. We take Gale and Church's result one step further here by using the difference from Poisson as a weighting for tf for our feature selection.

As figure 11.5 shows, the advantage of the difference-to-Poisson sort naturally falls off as vocabulary size increases, until, at about 14K terms, it meets with the curves of the other sorts. Table 11.6 shows the precentual increase in Ctrack cost over an unconstrained vocabulary for the best two sorting metrics. A vocabulary of only 10 thousand terms comes within 8% of the unconstrained vocabulary for the $tf{\cdot}dpidf$ sort. Increasing the vocabulary size to 300K only reduces the increased cost to around 4%.

We examined the 1K vocabularies of the $tf{\cdot}dpidf$ and $tf{\cdot}idf$ based dictionaries and found that of 1000 terms almost 20% (193) differ. In general the quality of the $tf{\cdot}dpidf$ keywords are superior to those of the $tf{\cdot}idf$ sort in the way described by Gale and Church. For example, of the 193 differing terms the $tf{\cdot}dpidf$ dictionary contained about 80 very specific proper nouns whereas the $tf{\cdot}idf$ dictionary contained only about 10 very generic proper nouns (e.g. *ABC, AIDS, Albright, Argentina* vs. *Bob, Calif, February, George*). These

Figure 11.4. Minumum topic-weighted Ctrack cost vs. vocabulary size for various sorting metrics (overview)



proper nouns clearly play an important role in the identification of specific topic areas. Moreover, the more than 100 remaining $tf \cdot dpidf$ terms were of much better quality as well (e.g. *accumulate, advertising, aircraft, airline* vs. *able, act, add, ahead*).

Table 11.6. Percent increase in Ctrack cost over unconstrained vocabulary.

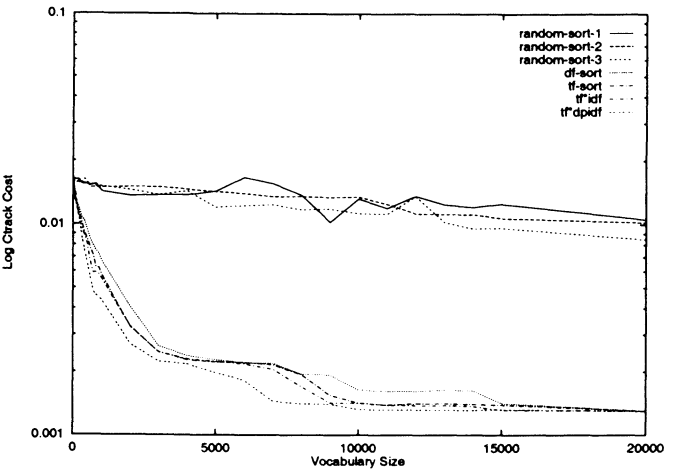
Vocabulary size	$tf \cdot dpidf$	$tf \cdot idf$
1K	254	347
5K	60.4	81.8
7K	17.8	66.9
10K	7.7	15.2
20K	5.4	5.6
300K	4.3	4.3

4. Conclusions and Directions for Future Work

Previous work, by ourselves and others, suggests that the penalty for mixed-language document sets in "topic tracking" is no more than about 30% in the TDT cost metric. The new experiment reported here shows that a set of less than 10K words has comparable performance, in the monolingual case, to a full vocabulary of 350K words.

The obvious next step is to combine these two results, and show that a small-vocabulary translation dictionary will allow the mixed-language case to

Figure 11.5. Minimum topic-weighted Ctrack cost vs. vocabulary size for various sorting metrics (detail)



approach monolingual performance. We doubt that simply reducing the vocabulary of our current English/Mandarin system will be a suitable test, because its translation dictionary is of such poor quality. However, the experiment should be tried. A better experiment would be to produce a good-quality translation dictionary for the 7K vocabulary based on the $tf \cdot dpidf$ metric, and test it. We plan to do this for a mock-TDT2 experiment in German, a language for which we have a good bilingual dictionary; we may also try to commission a Mandarin translation dictionary of this size.

Other obvious experiments include testing other term-selection metrics, such as mutual information between words and documents; and investigating the effects of treating proper names separately, as names can often be recognized and transliterated dynamically, rather than being stored in a pre-determined list.

References

[1] Kenneth W. Church, William A. Gale, "Inverse Document Frequency (IDF): A Measure of Deviations from Poisson," *Third Workshop on Very Large Corpora*, 1995.

[2] G. Doddington, "The 1999 Topic Detection and Tracking (TDT3) Task Definition and Evaluation Plan" *Available at http://www.nist/gov*, 1999.

[3] G. Doddington, "The Topic Detection and Tracking Phase 2 (TDT2) Evaluation Plan," *Available at http://www.nist.gov/speech/tdt_98.htm*, 1998.

[4] G. Doddington, "The TDT Pilot Study Corpus Documentation," *Available at http://www.ldc.upenn.edu/TDT/Pilot/TDT.Study.Corpus.v1.3.ps*, 1997.

[5] Linguistic Data Consortium, "North American News Text Corpus (Suppliment) and AP Worldstream English," *Available from http://www.ldc.upenn.edu*, 1997.

[6] A. Martin, G. Doddington, T. Kamm, M. Ordowski, M. Przybocki, "The DET Curve in Assessment of Detection Task Performance," *EuroSpeech 1997 Proceedings Volume 4*, 1997.

[7] G. Salton and M.J. McGill, "Introduction to Modern Information Retrieval," *McGraw Hill Book Co.*, New York, 1983.

[8] J. Michael Schultz, Mark Liberman, "Topic Detection and Tracking using idf-weighted Cosine Coefficient," *DARPA Broadcast News Workshop Proceedings*, 1999.

Chapter 12

An NLP & IR Approach to Topic Detection

Hsin-Hsi Chen and Lun-Wei Ku
Department of Computer Science and Information Engineering
National Taiwan University
Taipei, TAIWAN

Abstract This paper presents algorithms for Chinese and English-Chinese topic detection. Named entities, other nouns and verbs are cue patterns to relate news stories describing the same event. Lexical translation and name transliteration resolve lexical differences between English and Chinese. A two-threshold scheme determines relevance (irrelevance) between a news story and a topic cluster. Lookahead information deals with ambiguous cases in clustering. The least-recently-used removal strategy models the time factor in such a way that older and unimportant terms will have no effect on clustering. Experimental results show that nouns and verbs as well as the least-recently-used removal strategy outperform other models. The performance of the named-entity-only approach decreases slightly, but it has no overhead of nouns-and-verbs approach with the least-recently-used removal strategy.

1. Introduction

News is an important information source. The dissemination power of the Internet enables people to get information more quickly and conveniently than before. Multilingual news stories are reported anytime and anywhere, and are disseminated across geographic barriers. Detecting the occurrences of new events and tracking the processes of the events are important for decision-making in this fast-changing network era. The aim of Topic Detection and Tracking (TDT) is to study and measure related technologies. It began with TDT 1997 (TDT Pilot Study) and continued through TDT 1998 to TDT 1999. In the initial TDT study, the notion of a topic was limited to be that of an event, which means something that happened at some specific time

and place. TDT 1998 and TDT 1999 broadened the definition of a topic to include other events and activities that are directly related to it.

In the Pilot Study, there were only three tasks, i.e., the segmentation task, the tracking task and the detection task. In TDT 1999, the number of tasks was increased to five, including the story segmentation task, the topic tracking task, the topic detection task, the first-story detection task, and the link detection task. This paper will focus on the topic detection task, which aims to detect and track topics not previously known to the system. Topic detection can be formulated as a clustering problem, in which the topic set is unknown before clustering. How many news stories can be used to determine topic clusters, which cue patterns can be employed to relate news stories, and how the time factor affects clustering performance are important issues. Yang, Pierce, and Carbonell (1998) proposed retrospective topic detection. They used the entire news collection to make detection decisions. Instead of browsing the complete collection, on-line topic detection (Allan, Papka and Lavrenko, 1998; Yang, Pierce and Carbonell, 1998) looks ahead only a few news stories. It can be adapted to fit the behaviour of the Web easily. Zamir and Etzioni (1998) proposed an incremental clustering method for use on the Web.

The materials that we will study in this paper cover Chinese and English news stories. English/Chinese monolingual topic detection and English-Chinese multilingual topic detection will be discussed. Multilingual topic detection is more challenging than the monolingual problem because it has to resolve linguistic differences. This paper will adopt natural language processing (NLP) and information retrieval (IR) technologies to deal with this problem. Some of our previous experiences with cross-language information retrieval (Bian and Chen, 2000; Chen, Bian and Lin, 1999; Chen, Huang, Din and Tsai, 1998) and multi-document (multilingual) summarization (Chen and Huang, 1999; Chen and Lin, 2000) will be employed. The basic idea is to capture the similarities between different granularities of objects in different languages, e.g., query and document, document and document, and meaningful-unit and meaningful-unit (e.g., a sentence). That is also important for topic detection. Machine translation (transliteration) can resolve lexical differences among multilingual news stories. Clustering and other techniques can be used to identify new topics and track subsequent directly related events.

This paper is organized as follows. Section 2 presents a general system framework. A language independent topic detection algorithm is proposed. Language dependent issues are discussed in Sections 3-5, including how to represent news stories and topics under consideration of the time-variance factor, how to measure the similarity between monolingual news stories and topics, how to interpret thresholds used in the algorithm, and how to extend

the system framework to the multilingual case. Section 6 presents the development experiments, in which an augmented version of the TDT-2 corpus was used. Sections 7 and 8 report and discuss the evaluation results of topic detection obtained using the TDT-3 corpus. Finally, Section 9 concludes the paper.

2. General System Framework

Multilingual topic detection deals with English and Mandarin news materials in the TDT 1999 project. Chinese is different from English in several ways. The lack of word boundaries in Chinese sentences is one of the major differences. Word segmentation (Chen and Lee, 1996; Sproat, *et al.*, 1994), which tries to identify words in a Chinese sentence, is indispensable for automatic topic detection in Chinese. Word segmentation is also necessary to translate Chinese words into English ones in multilingual topic detection. In the general system framework, we will touch on language independent issues only. The specific features of each language and extension to multilingual detection will be discussed in detail in the subsequent sections.

Given a sequence of news stories, the topic detection task involves detecting and tracking topics not previously known to the system. The algorithm proceeds as follows. Initially, the first news story d_1 is assigned to topic t_1. Assume there already are k topics when a new article d_i is considered. That is, topics $t_1, t_2, ..., t_k$ ($k < i$) have been detected. News story d_i may belong to one of k topics, or it may form a new topic t_{k+1}. That is determined by the similarity measure defined below. Assume that a news story d is represented as a term vector V_d. Similarly, a topic t is represented as a term vector V_t. The similarity S_{td} between news story d and topic t is described below.

$$S_{td} = \frac{V_t \bullet V_d}{|V_t||V_d|}$$

Two thresholds, a low threshold (TH_l) and a high threshold (TH_h), are specified. Their interpretations are given below:

if $S_{td} < TH_l$, then news story d is irrelevant to topic t;

if $S_{td} \geq TH_h$, then news story d is relevant to topic t;

if $TH_l \leq S_{td} < TH_h$, then the relationship between d and t is undecidable.

Consider incoming news story d_i. If there exists a topic t_k such that d_i is relevant to t_k, then we say that the news story touches on an old event. We select a topic, say t_k, with the highest similarity with d_i, and insert d_i into t_k. The term vector V_{tk} is changed accordingly. In contrast with this case, if d_i is

irrelevant to all the topics, then we say that the news story deals with a new event. If the similarity is not above the high threshold or below the low threshold, then it is not decidable at this stage. In our algorithm, we should consider the deferral period DEF for detection decisions. The next DEF (e.g., 10) news stories are examined further. Some of these lookahead news stories may belong to topics t_1, ..., t_k. Recall that the term vectors for the corresponding topics may be changed when such news stories are inserted. In such a situation, the similarity measure between the changed topic and the undecidable news story may be above the high threshold or below the low threshold. In other words, relevance or irrelevance may be decided after the next DEF news stories are read. If it is still undecidable, we change the decision procedure in the following way. Here, TH_m is equal to $(TH_l + TH_h)/2$:

if $TH_l \leq$ new $S_{td} < TH_m$, then d is irrelevant to t;
if $T_m \leq$ new $S_{td} < TH_h$, then d is relevant to t.

For each decision, the system has to output a "confidence value" to indicate how confident it is. The conference value C_f of the detection decision is defined as follows:

(1) news story d is relevant to topic t
$C_f = (S_{td} - TH_m)/(1 - TH_m)$;
(2) news story d is irrelevant to topic t
$C_f = (TH_m - S_{td})/TH_m$.

This algorithm is repeated until all the news stories are considered. In this algorithm, several issues have to be discussed further.

(1) How can a news story and a topic be represented?
(2) How can the similarity between a news story and a topic be calculated?
(3) How can the two thresholds, i.e., TH_l and TH_h, be interpreted?
(4) How can the system framework be extended to multilingual case?

3. Representation of News Stories and Topics

3.1 Term Vectors for News Stories

Because Chinese sentences are composed of characters without any word boundaries, a word segmentation system is employed (Chen, Ding and Tsai, 1998). Besides word segmentation, this system also extracts useful named entities such as named organizations, people, and locations, along with date/time expressions and monetary and percentage expressions. In the

formal run of MET-2[1] (Chen, Ding, Tsai and Bian, 1998), the F-measures P&R, 2P&R and P&2R for extraction of Chinese named entities are 79.61%, 77.88% and 81.42%, respectively. For English materials, we only employ some simple heuristic rules to recognize named entities. For example, continuous capitalized words are regarded as named entities.

A news story is represented as a term vector. Because the backbone of a sentence is a predicate-argument structure, verbs and nouns are selected as candidate terms. In addition, people, affairs, time, places and things are five basic entities in a document. When we find the fundamental entities, we can understand a document to some degree. Therefore, named entities and other nouns are differentiated. To extract nouns and verbs, news stories are tagged. The ApplePie tagger, version 5.9, and a Chinese tagger are employed to English and Mandarin materials, respectively.

A formula modified from the traditional tf*idf expression is employed to measure the weight of each candidate term. Assume that there already are n topics when a new article d_i is considered in the detection algorithm. The weight w_{ij} of a candidate term f_j in d_i is computed as follows:

$$w_{ij} = \ln(tf_{ij}) \times idf_j$$

$$idf_j = \ln(\frac{n}{n_j})$$

where tf_{ij} is the number of occurrences of f_j in d_i,

n is the total number of topics that the system has detected, and

n_j is the number of topics in which f_j occurs.

The candidate terms are sorted by weight. The first N (e.g., 50) terms are selected and form a vector for a news story.

3.2 Term Vectors for Topics

During topic detection, a topic is composed of news stories that have been detected up to now. These news stories form a cluster. We will select a term vector to represent the common characteristics of the news stories in the cluster. Because the event changes with time, the time-variance issue has to be addressed when the vector is determined. Consider the detection algorithm. For an incoming news story d_i, assume that topic t_k has the highest similarity with d_i, and that the similarity is above TH_h. That is, d_i is related to

[1] MET-2 is the second multilingual entity task evaluation, which is run in conjunction with MUC-7 (the seventh message understanding competition). Systems are evaluated using recall (R) and precision (P) metrics, and the F-measure.

t_k and will be inserted into the cluster for t_k. The term vector V_{tk} is changed in the following way. Two strategies shown below may be adopted.

(1) Top-N-Weighted strategy
This strategy is simple. We select N terms with larger weights from the current V_{tk} and V_{di}. These terms form a new vector for topic t_k.

(2) LRU+Weighting strategy
This strategy is more complex. We keep M candidate terms for each topic. For consideration of the time-variance issue, each candidate term is tagged with the time label of the latest news story containing the term. When d_i is inserted, the number of candidate terms for t_k may be larger than M. In this case, the list for the candidate terms is full. The least-recently-used terms will be replaced. In other words, the terms in V_{di} are inserted into t_k, and N older candidate terms with lower weights are deleted. This means that we always keep the more important terms and the latest terms in each topic cluster. Thus, both recency and weight are incorporated.

Besides the fact that older candidate terms may be replaced by newer terms, the terms' weights can also be changed dynamically. The updating process deals with the phenomenon that there may be sub-events directly related to a topic. If an event lasts longer, it covers more and more sub-events. Therefore, the term vector of a topic has to timely reflect the change of a topic. In our algorithm, we remember the number of topics in which a term occurs. Whenever we insert a news story into an existing topic or create a new topic, the weight of each term is recomputed. As time goes by, the importance of terms will be increased or decreased.

4. Similarity and Interpretation of a Two-Threshold Method

In the detection algorithm, the cosine function is used to measure the similarity between a news story and a topic. However, exact term matching is not enough because the same concept in different news stories can be specified by different terms. The query expansion approach in information retrieval is adopted.

High (TH_h) and low (TH_l) thresholds indicate if the similarity can reflect the relationship between a news story and a topic. The relationship can be strong (i.e., similarity $\geq TH_h$), weak (i.e., similarity $< TH_l$), or unknown ($TH_l \leq$ similarity $< TH_h$). For an unknown relationship, we can use lookahead. Because news events generally appear during a certain short period of time, the news stories that are looked ahead have a high probability of belonging to the same topic as the current news. DEF defines how many news stories a

system can look ahead. Figures 1 and 2 show the basic concept behind our two-threshold method. Here, we use distance to interpret similarity. The higher the similarity two objects have, the smaller their distance is.

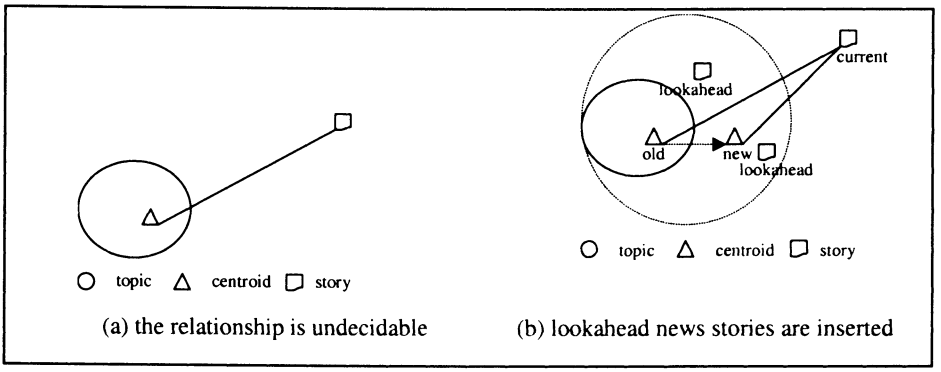


Figure 1. Relationship from undecidable to relevant

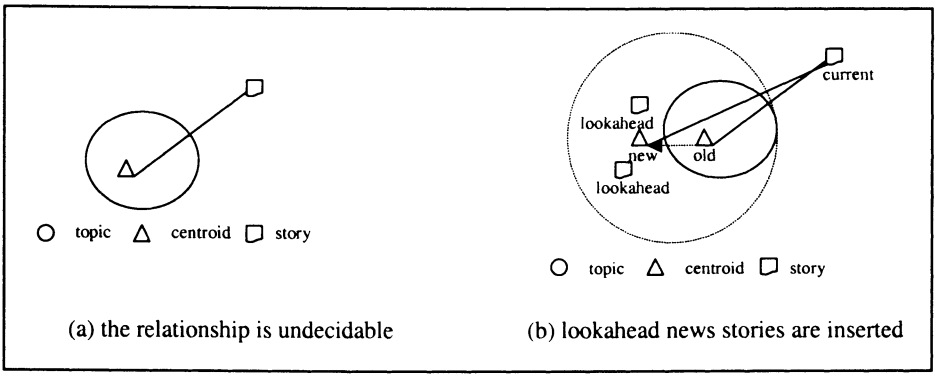


Figure 2. Relationship from undecidable to irrelevant

Figures 1(a) and 2(a) show that the detection decision is undecidable for a news article in the first phase, and that the relevant news stories in the lookahead part may affect the decision in the second phase. The circle denotes the cluster created so far, and the small triangle denotes its centroid. Figure 1(b) shows that the relevant news stories in the lookahead part are close to the current news, so that they act as a bridge to bring the current news story and the new topic centroid closer. On the other hand, Figure 2(b) shows that the relevant news stories move the topic centroid farther away from the current news story. In this situation, the current news story may be inserted into another cluster, or a new topic cluster may be created.

The area between the high and low thresholds is called the ambiguous area. There are two ways to adjust the thresholds. One is to adjust the high and low

thresholds only and keep the ambiguous area the same (i.e., shift these values perpendicularly). The other is to change the ambiguous area also (i.e., enlarge or shorten it). Recall that a news story whose similarity score falls in the ambiguous area will be left un-judged until all the lookahead news stories are read. If the ambiguous area is enlarged, the probability of undecidability increases, and lookahead is important. On the other hand, if the ambiguous area is shortened, then most of the decisions are made immediately, so that much of the lookahead information will not be used. The side effect of choosing an area is efficiency. In Section 6, we will show the performance under different assignments.

5. Multilingual Topic Detection

5.1 Lexical Translation

The topic detection task is that of identifying the news stories that describe a seminal event or activity. Because the news stories are in English or Chinese, some kind of translation is required to capture the lexical differences. Besides the problem of translation ambiguity, different news stories may use different names to refer to the same entity. The translation of named entities, which are usually unknown words, is another problem.

Given a sequence of news stories in English or Chinese, we can unify the language usage by means of a machine translation system. For example, we can translate all the English news stories into Chinese, and vice versa. TDT 1999 provides such a set of English materials translated from the original Chinese materials by the Dragon and Systran systems. An English topic detection algorithm can be applied directly to the unified news stories. Because the structural information of a sentence is not used by the detection algorithm, lexical transfer in machine translation affects the detection performance more than structural transfer does. Thus, this paper will focus on lexical translation only.

To explore the difficulties involved in lexical translation of different languages, a previous study (Chen, Bian and Lin, 1999) gathered the sense statistics of English and Chinese words. That paper reported that Chinese is comparatively unambiguous compared to English. In this study, the Chinese thesaurus tong2yi4ci2ci2lin2 (Mei, *et al.*, 1982) and Roget's thesaurus were used to count the statistics of the senses of words. On average, an English word has 1.687 senses, and a Chinese word has 1.397 senses. If the top 1000 high frequency words are considered, then the English words have 3.527 senses, and the bi-character Chinese words have only 1.504 senses. In this study, Mandarin news stories were translated into English ones.

If a Chinese word has more than one English translation, then we disambiguate its use based on the contextual information. For Chinese named entities not listed in the lexicon, name transliteration similar to the algorithm of (Chen, Huang, Ding, and Tsai, 1998) is introduced for matching of non-alphabetic (e.g., Chinese) and alphabetic (e.g., English) languages. The lexical selection algorithm is presented first, followed by the name transliteration algorithm. We follow the work of Bian and Chen (2000), who combined the dictionary-based and corpus-based approaches for lexical translation. A Chinese-English bilingual dictionary provides the translation equivalents of each term. The bilingual dictionary is integrated based on four resources, including the LDC Chinese-English dictionary[2], Denisowski's CEDICT[3], the BDC Chinese-English dictionary v2.2 and a dictionary used in query translation in the MTIR project (Bian and Chen, 2000). The dictionary contains 200,037 words, where a word may have more than one translation.

The word co-occurrence information trained from a target language text corpus is used to disambiguate the translation. The TREC-6 text collection, which contains 556,077 documents and is about 2.2G bytes in size, is employed to compute the co-occurrence statistics. This method uses the context around the translation equivalents to decide the best target word. The translation of a term can be disambiguated using the co-occurrence of the translation equivalents of this term and other terms. Mutual information (Church, *et al.*, 1989) is adopted to measure the co-occurrence strength. The mutual information $MI(x,y)$ is defined as follows:

$$MI(x, y) = \log_2 \frac{p(x, y)}{p(x)p(y)}$$

where x and y are words,
$p(x)$ and $p(y)$ are probabilities of words x and y, and
$p(x,y)$ is their co-occurrence probability.

When a source sentence is translated, the basic terms are recognized first, and then the translation equivalents of each term are retrieved from a bilingual dictionary. After that, MI values are used to select the best translation equivalent. The selection step is carried out based on the order of the terms. For a term, we compare the MI values of all the translation equivalent pairs (x, y), where x is the translation equivalent of this term, and y is the translation equivalent of the other terms within a sentence. The word pair (x_i, y_j) with the highest MI value is extracted and the translation equivalent x_i is regarded as the best translation equivalent of this term. All the selections form the final translation.

---

[2] The LDC term list is available at http://morph.ldc.upenn.edu/Projects/Chinese/.

[3] The Denisowski's CEDICT is available at http://ftp.cc.monash.edu.au/pub/nihongo/.

5.2 Name Transliteration

The name transliteration algorithm deals with the translation of named organizations, people, and locations. Because proper names are usually unknown words, it is hard to find them in a monolingual dictionary, not to mention in a bilingual dictionary. Compared with disambiguation in the above lexical selection algorithm, how to map non-alphabetic characters into alphabetic ones is the major concern in name transliteration. Machine transliteration can be classified into four types based on the transliteration direction and character sets (Lin and Chen, 2000). Chinese-English machine transliteration is a backward transliteration between different character sets.

There are two approaches to backward transliteration from Chinese to English, i.e., the grapheme-based (Chen, Huang, Ding and Tsai, 1998) and phoneme-based (Lin and Chen, 2000) approaches. The grapheme-based approach romanizes a Chinese proper name in using the Pinyin and Wade-Giles systems, and then computes the similarity between the romanized proper name and English candidates using a scoring function. The English candidates come from the vector for a news story or a topic cluster (refer to Section 5.4). The scoring function is defined as follows:

$$score = \sum_i \frac{f_i \times (el_i / (2 \times cl_i) + 0.5) + o_i \times 0.5}{el}$$

where el denotes the length of the English proper name,

el_i denotes the length of syllable i in the English proper name,

cl_i denotes the number of Chinese characters corresponding to syllable i,

f_i denotes the number of matching first-letters in syllable i,

and o_i denotes the number of matching other letters in syllable i.

This formula considers the lengths of matching characters within syllables. For example, the Wade-Giles romanization of the Chinese proper name '埃斯其勒斯' is 'ai.ssu.chi.le.ssu'. The corresponding English proper name is Aeschylus. The similarity based on this scoring function is as follows. The first letter is in upper case.

aes	chy	lus	(English proper name)
AiSsu	Chi	LeSsu	(Chinese romanization)

The corresponding parameters are listed below. The matching score is 0.83.

el_1=3, cl_1=2, f_1=2, o_1=0, el=9,
el_2=3, cl_2=1, f_2=1, o_2=1,

el_3=3, cl_3=2, f_3=2, o_3=0.

We also add pronunciation rules in ranking. For example, *ph* usually has the *f* sound.

In the phoneme-based approach, both Chinese proper names and English proper names are mapped into phonemes in using the International Phonetic Alphabet (IPA). The similarity between two proper names is computed on the phoneme level instead of the grapheme level. For example, the Chinese name "亞瑟" and the corresponding English name "Arthur" are mapped into "IY AA S r" and "AA R TH ER", respectively. Dynamic programming will find the best alignment, shown below, and a similarity score will be computed:

亞瑟	IY	AA	_	S	r
Arthur	_	AA	R	TH	ER

The experiments conducted by Lin and Chen (2000) showed that the phoneme-based approach outperforms the grapheme-based approach. In mate matching of 1,261 candidates, the average rank was 7.80, and 57.65% of the candidates were ranked as number one. We adopt the phoneme-based approach to deal with the machine transliteration problem in English-Chinese multilingual topic detection.

5.3 Representation of Multilingual News Stories and Topics

In the multilingual topic detection model, each Mandarin news story is translated into an English one. This is done before term vectors are generated because contextual information can be used. Representation of news stories and topics in multilingual topic detection is a little difference from that in monolingual topic detection. In Mandarin news stories, a vector is composed of term pairs (Chinese-term, English-term), where the English-term is translated from the Chinese-term by the above translation (transliteration) algorithm. For English news stories, a vector is composed of term pairs (nil, English-term). Here, we will not translate English terms into Chinese. The topic detection algorithm is used as described above. A seminal event and the directly related events form a cluster for a topic incrementally. The representation of a multilingual topic is similar to that of a monolingual topic except that there is an English version (either translated or native) for each candidate term.

5.4 Similarity Measure

The similarity computation between a news story and a topic is modified as follows. Assume that the next news story to be considered is d_i, and that the topic compared is t_k. The news d_i may be in Chinese or in English, and there already are Mandarin and English stories in t_k. Consider the following two cases:

(1) d_i is a Mandarin news story represented as $<(c_{i1},e_{i1}), (c_{i2},e_{i2}), \ldots, (c_{iN},e_{iN})>$. We use c_{ij} $(1 \le j \le N)$ to match the Chinese terms in V_{tk}, and use e_{ij} $(1 \le j \le N)$ to match the English terms. Query expansion is done if necessary.

(2) d_i is an English news story represented as $<(nil,e_{i1}), (nil,e_{i2}), \ldots, (nil,e_{iN})>$. We use e_{ij} $(1 \le j \le N)$ to match the English terms in V_{tk}, and English translation to match the Chinese terms. Query expansion is done if necessary.

In query expansion, if a term in a news vector cannot be matched to any terms in a topic vector, we expand it with synonyms from a thesaurus like tong2yi4ci2ci2lin2, abbreviated as Cilin (Mei, *et al.*, 1982) in Chinese or WordNet (Fellbaum, 1998) in English. The weight of an expanded term is half of the weight of the original term.

Cilin is composed of 12 large categories, 94 middle categories, 1,428 small categories, and 3,925 word clusters. Table 1 shows large and middle categories. Word clusters are adopted for expansion. There are no named entities or phrases in Cilin. WordNet, an electronic English lexical database, has been widely applied to different problems (Harabagiu, 1998; Rila, 1998; Ruiz, *et al.*, 1999), such as information retrieval, lexical acquisition, natural language generation, word sense disambiguation, and so on. In WordNet, Synset, which is a set of synonyms, is a basic block. Given a synset, we add the synonyms within the synset to a vector for expansion. WordNet 1.6 was adopted in our experiments.

Table 1. Semantic Structures of Cilin

A. PERSON (人): **Aa**. general name (泛稱), **Ab**. people of all ages and both sexes (男女老少), **Ac**. posture (體態), **Ad**. nationality/citizenship (籍屬), **Ae**. occupation (職業), **Af**. identity (身分), **Ag**. situation (狀況), **Ah**. relative/family dependents (親人/眷屬), **Ai**. rank in the family (輩次), **Aj**. relationship (關係), **Ak**. morality (品行), **Al**. ability and insight (才識), **Am**. religion (信仰), **An**. comic/clown type (丑類)
B. THING (物): **Ba**. generally called (統稱), **Bb**. imitate form (擬狀物), **Bc**. part of an object (物體的部分), **Bd**. a celestial body (天體), **Be**. terrian features (地貌), **Bf**. meteorological phonomena (氣象), **Bg**. natural substance (自然物), **Bh**. plant (植物), **Bi**. animals (動物), **Bj**. micro-organism (微生物), **Bk**. the whole body (全身), **Bl**. secretions/excretions (排泄物/分泌物), **Bm**. material (材料), **Bn**. building (建築物), **Bo**. machines and tools (機具), **Bp**. appliances (用品), **Bq**. clothing (衣物), **Br**. edibles/medicines/drugs (食品/藥物/毒品)
C. TIME AND SPACE (時間與空間): **Ca**. time (時間), **Cb**. space (空間)
D. ABSTRACT THINGS (抽象事物): **Da**. event/circumstances (事情/情況), **Db**. reason/logic (事理), **Dc**. looks (外貌), **Dd**. functions/properties (性能), **De**. character/ability (性格/才能), **Df**. conscious (意識), **Dg**. analogical things (比喻物), **Dh**. imaginary things (臆想物), **Di**. society/politics (社會/政法), **Dj**. economy (經濟), **Dk**. culture and education (文教), **Dl**. disease (疾病), **Dm**. organization (機構), **Dn**. quantity/unit (數量/單位)
E. CHARATERISTICS (特徵): **Ea**. external form (外形), **Eb**. surface looks/seeming (表象), **Ec**. color/taste (顏色/味道), **Ed**. property (性質), **Ee**. virtue and ability (德才), **Ef**. circumstances (境況)
F. MOTION (動作): **Fa**. motion of upper limbs (hands) (上肢動作), **Fb**. motion of lower limbs (legs) (下肢動作), **Fc**. motion of head (頭部動作), **Fd**. motion of the whole body (全身動作)
G. PSYCHOLOGICAL ACTIVITY (心理活動): **Ga**. state of mind (心理狀態), **Gb**. activity of mind (心理活動), **Gc**. capability and willingness (能/願)
H. ACTIVITY (活動): **Ha**. political activity (政治活動), **Hb**. military activity (軍事活動), **Hc**. administrative management (行政管理), **Hd**. production (生產), **He**. economic activity (經濟活動), **Hf**. communications and transportation (交通運輸), **Hg**. education and hygiene scientific research (教衛科研), **Hh**. recreational and sports activities (文體活動), **Hi**. social contact (社交), **Hj**. life (生活), **Hk**. religious activity (宗教活動), **Hl**. superstitious belief activity (迷信活動), **Hm**. public security and judicature (公安/司法), **Hn**. wicked behavior (惡行)
I. PHENOMENON AND CONDITION (現象與狀態): **Ia**. natural phenomena (自然現象), **Ib**. physiology phenomena (生理現象), **Ic**. facial expression (表情), **Id**. object status (物體狀態), **Ie**. situation (事態), **If**. circumstances (mostly unlucky) (境遇), **Ig**. the beginning and the end (始末), **Ih**. change (變化)
J. TO BE RELATED (關聯): **Ja**. association (聯繫), **Jb**. similarities and dissimilarities (異同), **Jc**. to operate in coordination (配合), **Jd**. existence (存在), **Je**. influence (影響)
K. AUXILIARY PHRASE (助語): **Ka**. quantitative modifier (疏狀), **Kb**. preposition (中介), **Kc**. conjunction (聯接), **Kd**. auxiliary (輔助), **Ke**. interjection (呼嘆), **Kf**. onomatopoeia (擬聲)
L. GREETINGS (敬語)

6. Development Experiments

6.1 Evaluation Criteria

The evaluation program version 1.8 released by TDT 1999 was used to evaluate the system's outputs. TDT 1999 evaluates the performance with miss and false alarm instead of recall and precision. Table 2 shows the topic contingency table.

Table 2. The Topic Contingency Table

	In topic	Not in topic
In topic (system)	(1)	(2)
Not in topic (system)	(3)	(4)

These two sets of metrics are defined as follows using this table:

(1) Miss = (3) / [(1) + (3)]
    False alarm = (2) / [(2) + (4)]
(2) Recall = (1) / [(1) + (3)]
    Precision = (1) / [(1) + (2)]

Both miss and false alarm are penalties. They can measure more accurately the behaviour of users who try to retrieve news stories. If miss or false alarm is too high, users will not be satisfied with these news stories. The detection performance is characterized by a detection cost, C_{det}, in terms of the probability of miss and false alarm:

$$C_{Det} = C_{Miss} \times P_{Miss} \times P_{target} + C_{FA} \times P_{FA} \times P_{non\text{-}target}$$

$$C_{Det}(norm) = C_{Det} / \min(C_{Miss} \times P_{target}, C_{FA} \times P_{non\text{-}target})$$

where C_{Miss} and C_{FA} are the costs of a Miss and a False Alarm, respectively;
P_{Miss} and P_{FA} are the conditional probabilities of a Miss and False Alarm;
P_{target} and $P_{non\text{-}target}$ (=1- P_{target}) are the *a priori* target probabilities.

In the following experiments, results were obtained for monolingual Mandarin newswire stories, and the formal evaluation results are given in Section 7.

6.2 Chinese Topic Detection

The Mandarin stories were selected from the Xinhua News Agency and ZaoBao News. Because the vocabulary sets and the Chinese coding sets used in Taiwan and in China are not the same, we had to transform simplified Chinese characters in the GB coding set into traditional Chinese characters in Big-5 coding set before testing. A word that is known may

become unknown due to transformation. For example, the character "晨" in "凌晨" (early morning) is a traditional Chinese character. However, "辰" is a simplified Chinese character, and it is also a legal traditional Chinese character that has another meaning. In other words, the mapping from GB to Big5 is "凌辰", which is an unknown word based on our dictionary. The different vocabulary sets between China and Taiwan also result in error segmentation. For example, "人工智慧" vs. "人工智能" (artificial intelligence), "軟體" vs. "軟件" (software), "紐西蘭" vs. "新西蘭" (New Zealand), "肯亞" vs. "肯尼亞" (Kenya), and so on.

In the experiments, we studied the effects of some important factors in the topic detection algorithm, such as the types of terms, the strategies for refining topic vectors, and the thresholds. Materials from the time period January-June 1998 (TDT-2 augmented) were used. Intuitively, named entities illustrate the key concepts of a document. At first, only named entities were considered as candidate terms in a vector, and the top-N-weighted strategy described in Section 3.2 was adopted. Table 3 shows the performance results under different thresholds. For comparison, Table 4 shows the performance results when the LRU+Weighting strategy was adopted. The performance improved in some cases and worsened in other cases, depending on the pairs of thresholds. The up arrow ↑ and the down arrow ↓ denote that the performance improved or worsened, respectively. Recall that cost should be minimized, so the up arrow ↑ means a lower cost. The performance change (%) of model *A* relative to model *B* was defined by (Cdet(norm) of model *A* – Cdet(norm) of model *B*) / Cdet(norm) of model *B*. When the change was negative, the performance improved relative to the comparative model.

Table 3. Named Entities Only & the Top-N-Weighted Strategy in Chinese Topic Detection

TH_{low}	TH_{high}	Topic-Weighted P(miss)	Topic-Weighted P(F/A)	Cdet (norm)
0	0.20	0.6809	0.0075	0.7178
0.05	0.25	0.6884	0.0102	0.7385
0.10	0.30	0.6542	0.0068	0.6877
0.15	0.35	0.6717	0.0045	0.6938
0.20	0.40	0.6716	0.0037	0.6899

Table 4. Named Entities Only & the LRU+Weighting Strategy in Chinese Topic Detection

TH_{low}	TH_{high}	P(miss)	P(F/A)	Cdet (norm)	Change (vs. Table 3)
0.	0.20	0.6546	0.0014	0.6613	7.87%↑
0.05	0.25	0.6569	0.0008	0.6607	10.53%↑
0.10	0.30	0.6732	0.0003	0.6749	1.86%↑
0.15	0.35	0.6949	0.0002	0.6957	0.27%↓
0.20	0.40	0.7591	0.0001	0.7595	10.09%↓

Table 5. Minor Changes in Thresholds

TH_{low}	TH_{high}	P(miss)	P(F/A)	Cdet (norm)	Change (vs. previous)
0.08	0.28	0.6614	0.0086	0.7038	
0.09	0.29	0.6241	0.0061	0.6539	7.09%↑
0.10	0.30	0.6542	0.0068	0.6877	5.17%↓
0.11	0.31	0.6824	0.0063	0.7131	3.69%↓
0.12	0.32	0.6449	0.0068	0.6785	4.85%↑
0.13	0.33	0.7302	0.0042	0.7505	10.61%↓
0.14	0.34	0.6793	0.0057	0.7070	6.06%↑

Table 6. Nouns-Verbs & the Top-N-Weighted Strategy in Chinese Topic Detection

TH_{low}	TH_{high}	P(miss)	P(F/A)	Cdet (norm)	Change (vs. Table 3)
0	0.20	0.9739	0.0052	0.9993	39.21%↓
0.05	0.25	0.9946	0.0004	0.9965	34.94%↓
0.10	0.30	0.8745	0.0060	0.9039	31.44%↓
0.15	0.35	0.7943	0.0015	0.8015	15.52%↓
0.20	0.40	0.8119	0.0003	0.8134	17.90%↓

Table 7. Nouns-Verbs & the LRU+Weighting Strategy in Chinese Topic Detection

TH_{low}	TH_{high}	P(miss)	P(F/A)	Cdet (norm)	Change (vs. Table 3)
0	0.20	0.5004	0.0025	0.5128	28.56%↑
0.05	0.25	0.5292	0.0015	0.5365	27.35%↑
0.10	0.30	0.6128	0.0008	0.6169	10.30%↑
0.15	0.35	0.6952	0.0003	0.6968	0.43%↓
0.20	0.40	0.7126	0.0002	0.7133	3.39%↓

We also conducted a series of experiments to examine if minor changes in thresholds would significantly affect the performance. Table 5 shows the experimental results. Named entities only and the top-N-weighted strategy were employed. When the ambiguous area shifted 0.01 within the range of the best thresholds (0.10, 0.30) as shown in Table 3, the mean was 0.6992 and the standard deviation was 0.03.

Now, we will consider the effects of verbs and other nouns. Besides named entities, N verbs and nouns with higher weights were included to enrich the term vector for a news story. Table 6 shows the performance results when the top-N-weighted strategy was adopted. The performance was worse than that in the earlier experiments. We applied the second strategy again. Table 7 lists the experimental results. The LRU+Weighting strategy was better than the top-N-weighted strategy when nouns and verbs were incorporated.

7. Evaluation

7.1 Chinese Topic Detection

The two models using named entities only, but with different strategies, achieved similar performance. When nouns and verbs were used, the LRU+Weighting strategy was better than the top-N-weighted strategy. When the top-N-weighted strategy was adopted, using named entities was better than using nouns and verbs. The situation was reversed when the LRU+Weighting strategy was used. The model using nouns and verbs, together with the LRU+Weighting strategy, exhibited the best performance. Nouns and verbs do contribute to topic detection, but they also introduce noise. The LRU+Weighting strategy can filter noise in some sense. The first strategy is suitable for features without noise. Table 8 summarizes the conclusion. The numbers denote the ranks.

Table 8. Comparisons of Term and Strategies

	Named Entities Only	Nouns and Verbs
The top-N-weighted strategy	2	3
The LRU+Weighting strategy	2	1

Finally, the data obtained in the formal run of TDT 1999, i.e., materials from the time period October-December 1998, was employed. Table 9 shows the experimental results. They agree with the above conclusion.

Table 9. Results with TDT-3 Corpus

TH_{low}	TH_{high}	Named Entities & LRU+W Cdet (norm)	Nouns-Verbs & LRU+W Cdet (norm)
0	0.20	0.5716	0.4327 (24.30%↑)
0.10	0.30	0.6166	0.4727 (23.34%↑)
0.15	0.35	0.6271	0.5610 (10.54%↑)
0.20	0.40	0.6812	0.4775 (29.90%↑)

Table 10 shows some official results of topic detection obtained on Jan 21, 2000. In the formal evaluation, the basic approach, i.e., named entities only using the top-N-weighted strategy, was adopted. Two sets of thresholds were adopted: for NTU1, TH_l=0.1 and TH_h=0.5; for NTU2, TH_l=0.2 and TH_h=0.4. Comparing NTU1 and NTU2, to enlarging the threshold period seems to have little impact. The approaches of both named entities only and nouns-verbs using the LRU+Weighting strategy (refer to Table 9) outperformed the basic approach.

Table 10. Some Official Results of Topic Detection

Evaluation Conditions	BBN	NTU1	NTU2	UMass
SR=nwt+bnasr TE=eng,nat boundary DEF=10	0.2314			0.1839
SR=nwt+bnasr TE=man,eng boundary DEF=10	0.2104			0.2472
SR=nwt+bnasr TE=man,nat boundary DEF=10	0.2490	0.7303	0.7252	0.4634
SR=nwt+bnasr TE=mul,nat boundary DEF=10	0.3417			0.4682

7.2 English-Chinese Topic Detection

The multilingual topic detection algorithm described in Section 5 was applied to English-Mandarin materials in TDT 1999. A dictionary was used for lexical translation. For name transliteration, we measured the pronunciation similarity among English and Chinese proper names. A Chinese named entity extraction algorithm was applied to extract Chinese proper names, and heuristic rules such as continuous capitalized words were used to select English proper names. In our experiment, the high threshold and the low threshold were set to 0.2 and 0.1, and the approach of nouns-and-verbs with the LRU+Weighting strategy was adopted. Table 11 shows the experimental results. The topic-weighted P(miss), topic-weighted P(F/A), and Cdet (norm) were 0.5115, 0.0034, and 0.5280, respectively.

Table 11. Performance of English-Chinese Topic Detection

type	TH_{low}	TH_{high}	P(miss)	P(F/A)	Cdet (norm)
English-Chinese	0.1	0.2	0.5115	0.0034	0.5280
Chinese	0.1	0.3	0.4673	0.0011	0.4727

Compared to Chinese topic detection, the performance of English-Chinese topic detection was a little poorer. Because tokens in news from audio in TDT-3 are all in lowercase, the impact of name transliteration is decreased. In addition, boundary errors in named entity extraction are also a problem. For example, the name "Lewinsky" can be transliterated into "陸文斯基" or "萊溫斯基" in Mandarin news stories. These terms can be segmented into "陸文", "斯基", "陸文斯", or "溫斯基" depending on the context. According to name transliteration, the scores for "斯基" ("溫斯基") and Lewinsky are very low. Consider another example. "Bill Clinton" in English corresponds to "柯林頓總統" in Chinese. The first name "Bill" is omitted in some Mandarin news stories, so the transliteration score is also low.

8. Discussion

Named entities, which denote people, places, time, events, and things, play an important role in a news story. Named entities are also nouns, so that they may be selected by the nouns-verbs approach. However, named entities may not occur as frequently as other nouns and verbs. Thus, we amplify the weights of named entities 2 and 3 times to verify whether better performance can be achieved. Table 12 lists the results obtained using the LRU+Weighting strategy.

Table 12. Named Entities with Amplifying Weights before Selecting

amplification	TH_{low}	TH_{high}	P(miss)	P(F/A)	Cdet (norm)
weight × 1	0	0.15	0.4010	0.0060	0.4304
weight × 2	0	0.15	0.4335	0.0038	0.4519
weight × 3	0	0.15	0.4559	0.0032	0.4714

Weight amplification causes the detection performance to worsen. This contradicts our expectation. One of the major reasons is that not all the named entities in a news story are important for detection. For example, the names of reporters or newspapers are noise to the detection. Only the discriminating named entities can contribute. For example, if an event happened in a small town in the United States, then all the events that happened in the United States may have some sort of relationship with this event through the named entity "the United States". We cannot always use the town name either because the same name may be used in other countries. Similarly, names of famous persons may cause false alarms.

To avoid such noise, we adopt another weight adjustment strategy. The weight of a named entity is amplified only after it has been selected as a vector term. In other words, only the discriminating named entities will be given higher weights. Table 13 shows the results obtained after this modification was made. The performance improved when the weight of a discriminating named entity was amplified. The result 0.3740 is the best obtained in our experiments on the TDT-3 corpus.

Table 13. Named Entities with Amplifying Weights after Selecting

amplification	TH_{low}	TH_{high}	P(miss)	P(F/A)	Cdet (norm)
weight × 1	0	0.15	0.4010	0.0060	0.4304
weight × 2	0	0.15	0.3630	0.0027	0.3763
weight × 3	0	0.15	0.3552	0.0037	0.3740

9. Concluding Remarks and Future Works

The major problem in topic detection is that of finding the critical linkage cues. They usually cannot be obtained by means of training beforehand. They may not occur many times in a news story, or even may not occur many times in a topic cluster. This paper has presented the roles of named entities, other nouns and verbs in topic detection. Several issues given below have been discussed in this paper:

(1) Segmentation and translation (transliteration)
We employ contextual information in Chinese word segmentation and lexical translation. Errors of segmentation and named entity extraction may be propagated to perform similarity measurement and, thus, topic clustering. GB-to-BIG5 code conversion and vocabulary differences between China and Taiwan are two error sources. Lexical selection errors may affect the performance of English-Chinese topic detection. To decrease the propagation errors, we postulate that Chinese is less ambiguous than English. Chinese terms in a Chinese news vector are matched to Chinese terms in a topic cluster, and their corresponding English translations are matched to English terms in a topic cluster. On the other hand, English terms in an English news vector are only matched to English translations of Chinese terms in a topic vector.

(2) Linkage cues
Named entities, other nouns and verbs are plausible linkage cues. Two strategies have been proposed. With the top-N-weighted strategy, if there is noise in the incoming news, the new centroid may not be representative of the topic cluster. The LRU+Weighting strategy is more fault-tolerant than the top-N-weighted strategy, but it accordingly has more overheads. Named entities behave well when the top-N-weighted strategy is used. When nouns and verbs are regarded as linkage cues, they should be incorporated with the LRU+Weighting strategy.

(3) Time factors
A specific event is always reported in a short period of time, so the time window is used in several papers. However, setting a "good" time window for any topic is not an easy task. In this paper, a least-recently-used removal strategy has been proposed. Some older and unimportant terms in a topic cluster are moved if necessary.

(4) Thresholds
A two-threshold scheme has been proposed. Objects with similarity measures above a high threshold (below a low threshold) are considered to be relevant (irrelevant). Objects with similarity

measures in the ambiguous area should be checked further by means of lookahead. This models the effects of the news stories reported in a short time period.

From the experiments, we know that named entities are good candidates for use in topic detection. In this paper, we have not fully utilized this type of cue. Besides rough English name extraction, quantity features like money, dates, numbers, and so on in Chinese have not been employed completely. This is because there are many different forms which can represent the same quantity, especially in the multilingual environment. Missing quantity features will cause the detection performances to worsen. Nouns phrases may be good cues, too. In this paper, only tagging, not parsing, has been employed.

References

Allan, James; Papka, Ron; and Lavrenko, Victor (1998) "On-line New Event Detection and Tracking," *Proceedings of the 21st Annual International ACM SIGIR Conference*, Melbourne, 1998, pp. 37-45.

Bian, Guo-Wei and Chen, Hsin-Hsi (2000) "Cross Language Information Access to Multilingual Collections on the Internet," *Journal of American Society for Information Science*, 51(3), 2000, pp. 281-296.

Chen, Hsin-Hsi; Bian, Guo-Wei and Lin, Wen-Cheng (1999) "Resolving Translation Ambiguity and Target Polysemy in Cross-Language Information Retrieval," *Proceedings of 37th Annual Meeting of the Association for Computational Linguistics*, University of Maryland, 1999, pp. 215-222.

Chen, Hsin-Hsi; Ding, Yung-Wei and Tsai, Shih-Chung (1998) "Named Entity Extraction for Information Retrieval," *Computer Processing of Oriental Languages*, Special Issue on Information Retrieval on Oriental Languages, 12(1), 1998, pp. 75-85.

Chen, Hsin-Hsi; Ding, Yung-Wei; Tsai, Shih-Chung and Bian, Guo-Wei (1998) "Description of the NTU System Used for MET2," *Proceedings of 7th Message Understanding Conference*, Fairfax, VA, 1998, http://www.muc.saic.com/ proceedings/muc_7_toc.html.

Chen, Hsin-Hsi and Huang, Sheng-Jie (1999) "A Summarization System for Chinese News from Multiple Sources," *Proceedings of the 4th International Workshop on Information Retrieval with Asian Languages*, 1999, Taipei, Taiwan, pp. 1-7.

Chen, Hsin-Hsi; Huang, Sheng-Jie; Ding, Yung-Wei and Tsai, Shih-Chung (1998) "Proper Name Translation in Cross-Language Information Retrieval," *Proceedings of 17th International Conference on*

Computational Linguistics and 36th Annual Meeting of the Association for Computational Linguistics, Montreal, Quebec, Canada, 1998, pp. 232-236.

Chen, Hsin-Hsi and Lee, Jen-Chang (1996) "Identification and Classification of Proper Nouns in Chinese Texts," *Proceedings of 16th International Conference on Computational Linguistics*, Copenhagen, Denmark, 1996, pp. 222-229.

Chen, Hsin-Hsi and Lin, Chuan-Jie (2000) "A Multilingual News Summarizer." *Proceedings of 18th International Conference on Computational Linguistics*, 2000, Saarland University, pp. 159-165.

Church, K., *et al.* (1989) "Parsing, Word Associations and Typical Predicate-Argument Relations," *Proceedings of International Workshop on Parsing Technologies*, 1989, pp. 389-398.

Fellbaum, C. (1998) *WordNet: An Electronic Lexical Database*, MIT Press, Cambridge, Mass., 1998.

Harabagiu, S. (1998) *Usage of WordNet in Natural Language Processing Systems*, Proceedings of the Workshop, Montreal, Quebec, 1998.

Lin, Wei-Hao and Chen, Hsin-Hsi (2000) "Similarity Measure in Backward Transliteration between Different Character Sets and Its Application to CLIR," *Proceedings of 13th Research on Computational Linguistics and Chinese Language Processing Conference*, Taipei, Taiwan, pp. 97-113.

Mei, J.; *et al.* (1982) *tong2yi4ci2ci2lin2*. Shanghai Dictionary Press.

Rila, M. (1998) "The Use of WordNet in Information Retrieval," *Proceedings of ACL Workshop on the Usage of WordNet in Natural Language Processing Systems*, 1998. pp. 31-37.

Ruiz, M.; *et al.* (1999) "CINDOR Conceptual Interlingua Document Retrieval: TREC-8 Evaluation," *Proceedings of Eighth Text Retrieval Conference*, 1999.

Sproat, Richard, *et al.* (1994) "A Stochastic Finite-State Word-Segmentation Algorithm for Chinese," *Proceedings of 32nd Annual Meeting of the Association for Computational Linguistics*, New Mexico, 1994, pp. 66-73.

Yang, Yiming; Pierce, Tom; and Carbonell, Jame (1998) "A Study on Retrospective and On-Line Detection," *Proceedings of the 21st Annual International ACM SIGIR Conference*, Melbourne, 1998, pp. 28-36.

Zamir, Oren and Etzioni, Oren (1998) "Web Document Clustering: A Feasibility Demonstration," *Proceedings of the 21st Annual International ACM SIGIR Conference*, Melbourne, 1998, pp. 46-54.

Index

Adaptation
  model, 75
Annotation of corpora, 45
ASR, 3, 141
Bayesian, 69
Beta-binomial model, 118
BORG, 92
Centroid, 199
Cluster centroid, 199
Cluster detection, 5, 28, 77, 96, 128, 142, 154, 205, 250
Cluster pipelining, 154
Clustering, 79, 96, 152, 245
  on-line, 96
  two-tiered, 145, 152
Combination
  scores, 73
  systems, 92, 98
Corpora, 33
  development, 35
  formats, 54
  pilot, 8, 19
  quality control, 35, 42, 46, 49, 54
  raw data, 36
  reuse of, 34–35
  TDT-1, 8
  TDT-2, 9, 19
  TDT-3, 10, 19
  TDT-4, 13
  transcription, 38
Cost function, 20
Cross-language, 3, 80, 107, 126, 138, 143, 164, 180, 212, 226, 250
Decision tree, 137
Deferral period, 79, 155
Density estimation, 214
Detection error tradeoff curves, 24
Detection, 5, 28, 77, 96, 128, 142, 154, 205, 250
Dictionary, 236
Dictionary-based translation, 180
Distribution
  beta-binomial, 118
  of link detection scores, 212
  unigram, 117
Document expansion, 184, 209
Effectiveness bounds, 216
EM algorithm, 69
Entity tracking, 167
Entity vector space, 167
Evaluation methodology, 20
Event, 18
  v.subject, 2
Expansion, 184, 209
Feature weighting, 202
Filtering, 216
First story detection, 5, 29, 99, 208, 217
Generation
  modeling, 117
Hidden Markov model, 69
History of TDT, 7
HMM, 69
Information retrieval, 69
  relevance feedback, 70
Interpolation, 127
Introduction, 1
Judgments
  relevance, 45
K-means
  incremental, 78
K-NN, 90, 199
Kullback-Leibler distance, 118, 201
Language modeling, 91, 200
Least recently used, 245
Lexical analysis, 160
Link detection, 7, 26, 101, 208
Local context analysis (LCA), 209
Maximum entropy model, 138
Microclusters, 142
Mixture models, 70
Model adaptation, 75
Modeling
  beta-binomial, 118
  generation, 117
  language, 91, 115, 200
  maximum entropy, 138
  statistical, 115
Models
  probabilistic, 67

Multilingual, 80, 107, 164, 180, 212, 226, 250
Name transliteration, 250
Named entities, 167, 245
Normalization, 179, 231
  scores, 72, 78
Normalized cost, 22
Part-of-speech tagging, 160
Pilot corpus, 8, 19
Pilot study, 8
Pipelining, 154
Probabilistic models, 67
Query expansion, 184, 209
Relevance feedback, 70
Relevance judgments, 45
Removal strategy, 245
Rocchio, 89
Score combination, 73
Score normalization, 72, 78, 179, 212, 231
Segmentation, 4, 29, 87, 135, 152
  manual, 39
Signal-to-noise, 177
Similarity, 152, 199
  story-topic, 69
Smoothing, 117
Spanish, 108
Speech recognition, 3, 141
Story link detection, 7, 26, 101, 208
Story segmentation, 4, 29, 87, 135, 152
  manual, 39
Story similarity, 199
Story, 18
Story-link annotation, 54
Subject
  v.event, 2
System combination, 92, 98
Systran, 191
Targetting, 131
TDT 1997, 8
TDT 1998, 9
TDT 1999, 10
TDT 2000, 10
TDT 2001, 13
TDT
  history, 7
  introduction, 1
  pilot study, 8
TDT-1 corpus, 8
TDT-2 corpus, 9, 19
TDT-3 corpus, 10, 19
TDT-4 corpus, 13
Temporal nature, 2
Term lists, 182, 189
Term vector space, 152
Term weighting, 202
Time
  use of, 76
Timeline generation, 219
Topic annotation, 45
Topic detection, 5, 28, 77, 96, 128, 142, 154, 205, 250
Topic selection, 45
Topic spotting, 69
Topic tracking, 6, 25, 75, 88, 120, 163, 178, 203, 229
Topic, 2, 18
  what is a, 42
Tracking, 6, 25, 75, 88, 120, 163, 178, 203, 229
Transcription of corpora, 38
Translation
  dictionary size, 236
  dictionary-based, 180
Translingual, 3, 80, 107, 126, 138, 143, 164, 180, 212, 226, 250
Two-tiered clustering, 145, 152
Unigram model, 117
Vector space, 152, 198, 229
  entity, 167
Word presence, 71