Emerging Trend Detection from Scientific Online Documents

by

Le Minh Hoang

submitted to
Japan Advanced Institute of Science and Technology
in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy

Supervisor: Professor Yoshiteru Nakamori

School of Knowledge Science
Japan Advanced Institute of Science and Technology

March 2006

Acknowledgements

First of all, I would like to express my deep gratitude to Professor Yoshiteru Nakamori for his supervision. I am very happy to have a supervisor who always has great ideas, one of which helped me to open my research direction. Without his knowledge, perceptivity, I would never have finished. He always gives me his helpful encouragements at the right time and I learnt many things from him. I greatly appreciate the way he kindly treats his students.

I would like to thank Professor Ho Tu Bao for his great, continuous help during three years I have been in JAIST. He taught me a lot of experience on doing research. He was always willing to help me to solve difficult things. I also admire the way he guide his students.

I am thankful to my examiners, Professor Hiroshi Motoda, Professor Kenji Satou and Professor Takashi Hashimoto, for reading the whole dissertation and giving me a lot of useful comments to improve my work.

I would like to say a big 'thank-you' to the Japanese Government Scholarship (Monbukagakusho) for supporting me to have a good chance to study in Japan, where I can obtain a lot of knowledge in the professionally-working environment.

I thank to all students and colleagues in JAIST for being the surrogate family during the many years and for their supports in difficult periods.

I would like to express my greatest thanks to my parents, who have spent whole life for my future. My father, who is too old for working, but still has to do anything for my family. He always encourages and reminds me of trying the best in study and research. I am indebted to my mother for her boundless affection. Mom, you are an

eternal flame in my heart and I hope you are happy in the heaven to see anything I have done as you expected.

I thank to my sister, who takes care of my parents when I have been absent from home. Nothing can compensate for her dolorous caused by the loss of our mother, but I hope she can overcome obstacles and difficulties to continue doing the best for her life with much success and happiness.

Finally but very important, I want to thank my wife for her love. She always makes me happy and optimistic in the life. She has made so many sacrifices for me that I can say this research could not be done without her encouragements.

JAIST, January 31 st, 2006

Le Minh Hoang

Abstract

The rapid increase in volume of scientific literature has led to researchers overload in their pursuit of knowledge. Staying up-to-date with recently published literature - and actually finding relevant sources - is becoming increasingly difficult, time-consuming, and impossible. Experience varies widely, but the time when every essential journal was held in all major academic libraries has passed.

Emerging trend detection is a new challenge and an attractive topic in text mining. With the continued increases in performance of computational technologies, more complex implementations of text-processing techniques are become possible, this has spurred research into the development of more sophisticated methodologies for developing emerging trend detection methods. However, there is no model that is particularly constructed for scientific corpora while existing models do not appear to be appropriate for this especially important kind of text databases. Previous work lack of an appropriate represent scheme for research topics and an effective method to identify emerging trends.

Building a model for emerging trend detection in scientific corpora is our major research objective. To this end, we have made the following contributions:

1. A model for emerging trend detection in scientific corpora, which presents various advantages in comparing to existing models.

2. A scheme for topic representation based on the rich information commonly provided in scientific papers, which can adapt to different kinds of scientific corpora and also can be efficiently modified.

3. Methods for topic identification, which are used to extract temporal features from documents. They appear to be more accurate and powerful than others in our experiments.

4. A topic verification method based on the interest and utility functions. This can be used for classification of emerging trends.

5. A prototype system to evaluate the model, that is used to evaluate if our model can achieve significant results in emerging trend detection.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 The Need for Emerging Trend Detection

The explosive growth of digital repositories of information has been enabled by recent developments in communication and information technologies [PY01]. In this fast changing information age, knowledge of emerging trends in the a particular area of interest is very important for people who want to stay up-to-date with recently published literature, such as research scholars, movie or film producers, fashion designer and follower, etc. For example, a market analyst might want to review technical and news-related literature for recent trends that will impact the companies he is tracking; people who are looking for their research topic have to read many papers in their research domain for reviewing related works, finding relevant sources, advanced techniques supporting for their researches.

What is an Emerging Trend? An emerging trend is a topic area that is growing in interest and utility over time [KGP+03]. For example, Extensible Markup Language (XML) emerged as a trend in the mid 1990s. Table 1.1 shows the results of an INSPEC® database search on the keyword "XML" from 1994 to 1999 in which we can see the trend of XML from 1994 to 1997; by 1998 it was well represented as a topic area.

Emerging Trend Detection (ETD) is the process of finding emerging trends in a specific domain in order to provide an automated alert when new developments are

Year	Number of documents
1994	3
1995	1
1996	8
1997	10
1998	170
1999	371

Table 1.1: Emergence of XML in the mid-1990s.

happening in an area of interest and utility. It is assumed that a detected trend is an indication that some event has occurred. To detect emerging trends, we have to look at the data to determine the underlying development. Whether the development is important or not is a decision that depends on the situation and the particular information needs of the person evaluating the data. The need to become aware of new developments in science, technology, or business is critical to researchers, decision makers at all levels of a corporation. These people need to stay up-to-date with current development in their daily work. They need data that is complete and available in a timely manner. Traditionally, people have learned about a majority of the new developments by reading various types of text documents or by getting the information from others who have read the documents. As the pace of new developments accelerates and the number of documents increases exponentially, it will no longer be possible for an individual to keep up with what is happening by using manual processes. There is a clear need for new tools and methodologies to bring some level of automation to detect trends and new developments. With the continued increases in performance of computational technologies, more complex implementations of text-processing techniques have become possible, this has spurred research into the development of more sophisticated methodologies for the detection of emerging trends.

1.2 Emerging Trend Detection for Scientific Text

One very significant task for ETD is to find emerging research trends in a collection of scientific articles. Imagine that we are researchers, looking for topics that have recently attracted much interest and utility in a particular domain. A manual review of all available articles in this domain would be very time-consuming and virtually impossible. In this situation, the automatic detection of emerging research trends can help researchers quickly understand the occurrence and the tendency of a scientific topic, and thus they can, for example, find the most recent, related topics in their research domain.

As a new and challenging problem in text mining, several ETD methods have been developed which takes as input a collection of textual data and identifies topic areas that are either novel or are growing in interest and utility within the corpus. Most of existing researches on ETD work around three main tasks: topic representation, topic identification, and topic verification. Each topic – the ETD central notion – is usually represented by a set of temporal features in the phase of *topic representation*. These features are then extracted from document databases using text-processing methods in the *topic identification* phase. After that, the *topic verification* phase plays the role of monitoring these features over time and classifying the topic by using interest and utility functions [KGP+03].

While many ETD models have been proposed, they are still poor in representing research topics [SA00, APL98] and inappropriate for determining and ranking interest and utility [Gev02, HHWN02]. There are two main reasons why these ETD models do not appear to be robust when applied to scientific corpora. First, many features can be extracted from scientific articles but may not be available in other textual data, meaning that these features cannot be integrated into general ETD models. The second limitation lay on the interpretation of interest and utility measures for evaluating research topics. This process is somewhat subjective and requires complex computations in analyzing the features associated with each topic.

1.3 Research Objective

The main objective of our work is to build an ETD model for scientific corpora. To this end, we aim to build an appropriate model structure and construct the following model components:

Topic representation: The first goal is to find a rich representation scheme for topics, that is reasonable, explainable and appropriate to distinguish emerging and non-emerging trends.

Topic identification: The second goal is to identify topics by features extracted from documents. We aim to develop effective methods for automatically extracting features that do not require user-interactions or explicit knowledge.

Topic identification: The third goal is to formulate two interest and utility measures. Due to the difficulty of the topic verification task, existing work on emerging trend detection usually detect topic areas that have grown in size and variety at an increasing rate over time. We want to evaluate the growth of a topic in interest and utility separately in order to make the topic verification method more reasonable in classification of emerging trends.

1.4 Contributions of This Work

In context of scientific emerging trend detection, our work is valuable in that it proposed a more appropriate ETD model and developed effective methods for each model components. The contributions of this work include:

1. A model for emerging trend detection in scientific corpora, which presents various advantages in comparing to existing models.

2. A scheme for topic representation based on the rich information commonly provided in scientific papers, which can adapt to different kinds of scientific corpora

and also can be efficiently modified. Features used for representing topics include topic name, citation information, influence, author reputations, weight of sources. These human-understandable features are extracted from documents and will be used for distinguishing emerging and non-emerging trends as well as providing evidences that indicate whether a topic is truly emerging.

3. Methods for topic identification, which are used to extract temporal features from documents. They appear to be more accurate and powerful than others in our experiments. Our method to detect topics of each document based on concept hierarchy is more accurate that other methods for scientific documents. We have also proposed a method to detect reasons for citations using finite-state machines, which has an appropriate definition of citation types and does not require user-interactions or explicit knowledge.

4. A topic verification method based on the interest and utility functions. We formulated two interest and utility measures for evaluating emerging trends, that enable us to make the topic verification method more reasonable in classification of emerging trends.

5. A prototype system to evaluate the model with some data sets, that is used to evaluate if our model can achieve significant results in emerging trend detection.

1.5 Structure of Dissertation

This dissertation is organized into six chapters:

Chapter 1

Overview and introduction to the need and importance of emerging trend detection in scientific online documents.

Chapter 2

The state of the art of emerging trend detection researches. Some emerging trend

detection methods are studied. Techniques required for emerging trend detection are presented. We summarize how other works represent topics, extract features and verify topics. Further, we discuss the advantages and limitations of each method.

Chapter 3

This chapter focuses on the structure of an emerging trend detection model for scientific corpora, including the representation of topics using temporal features

Chapter 4

Methods for the topic identification task, which are used for extracting temporal features associated with topics, are explained. With details and supportive experimental results, topic detection and citation type detection algorithms will be introduced, followed by the comparison with other works.

Chapter 5

This chapter focuses on building the interest and utility measures, which are key components of the topic verification task. An overview of the prototype system to test the model, and some experimental results are also discussed.

Chapter 6

Conclusions, importance and significance of this work, future works are presented.

Chapter 2

Current Models and Methods for Emerging Trend Detection

In this chapter, we provide an overview of the status of emerging trend detection research. Although many emerging trend detection models have been proposed, the common components of an ETD model have not, to date, been clearly defined. Most of the related works developed ETD methods that take as input a collection of textual data and identify topic areas that are either novel or are growing in interest and utility within the corpus. Each topic – the ETD central notion – is usually represented by a set of temporal features extracted from document databases using text-processing methods. After that, these features are monitored over time, and emerging trends are detected [KGP+03]. The effectiveness of an ETD model completely depends on how appropriately a topic is represented in computers, how well the features associated with a topic are extracted from the documents and how reasonably the method for verifying topics are constructed. Therefore, the approaches taken in these various tasks can be summarized in the three main tasks: *Topic representation*, *Topic identification* and *Topic verification*.

Current ETD methods fall generally into two categories: fully-automatic and semi-automatic. The fully-automatic methods take in a corpus and develop a list of emerging topics. A human reviewer then peruses these topics and the supporting evidence found

by the method to determine which are truly emerging trends. These methods often include a visual component that allows the user to track the topic in an intuitive manner [DHJ+98], [SA00]. Semi-automatic methods [PD95], [RGP02] require user to input a topic and provide the user with evidence that indicates whether the input topic is truly emerging, usually in the form of reports and screens that summarize the evidence available on the topic.

In the following sections, we will briefly describe some semi-automatic and fully-automatic methods. For each ETD method, we will clarify how the work represent topics, extract feature and verify topics in order to compare with our ETD model which is described in further chapters.

2.1 Pioneering Work: The TDT Project

TDT research began in 1997, that develops algorithms for discovering and threading together topically related material in streams of data, such as newswire and broadcast news, in both English and Mandarin Chinese. Although the TDT project did not directly focus on emerging trend detection, it provided a data repository for many ETD researches.

The TDT data sets are sets of news stories and event descriptors. Each story/event pair is assigned a relevance judgment. A relevance judgment is an indicator of the relevance of the given story to an event. Table 2.1 includes several examples of the relevance judgment assignment to a story/event pair. Thus, the TDT data sets can be used as both training and test sets for ETD algorithms. The Linguistic Data Consortium (LDC) [LDC] currently has three TDT corpora available for system development, the TDT Pilot study (TDT-Pilot), the TDT Phase 2 (TDT2), the TDT Phase 3 (TDT3), as well as the TDT3 Arabic supplement.

Not all of the ETD methods we describe rely on the TDT data sets. Other approaches for the creation of test data have been used, such as manually assigning relevance judgments to the input data and comparing the system results to the results

Story Description	Event	Relevance Judgment
Story describes survivor's reaction after Oklahoma City Bombing	Oklahoma City Bombing	Yes
Story describes survivor's reaction after Oklahoma City Bombing	US Terrorism Response	No
Story describes FBI's increased use of surveillance in government buildings as a result of the Oklahoma City Bombing	Oklahoma City Bombing	Yes
Story describes FBI's increased use of surveillance in government buildings as a result of the Oklahoma City Bombing	US Terrorism Response	Yes

Table 2.1: Story/event pairs in TDT datasets

produced by a human reviewer. This approach is tedious and necessarily limits the size of the data set. Some related works use other databases such as INSPEC® [INS], which contains engineering abstracts, or the United States patent database, which allows searching of all published US patents.

2.2 Semi-Automatic Methods

Semi-automatic methods require user to input a topic and provide the user with evidence that indicates whether the input topic is truly emerging, usually in the form of reports and screens that summarize the evidence available on the topic. Therefore, they focus on each individual topic, not on the complex relations of all topics in a given text database. Semi-automatic methods do not make any decisions on emerging trends, the users have to make decisions based on the evidence provided in the output.

2.2.1 Technology Opportunities Analysis (TOA)

Alan L. Porter and Michael J. Detampel describe a semi-automatic trend detection method for technology opportunities analysis in [PD95]. The first step of the process

is the extraction of documents (such as INSPEC® abstracts) from the knowledge area to be studied. The extraction process requires the development of a list of potential keywords by a domain expert. These keywords are then combined into queries using appropriate Boolean operators to generate comprehensive and accurate searches. The target databases are also identified in this phase (e.g., INSPEC® , COMPENDEX® [COM], US Patents [Sit], etc.).

The queries are then input to the Technology Opportunities Analysis Knowbot module [PD95] to extract the relevant documents (abstracts) and provides an analysis of the data. Word counts, date information, word co-occurrence information, citation information and publication information are used to track activity in a subject area. TOAK facilitates the analysis of the data available within the documents. For example, it can quickly generate the lists of frequently occurring keywords, the lists of author affiliations, countries, or states.

In [PD95], the authors present an example of how TOAK can be used to track trends in the "multichip module" sub field of electronic manufacturing and assembly. Table 2.2 shows a list of keywords that appear frequently with "multichip module" in the INSPEC® database. The authors observed that multichip modules and integrated circuits (particularly hybrid integrated circuits) co-occurred very frequently. An additional search using the US Patent database showed that many patents had been issued in the area of multichip modules. Furthermore, the integrated circuits activity was more likely to be US based, while large scale integration activity was more likely to be based in Japan.

TOAK is meant to be used by a human expert in an interactive and iterative fashion. The user generates initial queries, reviews the results and is able to revise the searches based on his/her domain knowledge. TOA represents an alternative approach to the time-consuming literature search and review tasks necessary for market analysis, technology planning, strategic planning or research.

Keyword	Number of articles	Keyword	Number of articles
Multichip modules	842	Circuit layout CAD	69
Packaging	480	Tape automated bonding	68
Hybrid integrated circuits	317	Printed circuit manufacture	66
Module	271	Printed circuit design	65
Integrated circuit technology	248	Thin film circuit	62
Integrated circuit testing	127	CMOS integrated circuits	56
Substrates	101	Soldering	50
VLSI	98	Optical interconnections	48
Surface mount technology	93	Lead bonding	44
Flip-chip devices	93	Integrated optoelectronics	43
Integrated circuit manufacture	88	Printed circuits	42
Ceramics	85	Production testing	41
Circuit reliability	80	Reliability	41
Polymer films	79	Microassembling	38
Cooling	70	Circuit CAD	35
Metallisation	69	Microprocessor chips	35

Table 2.2: Co-occurrences with "multichip modules"

Topic representation

TOA uses INSPEC® database server as the primary corpus. Each topic is represented by a list of keywords (a single word or multiple words termed n-grams[1]) and their possible combinations using Boolean operators. Each keyword is associated with a number of keyword occurrence and pairwise co-occurrence, which are calculated per year and over all years.

Topic identification

Because TOA uses a simple representation for topics and the set of keywords is manually generated, the topic identification task is simply a process of counting n-gram occurrence and pairwise co-occurrence over all articles in INSPEC® database.

[1]An n-gram is a sequence of n words. For example, the phrase 'data mining' is a bigram (or 2-gram).

Topic verification

TOA does not provide any decision for emerging trends. Instead of making an automatic topic verification module, it uses frequency tables, histograms, weighted ratios, log-log graphs, Fisher-Pry curves, and technology maps to visualize the trend of the input topic, identification of trends is left to users.

2.2.2 Constructive, Collaborative Inquiry-Based Multimedia E-Learning (CIMEL)

CIMEL is a multi-media framework for constructive and collaborative inquiry-based learning. It includes an semi-automatic emerging trend detection method [RGP02] in order to enhance computer science education. With a given input topic, the method identifies the main topic area for research and recent conferences and workshops in this area. After reviewing contents and creating a list of candidate emerging trends, the method verifies each candidate using a database search tool and/or a Web search engine.

Topic representation

The corpus for this semi-automatic methodology can be any web resource. Each topic is associated with several features as shown in Table 2.3.

Topic identification

Unlike TOA, CIMEL provides specific parameters for identifying an emerging trend, rather than relying solely on the domain expertise of the user. Most of these features associated with a topic are automatically extracted from the corpus. Like TOA, this method is is restricted by the electronic availability of documentation in a given subject area. Furthermore, the INSPEC® query tool is currently based on abstracts that are downloaded to a local database, which must be periodically refreshed.

Feature	Type	Generation
Domain Name	n-grams	Manual
Topic Name	n-grams	Manual
Supporting terms	n-grams	Automatic
Search query	n-grams	Automatic
Date	Year	Automatic
Number of domains/topics in a document	Frequency	Automatic
Number of supporting terms	Frequency	Automatic
Line or paragraph containing the domain, topic and supporting terms in a given document	n-grams	Manual
Number of unique authors per year	Frequency	Automatic
Number of unique documents per year	Frequency	Automatic
Number of unique author set per year	Frequency	Automatic
Number of unique journal/proceedings per year	Frequency	Automatic

Table 2.3: Representation of a topic in CIMEL

Topic verification

Like TOA, the ETD method implemented in CIMEL relies on the user to detect emerging trends. No machine-learning component is employed. Instead CIMEL relies on a precisely defined manual process.

2.3 Fully-Automatic Methods

Fully-automatic methods do not require user's topic input, they take in a corpus and develop a list of emerging topics. Therefore, they must consider the complex relations of all topics in a given text database in order to make decisions on emerging trends.

2.3.1 TimeMines

The ETD method implemented in TimeMines system [SA00] takes free text data, with explicit date tags, and develops an overview time-line of statistically significant topics covered by the corpus. This method employs hypothesis-testing techniques to determine the most relevant topics in a given time-frame. Only the most significant

and important information (as determined by the program) is presented to the user.

TimeMines begins processing with a default model that assumes the distribution of a feature depends only on a base rate of occurrence that does not vary with time. Each feature in a document is compared to the default model. A statistical test is used to determine if the feature being tested is significantly different from what the model would expect. If so, the feature is kept for future processing, otherwise it is ignored.

The reduced set of features developed using the first round of hypothesis testing is then input into a second processing phase which groups related features together. The grouping again relies on probabilistic techniques that combine terms that tend to appear in the same time-frames into a single topic. Finally, a threshold is used to determine which topics are most important and these are displayed via the time-line interface. The threshold is set manually, and is determined empirically. Like TOA, TimeMines presents a model of the data without drawing any specific conclusions about whether or not a topic is emergent. It simply presents the most statistically significant topics to the user, and relies on the user's domain knowledge for evaluation of the topics

Topic representation

In TimeMines, an initial attribute list of all named entities and certain noun phrases is generated. A named entity is defined as a specified person, location, or organization. The documents are thus represented as a bag of attributes, where each attribute is true or false (i.e., whether the named entity or noun phrase is contained in the document or not).

TimeMines uses a statistical model based on hypothesis testing to choose the most relevant features. As noted, the system assumes a stationary random model for all features (n-grams and named entities) extracted from documents. The stationary random model assumes that all features are stationary (meaning their distributions do not vary over time) and the random processes generating any pair of features are independent. Features whose actual distribution matches this model are considered to contain no

new information and are discarded. Features that vary greatly from the model are kept for further processing. The hypothesis testing is time dependent, i.e for a specific block of time, a feature either matches the model (at a given threshold) or violates the model.

Topic identification

In TimeMines, named entities are extracted using the Badger IE system [FSM+95] with Noun phrases match the regular expression $(N|J) \star N$ for up to five words, where N is a noun, J is an adjective, | indicates union, and $\star$ indicates zero or more occurrences. Each document has a presence attribute for each named entity and n-gram. A presence is assigned the value 'True' if the named entity or n-gram occurs in the document, else it is assigned the value 'False'.

Topic verification

TimeMines uses an algorithm based on hypothesis testing. Using the reduced feature set, TimeMines checks for features within a given time period that have similar distributions. These features are grouped into a single 'topic'. Thus each time period may be assigned a small number of topic areas, each represented by a larger number of features.

The final determination of whether or not a topic is emerging is left to the user, but unlike CIMEL and TOA, the user need not direct the ETD process. This method is completely automated: given a time-tagged corpus it responds with a graphical representation of the topics that dominate the corpus during specific time periods.

2.3.2 New Event Detection

New event detection, also referred to as first story detection, is specifically included as a subtask in the TDT initiative. New event detection requires identifying those news stories that discuss an event that has not already been reported in earlier stories.

Feature	Type	Generation
Unigram	A single word	Automatic
Number of times unigram occurs per story	Frequency	Automatic
Total number of unigrams per story	Frequency	Automatic
Average number of unigrams per story	Mean	Automatic
Number of stories in which unigram occurs	Frequency	Automatic
Number of stories	Count	Automatic
Date	Date	Automatic

Table 2.4: Representation of a topic in New Event Detection

New event detection operates without a predefined query. Typically algorithms look for keywords in a news story and compare the story with earlier stories. The method taken in [APL98] implies that the input be processed sequentially in date order: i.e., only past stories can be used for evaluation, not the entire corpus.

A new event detection algorithm is based on a single pass clustering algorithm. The content of each story is represented as a query. When a new story is processed, all the existing queries (previous stories) are run against it. If the 'match' exceeds a predefined threshold (discussed below) the new story is assumed to be a continuation of the query story. Otherwise it is marked as a new story.

Topic representation

All stories in the TDT corpus deemed relevant to twenty five selected 'events' were processed. For new event detection, each story (topic) is represented by a set of single words (called unigrams) associated with some features as shown in Table 2.4.

Topic identification

In [APL98], the n most frequent single words comprise the query, and are weighted and assigned a belief value by the Inquery system [ABC+95], indicating the relevance of each word in the story to the query. Belief is calculated using term frequency and inverse document frequency. Term frequency is derived from the count of times the

word occurs in the story, the length of the story, and the average length of a story in the collection. Inverse document frequency is derived from the count of stories in the collection and the count of stories that contain the word.

Topic verification

The method of [APL98] is based on a single-pass clustering algorithm that detects new stories by comparing each story processed to all of the previous stories/queries detected. As each incoming story is processed, all previous queries are run against it. If a story does not match any of the existing queries, the story is considered a new event.

The system relies on a threshold to match the queries to the incoming stories. The initial threshold for a query is set by evaluating the query with the story from which it originated. If a subsequent story meets or exceeds this initial threshold for the query, the story is considered a match. The threshold is used as input to a function based on the Inquery system described above [ABC$^+$95]. Since new event detection implies that documents are processed in order, however, traditional IR metrics that are usually applied to an entire corpus (such as the number of documents containing the term and average document length) are not readily available. To overcome this problem, an auxiliary collection is used to provide this information to the Inquery system to take advantage of the time dependent nature of the news story collection by using a time penalty that increases the value required to 'match' a story as stories grow further apart in time.

Like the TimeMines, the new event detection method described here is completely automated. Given a corpus, it provides a list of new events in the form of news stories that first describe an occurrence of an event. New event detection differs somewhat from ETD in that it is focused on the sudden appearance of an unforeseen event rather than the emergence of a trend.

2.3.3 ThemeRiver™

Similar to TimeMines, ThemeRiver™ [HHWN02] summaries the main topics in a corpus and presents a summary of the importance of each topic via a graphical user interface. The topical changes over time are shown as a "river" of information. The river is made up of multiple streams. Each stream represents a topic and each topic is represented by a color and maintains its place in the river relative to other topics.

Topic representation

The corpus in the example presented in [HHWN02] consisting of speeches, interviews, articles, and other text about Fidel Castro over a 40-year period. ThemeRiver™ automatically generates a list of possible topics, called theme words, of which a subset is manually chosen as attributes. Counts of the number of documents containing a particular theme word for each time interval provide the input for the method. An alternate count, using the number of occurrences of the theme word for each time interval is suggested but not implemented in this work.

Topic identification

An automatic method for generating the initial list of theme words was not specified, nor was the procedure for deciding which or how many of the theme words should be included in the subset. Theme word frequencies are computed after these attributes are chosen, effectively making attribute selection a manual process.

Topic verification

ThemeRiver™ does not provide any algorithm for verifying topics. It only provides a view of the data that an experienced domain expert can use to confirm or refute a hypothesis about the data. ThemeRiver™ begins by converting time-tagged data into time intervals. A set of terms, or themes, that represent the data is chosen and the river is developed based on the strength of each theme in the collection. As noted,

the themes are chosen by automatically developing a list of words that are present in the data and then manually selecting a subset that represent various topics. The number of documents containing the word determines the strength of each theme in each time interval. Other methods of developing the themes and strengths are possible. The visual component of ThemeRiver^{TM} is the most important aspect of this work, particularly as it applies to trend detection.

2.3.4 PatentMiner

The ETD method implemented in PatentMiner system can discover trends in patent data using a dynamically generated SQL query based upon selection criteria input by the user [LAS97]. This method uses IBM DB2 database which contains all granted US patents as its dataset. There are two major components to the system, phrase identification using sequential pattern mining [SA96] and trend detection using shape queries.

Topic representation

PatentMiner represents each topics by phrases. A phrase can be any sequence of words, with a minimum and maximum 'gap' between any of the words. Gaps can be described in terms of words, sentences, paragraphs, or sections. For example, the minimum sentence gap is one for the phrase "emerging trends" means two phrases "emerging" and "trends" must occur in separate sentences. If the maximum paragraph gap is one, two phrases 'emerging' and 'trends' must occur in the same paragraph. Table 2.5 summarizes features associated with a topic in PatentMiner.

Topic identification

Several procedures prepare the data for the topic identification task. Stop-words are removed. Identifiers are assigned to the remaining words, indicating position in the document and occurrences of sentence, paragraph, and section boundaries. After a

Feature	Type	Generation
Phrases	n-grams	Manual
Minimum gap, with distinct gaps for words, sentences, paragraphs, and sections.	Size	Manual
Maximum gap, with distinct gaps for words, sentences, paragraphs, and sections.	Size	Manual
Time window, (PatentMiner groups words in a phrase before determining gaps)	Size	Manual
Support, number of search phrases returned divided by total number of phrases	Ratio	Manual
Date	Date	Manual
Graphical trend appearance over time, e.g., spiked or downwards	Shape	Manual

Table 2.5: Representation of a topic in PatentMiner

subset of patents is specified by category and date range, the Generalized Sequential Patterns (GSP) algorithm [LAS97] selects phrases. Only phrases with support greater than a user-defined minimum are considered.

The number of phrases selected can be substantial, given their very open-ended nature. A sub-phrase of a phrase may be ignored if the support of the two phrases is similar. Or, a sub-phrase (general, higher-level) might be preferred over a longer phrase (specific, lower-level) initially, after which specific lower-level phrases could be easier to identify.

The topic identification method in PatentMiner takes a different approach compared to other methods which use traditional IR techniques to extract features from the text corpus. It adapts a sequential pattern matching technique that is frequently used in data mining systems. The pattern matching system looks for frequently occurring patterns of words. The words may be adjacent, or separated by a variable number of other words (up to some maximum that is set by the user). This technique allows the system to identify frequently co-occurring terms and treat them as a single topic.

Documents in the input data set are binned into various collections based on their

date information. The above technique is used to extract phrases from each bin and the frequency of occurrence of each phrase in all bins is calculated. A shape query is used to determine which phrases to extract, based on the user's inquiry. The shape query processing is another learning tool borrowed from data mining. In the PatentMiner system, the phrase frequency counts represent a data store that can be mined using the shape query tool. The shape query has the ability to match upward and downward slopes based on frequency counts. The shape query allows the user to graphically define various shapes for trend detection (or other applications) and retrieves the phrases with frequency distributions that match the query.

Topic verification

The presentation of PatentMiner lacks topic verification component. While it automatically generates and displays potential trends, no claim is made as to the validity of these trends. The visualization is intuitive, but no user study on its effectiveness is reported in [LAS97]. In addition, no metrics are employed to verify that the trends discovered are correctly identified.

2.3.5 Hierarchical Distributed Dynamic Indexing

The Hierarchical Distributed Dynamic Indexing (HDDI^{TM}) system supports core text processing including information/feature extraction, feature subset selection, unsupervised and supervised text mining and machine learning as well as evaluation for many applications, including an emerging trend detection method.

In [PY01], the authors describe an approach to the detection of emerging trends in text collections based on semantically determined clusters of terms. The HDDI^{TM} system is used to extract linguistic features from a repository of textual data and to generate clusters based on the semantic similarity of these features. The algorithm takes a snapshot of the statistical state of a collection at multiple points in time. The rate of change in the size of the clusters and in the frequency and association of features

Feature	Type	Generation
Number of times concept occurs in trial year	Frequency	Automatic
Number of times concept occurs in year before trial year	Frequency	Automatic
Number of times concept occurs in the year two years before trial year	Frequency	Automatic
Total number of times concept occurs in all years before trial year	Frequency	Automatic
Number of concepts in region containing the concept in trial year	Count	Automatic
Number of concepts in region containing the concept in the year before trial year	Count	Automatic
Number of words in the concept with length at least four	Count	Automatic

Table 2.6: Representation of a topic in HDDI^{TM}

is used as input to a neural network that classifies topics as emerging or non-emerging.

Topic representation

In HDDI^{TM} , a topic is represented by a region of semantic locality which is a group of concepts created using sLoc [BP00]. Each topic is associated with seven features as shown in Table 2.6

Topic identification

Initially, topic identification method in HDDI^{TM} requires parsing and tagging before extraction. The parser retains only relevant sections of the original documents. The tagger maps a part-of-speech label to each word using lexical and contextual rules [Bri92]. A finite-state machine extracts complex noun phrases (concepts) according to the regular expression:

$$C?\,(G|P|J)*N+(I*D?C?\,(G|P|J)*N+)* \tag{2.1}$$

where C is a cardinal number, G is verb (gerund or present participle), P is a verb (past participle), J is an adjective, N is a noun, I is a preposition, D is a determiner, ?

indicates zero or one occurrence, | indicates union, $*$ indicates zero or more occurrences, and + indicates one or more occurrence. Counts of each concept and counts of co-occurrence of concept pairs are recorded at this point.

An asymmetric similarity between concept pairs is calculated based on a cluster weight function described in [CL92]. The concepts are then grouped into regions of semantic locality using sLoc, an algorithm described in [BP00]. The maximum, mean and standard deviation of the similarity, along with a parameter α that is a multiplication factor of standard deviations, determine the threshold r used in the first step of the sLoc algorithm. Cluster size is used in the last step, as α decreases, r increases and the number of connections between pairs of concepts decreases, resulting in smaller but more focused semantic regions.

Topic verification

The emerging trend detection method in HDDI^{TM} assumes that an emerging concept have to satisfy two principles: it should grow semantically richer over time and it should occur more often as more items reference it. Using a cluster-based rather than an item-based approach, an artificial neural network model takes seven features and one tuning threshold parameter to classify a concept as emerging or not. The structure of the artificial neural network and the topic verification process were shown in the Figure 2.1.

2.4 Other Related Works

Feldman and Dagan [FD95] proposed a technique for development of a hierarchical data structure from text databases. This data structure then facilitates the study of concept distributions in text. The authors propose comparing the concept distributions from adjacent time periods. This approach to trend analysis seems promising; however, we were not able to obtain a more detailed description of the approach, or the experimental results, so we are unable to present a more comprehensive summary. The authors

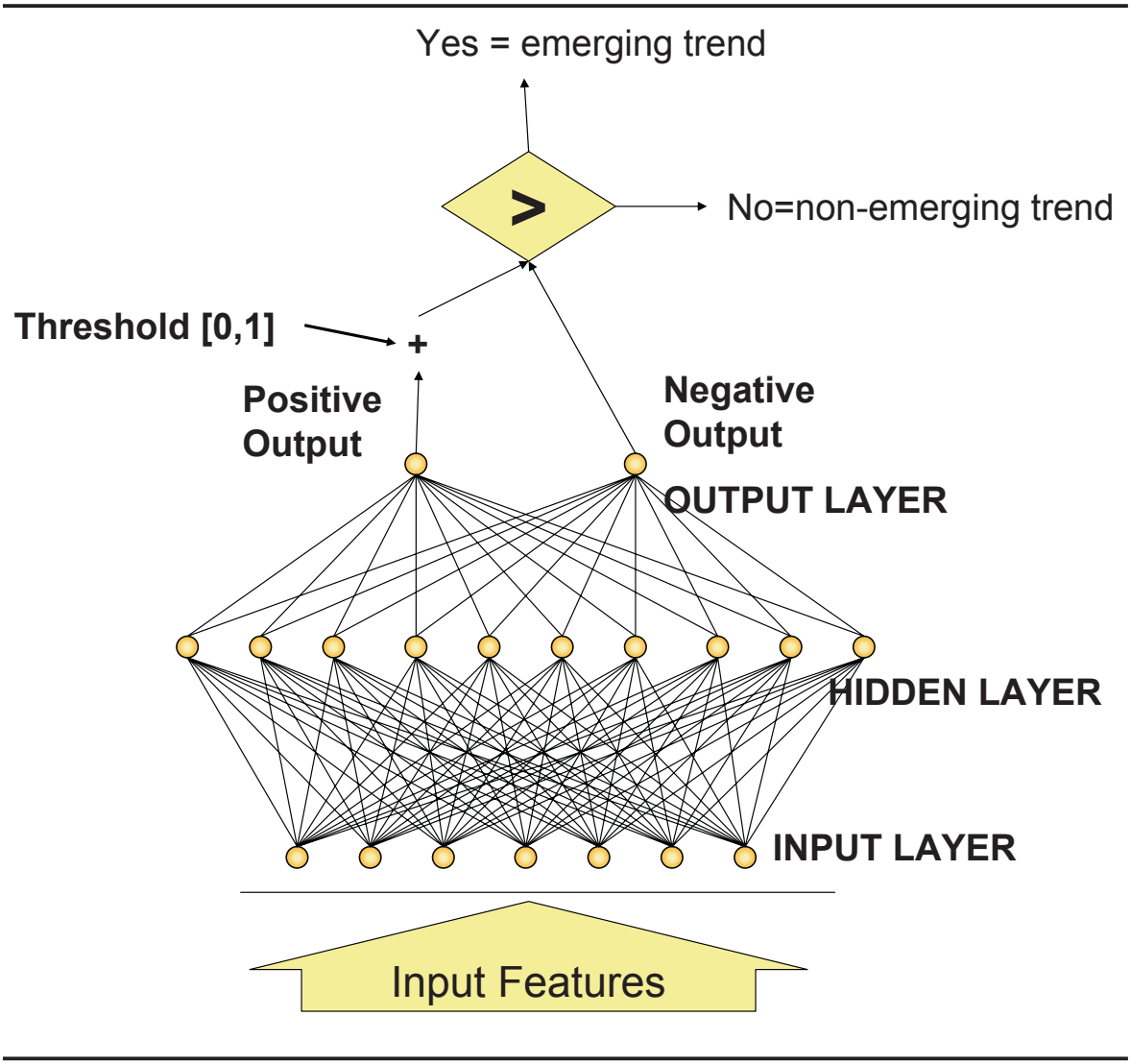

Figure 2.1: Using an artificial neural network to detect emerging trends

has also been active in the development of commercial products for emerging trend detection.

We have focused on research efforts that identify trends based primarily on the use of words and phrases; however, several research groups are using a different approach. The works in [CC99], [PFL+00] and [Ley02] present algorithms that primarily employ citation information for trend detection.

Several ETD methods focus more on the visualization of textual data and can be adapted to trend detection at the discretion of the user. For examples, Envision [NFH+96], allows users to explore trends graphically in digital library metadata (including publication dates) to identify emerging concepts. It is basically a multimedia digital library of computer science literature, with full-text searching and full-content retrieval capabilities. The system employs the use of colors and shapes to convey important characteristics of documents. For example, the interface uses color to show the degree of relevance of a document.

Plaisant et al. describe a visual environment called Lifelines for reviewing personal medical histories in [PMS+98]. The visualization environment presented in their work exploits the timeline concept to present a summary view of patient data.

Lavrenko et al. develop a method named EAnalyst [LSL+00] for analyzing two types of data, textual and numeric, both with time stamps. The system predicts trends in numeric data based on the content of textual data preceding the trend. For example, the system predicts the trend in stock prices based on articles published prior to the appearance of the (numeric) trend.

2.5 Summary

In this chapter, we have described previous work related to topic emerging trend detection. We described how other works represent topics, extract features and verify topics.

In both semi-automatic and fully-automatic methods, the effectiveness completely

depends on how appropriately a topic is represented in computers, how well the features associated with a topic are extracted from the documents and how reasonably the method for verifying topics are constructed. Because these tasks is performed on the text in the corpus, it is difficult to find the best model for any kind of textual data.

When applying existing methods to scientific corpora, many features of scientific article (such as citation information, influence, author reputation, journal/proceedings) are not represented and extracted. This makes these methods lack of rich representation for topic and good interpretation for the interest and utility measures.

How to build an appropriate ETD model for scientific corpora is the main challenge for our research. The key idea for solving this problem is representing useful features as much as possible, using automatic text-processing techniques effectively and utilizing available knowledge source such as WordNet [Wor]. The technical details will be described in next chapters.

Chapter 3

An ETD model for Scientific Corpora and Topic Representation

In this chapter, we present a new model for detecting emerging trends from scientific corpora and its topic representation method. We will discuss details of model components and the reason why we use this representation for topics in the following sections.

3.1 The Model Structure

Our ETD model is used for developing a fully-automatic emerging trend detection method, which takes a scientific corpus D as the input and produces a set of emerging trends E. The framework can be described as follows:

At the beginning of the ETD process, we analyze the input corpus D to extract topics from all documents. After that, these topics are then organized in a concept hierarchy T. Each topic is associated with a time-series of some temporal features defined in the *topic representation* module (TR) inside the model. These features are extracted from the corpus using text-processing techniques implemented in *topic identification* module (TI). At the final stage, topics and theirs features is sent to the *topic verification* (TV) module to identify emerging trends using two interest and

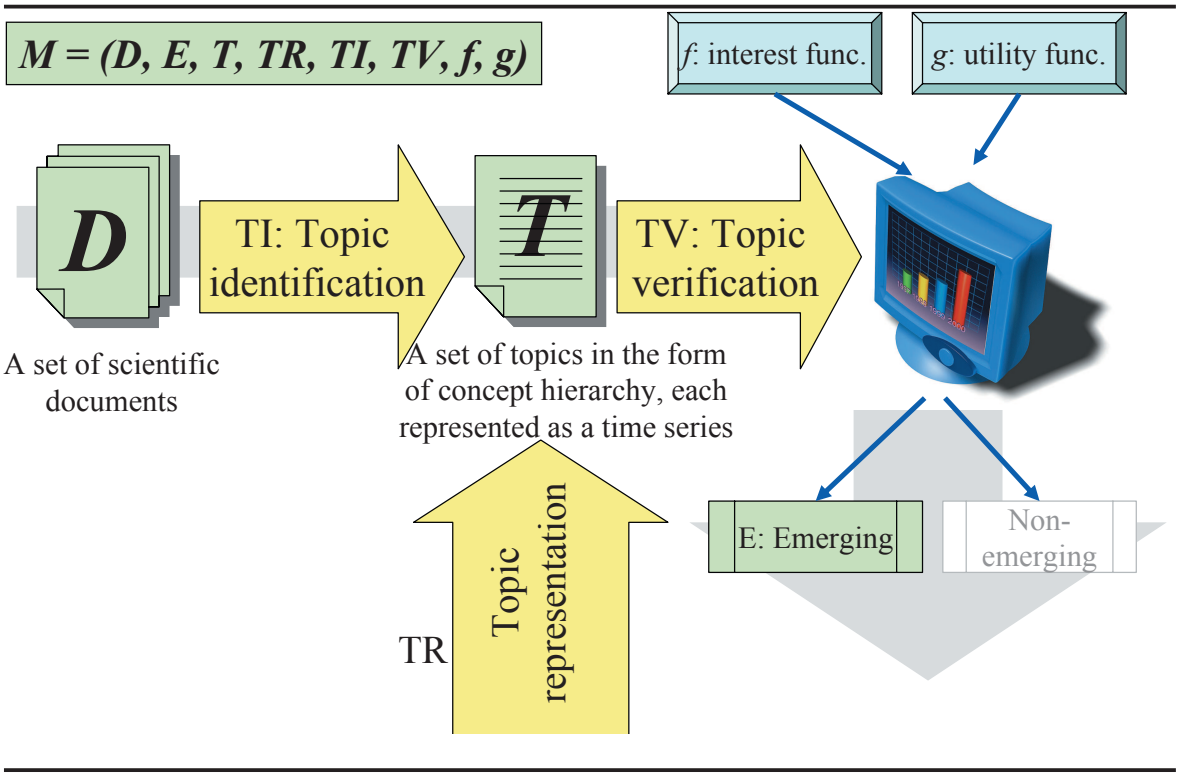

Figure 3.1: An ETD model for scientific corpora

utility functions (f and g). The ETD process in this model is shown in Figure 3.1.

In summary, the model structure consists of:

$$M = \{D, E, T, TR, TI, TV, f, g\}$$

where:

- $D = \{d_j\}$: The input: a set of scientific articles.
- E: The output: a set of output emerging trends
- $T = \{t_i\}$: A set of topics organized in the form of concept hierarchy.
- TR: The topic representation module.
- TI: The topic identification module.
- TV: The topic verification module.
- $f(.)$: The measure of growth in interest
- $g(.)$: The measure of growth in utility

3.2 Building the Topic Hierarchy

Given an input corpus D consisting of scientific articles, we want to detect topics that are mentioned in D and organize them in the form of a hierarchy T. The method for building the concept hierarchy T can be described as follows:

We extract noun phrases using the method described in [Bri92]. Based on the parts of speech tagging technique, this method includes the use of both lexical and contextual rules for identifying various parts of speech. The noun phrases extracted by the method compose of multiple modifications, including gerund verb forms [PY01].

The $tf \star idf$ measure [SC73] is then used for selecting keywords from these noun phrases. The keywords provided by authors are selected regardless their $tf \star idf$ ranking. We model the keyword co-occurrence by a weighed directed graph, where each node is a keyword and the weight of arc from node kw_i to node kw_j is the probability of occurrence of keyword kw_i in a document containing keyword kw_j:

$$c\,(i,j) = P\,(kw_i|kw_j)$$

The algorithm to build the topic hierarchy can be described as follows:

Step 1:

The weights of all arcs of the graph are normalized.

Step 2:

Arcs of weights smaller than a certain threshold τ are virtually pruned. Note that an arc from keyword kw_i to keyword kw_j can be pruned while the arc back from kw_j to kw_i still remains.

Step 3:

The Tarjan algorithm [Tar72] are performed on the graph to find strongly connected components.

Step 4:

Keywords in a strongly connected component are grouped into a topic. Arcs between topics form a directed acyclic graph (DAG).

Step 5:

For each topic t_j. We only keep arcs come from the topic t_i that has maximum relation weight to the current topic, i.e, the sum of weights on all arcs from t_i to t_j is maximum.

Step 6:

The remained arcs between topic form a hierarchical structure.

3.3 Topic Representation

Given a trial period of length Δ years, we represent each topic t_i in T by a time series: $t_i = \left(t_i^1, t_i^2, \ldots, t_i^\Delta\right)$ where t_i^k represents t_i in the k^{th} year in the trial period. Each t_i^k is associated with 6 parameters:

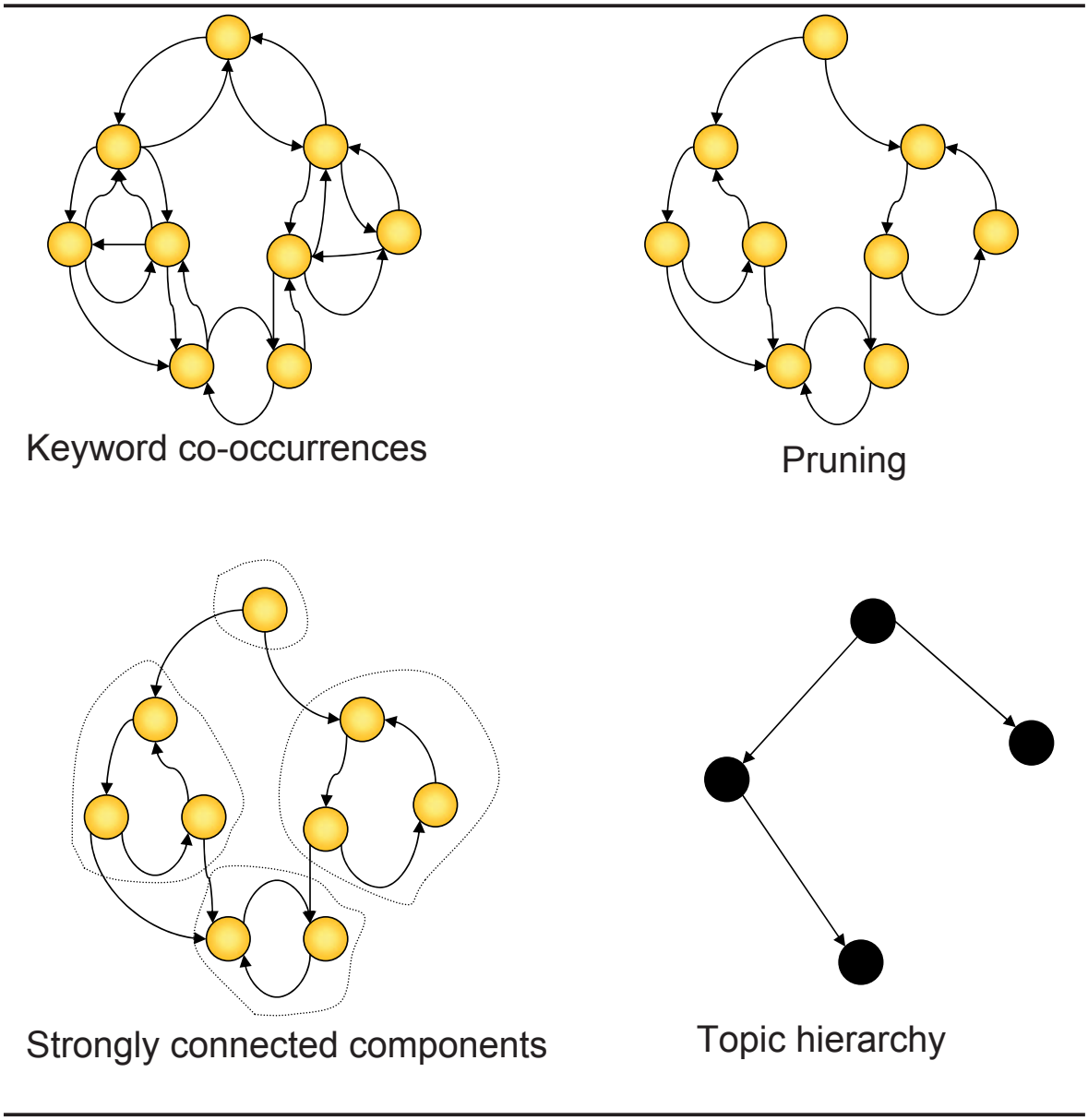

Figure 3.2: Building the topic hierarchy

t_i=NNs	1998	1999	2000	2001	2002	2003
$t_i^k\,(1)$	0.06	0.10	0.08	0.10	0.09	0.06
$t_i^k\,(2)$	0.20	0.33	0.28	0.06	0.11	0.04
$t_i^k\,(3)$	0.41	0.40	0.50	0.12	0.07	0.32
$t_i^k\,(4)$	0.17	0.40	0.06	0.12	0.33	0.02
$t_i^k\,(5)$	0.65	0.55	0.13	0.24	0.67	0.11
$t_i^k\,(6)$	0.33	0.44	0.22	0.33	0.44	0.56

Table 3.1: Parameters associated with the topic "neural networks".

- $t_i^k\,(1)$: determines how often the topic t_i is mentioned in the k^{th} year
- $t_i^k\,(2)$: the weight of citations in the k^{th} year to t_i, in which t_i is cited for referring to a theoretical basis, using methods or making comparison.
- $t_i^k\,(3)$: the number of citations in the k^{th} year to t_i
- $t_i^k\,(4)$: the influence of t_i on other topics in the k^{th} year
- $t_i^k\,(5)$: the weight of author reputations of t_i in the k^{th} year
- $t_i^k\,(6)$: the weight of sources (journals/proceedings) talking about t_i in the k^{th} year

In existing ETD models, a topic is often represented by n-grams, term frequencies, and term co-occurrences [PY01, SA00] associated with date tags, author names, citations [Gev02, RT01], etc. In our model, we do not only consider individual topics, but also view each topic in its relation to other topics in order to examine its change in interest and utility over time. For this purpose, each topic is organized in the concept hierarchy and associated with many features extracted from scientific articles. This gives our model a richer representation scheme for topics so as to compute the growth in interest and utility more reasonably.

Table 3.1 shows parameters associated with the topic "neural networks" extracted from our database. The tendency of each parameter is displayed in Figure 3.3.

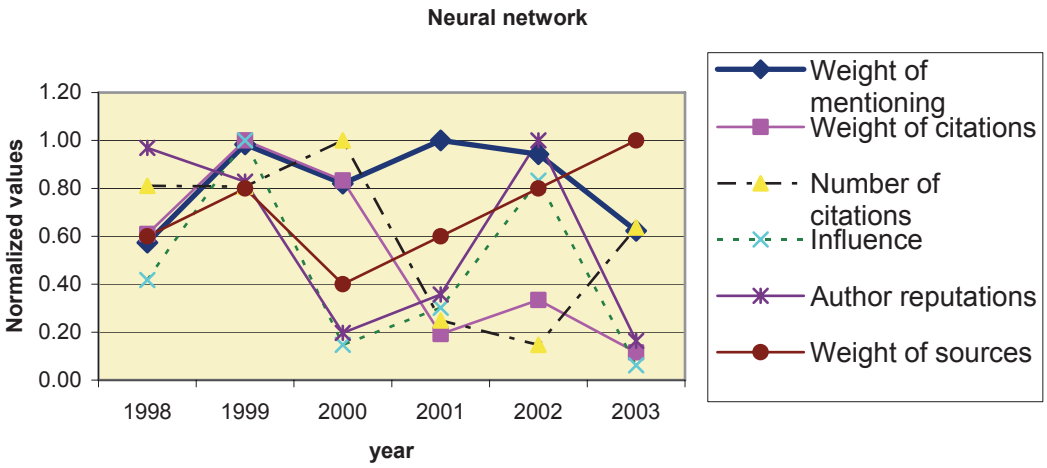

Figure 3.3: The tendencies of parameters associated with the topic "Neural networks".

3.4 Summary

In this chapter, we have described the structure of an ETD model for scientific corpora, which is used for developing a fully-automatic emerging trend detection method.

The model takes a scientific corpus as the input and produces the output as a set of emerging trends. No user interaction is modeled inside the detection process in order to deal with large corpora. We have introduced a method to build the topic hierarchy, which can be independently developed with other model components.

Topic representation is an important part of ETD models, which controls the behaviors of all other tasks in an ETD process. However, which features reflect the interest and utility of a topic? There is no perfectly precise definition for those features because evaluation of interest and utility is subjective and depends much on user opinions.

How researchers evaluate an emerging research topic? An emerging research topic should be mentioned many times in recent years, its theories and methods must have wide range of applications. Due to its usefulness for the development of other topic areas, it is usually talked/written by influential researchers, in many important scientific resources.

According to the above view of emerging research topic, we attempt to model these

criteria by associating each topic with its weight of mentioning, citation information, influence, author reputations and weight of sources. This representation scheme enables us to compute the interest and utility more reasonably, even if the topic representation module need to be improved, we need not to re-construct the model structure, need not to modify other modules so much because they are partially independent.

Chapter 4

Topic Identification

Topic identification is to fill up the representation of each topic by feature values. This chapters presents methods for extracting features associated with each topic as described in Chapter 3. We will focus on two main issues: topic detection and citation type detection. The computations of other parameters are also discussed.

4.1 Topic Detection

Topic is the most frequently used, unexplained, term in the discourse analysis literature. In the topic detection research, the definition of topic is usually simplified to discourse topic, which is defined as "what is being talked/written about" [BY83].

The first task of feature extraction is identifying topics of a given document in order to compute the weight of mentioning a topic in a year. In context of text mining, this problem – called topic detection [1]– is a very important part of automatic text processing techniques, such as information retrieval, text categorization, text summarization, etc.

Many methods of topic detection have been developed, which can be divided into three groups: statistical methods, knowledge-based methods, and hybrid methods. Statistical methods [SC73, DDL+90] infer the topic in the text from term frequencies, term

[1]The preferred name of this problem is *"topic identification"*, but in context of our model, a topic has richer representation and the topic identification task has to do more than identifying the topic name, so we use another name to distinguish them

locations, term co-occurrences, etc. without using external knowledge bases whereas knowledge-based methods [DeJ82, Leh82, RL94] rely on a syntactic/semantic parser, machine-readable dictionaries, etc. Hybrid methods [LM92, Hea94] combine statistical and knowledge-based advantages to improve the robustness of the identification process.

Statistical techniques do not rely on knowledge-intensive resources and parsing are usually faster, more reliable, and more robust than knowledge-based methods. However, their lack of deep understanding results in lower accuracy of the systems. On the other hand, knowledge-based techniques normally require enormous human effort to build the necessary knowledge. Nonetheless, the effectiveness of the invested human effort is demonstrated in the high accuracy over the intended domains.

We want to combine the advantages of both statistical and knowledge-based methods to achieve simultaneous high performance and cost-effectiveness. To this end, we describe a method of performing topic counting and generalization. By setting appropriate cutoff values for such parameters as concept generality and child-to-parent frequency ratio, we control the amount and level of generality of topics extracted from the text.

4.1.1 Word Frequency and Word Significance

Associating word frequency, i.e., word counting, with word significance was proposed in Luhn's pioneer work in automatic indexing [Luh57] and extracting [Luh58]. His proposal was based on the following assumptions:

1. Writers of a paper often emphasize an aspect of a subject through the repetition of certain words.

2. Writers usually use one sense of a word throughout a text.

3. Only a limited number of words are available to express a particular concept, even though writers might choose different words for the same concept for stylistic

reasons.

The first assumption enables one to use word frequency to estimate word significance without resorting to linguistic analysis (such as syntactic or semantic methods) that are expensive to implement and not robust enough even at today's scale of technology. The second assumption allows the reader not to be confusing of word-sense ambiguities. The third assumption can be addressed by using a thesaurus, which can be automatically acquired if enough sample texts are available.

Luhn also recognized that some high frequency words, such as the closed-class words the, a, in, to, were too common to be significant. He set up a high cutoff which filtered out high frequency common words, and a low cutoff which eliminated insignificant low frequency words. Words between these two cutoffs he considered as possessing "resolution power" (the ability of words to discriminate text contents). The two cutoffs were determined experimentally.

Extending Luhn's idea of insignificant common words to a complete text collection instead of just one document, Sparck Jones [Jon88] proposed a new word significance assignment scheme called inverse document frequency (idf) as follows:

$$idf = \log\left(\frac{N}{n}\right) + 1$$

where N is the number of documents in the collection and n is the number of documents in which the word occurs. The idf is smallest, i.e., most insignificant, for words occur in every document. Such words have no discrimination power over the collection. For words occuring only once in the entire collection, idf is maximal. Documents containing such words can be uniquely identified by the presence or absence of these words. Inverse document frequency is a very simple and useful term significance measure. It has been used in conjunction with Luhn's original idea of within-document term frequency and in much other Information Retrieval research.

Salton and Yang [SC73]'s combined within-document word frequency tf and inverse document frequency idf into a new term weighting scheme, which we now call $tf \star idf$.

They showed significant performance improvement over within-document frequency alone in information retrieval tasks. They also tried 287 different term weighting assignment methods, and reconfirmed that tf * idf has best performance.

Although term weighting schemes such as tf, idf, and tf * idf have been well developed and applied in many practical cases, it has been criticized in the following aspects:

First: Underlying the use of word frequency is the assumption that the more a word is used in a text, the more important it is in that text. This method recognizes only the literal word form and nothing else. Some morphological processing may help, but pronominalization and other forms of coreferentiality generally defeat simple word counting.

Second: Lexical ambiguity of words undermines word counts. For example, the frequency of the word "bank" is counted as 3 in "I bank my money in the bank on the bank of the Mississippi" is not really correct.

Third: Straightforward word counting can be misleading since it misses conceptual generalizations. For example, "I bought some apples, oranges, and grapes". What would be the topic of this sentence? We can draw no conclusion by using word counting method; where the topic actually should be: "I bought some fruits". The problem is that the word counting method does not consider semantic relations among these words.

We extend word frequency methods to incorporate knowledge about relations among words recorded in knowledge bases. As mentioned above, a weakness of word counting is that we cannot capture the deeper relations among words: two words may have very different spelling but may be very close semantically. So we need to count not word frequencies but topic frequencies.

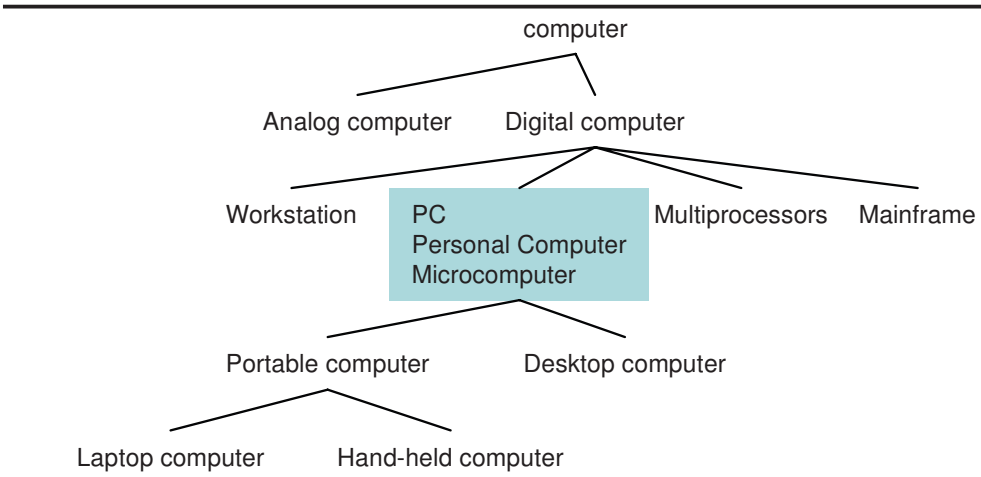

Figure 4.1: The topic hierarchy.

4.1.2 Topic Counting

In our ETD model, the set of topics T is already organized in the form of a concept hierarchy. Each topic is a set of synonymic terms and the hierarchical structure is based on hyponymy relationships between topics. Figure 4.1 is an example of the topic "computer" and its sub-topics.

Given a topic $t_i \in T$ and a document $d_j \in D$, we scan the entire document to count how many times a topic is mentioned, whenever a topic is counted, its parent topics are also counted. Let Count (t_i, d_j) be the number of times we count the topic t_i while scanning the document d_j.

4.1.3 Topic Selection

When scanning a document to count the number of mentioning a topic, we have to match each word (or n-grams) to the set of terms represented for the topic. If the word (or n-grams) matches two or more topics (polysemy mapping), we have to choose the corresponding topic for counting.

To this end, when scan to a word (or n-grams) w in which the word-sense ambiguity happened. We add w into a queue for later processing and continue until the entire document is scanned.

For a word w in the queue, we check all topics which have a term matched this word. The word w is then matched to the topic having highest sum of counters along the path from root topic to this topic:

$$t = \arg\max_{w \in t_i} \sum_{t_k \supset t_i} c\left(t_k, d_j\right) \tag{4.1}$$

where d_j is the document, t is the topic matched the word (n-grams) w that have to be identified, t_i is a topic that has a term matched w, t_k is a hypernym of t_i.

4.1.4 Generalization

The power of the concept frequency method lies in generalization. For example, we may have the following three sentences in a text:

S1: Desktop computer prices fall.

S2: Laptop computer prices rise.

S3: Hand-held computer prices fall.

The word "prices" appears 3 times; "fall" 2 times; "Desktop computer", "Laptop computer", "Hand-held computer", and "rise" once. But if we count topics, the topic "PC" appears 3 times; "prices" 3 times; "change" 3 times. Thus these three sentences could be generalized as:

S4: PC prices change.

Now a second problem presents itself: though we may find all the times sub topics are mentioned, we would also like to know that these topics can be combined into one topic so that we can identify topic of texts using the generalization rather than the particular.

To this end, we define the *branch ratio* of a topic:

$$R\left(t\right) = \frac{\max\left(\text{weight of all the immediate children}\right)}{\sum\left(\text{weight of all immediate children}\right)} \tag{4.2}$$

where weight is the number of times a topic is mentioned in the text (for a leaf node), or the sum of the weights of all immediate children (for non-leaf nodes).

It is obvious that the ratio is 1.0 if only one topic is mentioned in the source; while it is 0.0 for any topic not mentioned in the source. We found that the definition of ratio, $R(t_i)$, is a way to identify the degree of generalization. The higher the ratio is, the less generalization power a parent node has over its immediate children. In our ETD model, the degree of generalization is selected according to the need of users for the degree of generalization of the output emerging trends.

4.1.5 Evaluation

We designed an experiment, in which we selected 100 topics into the concept hierarchy T. For each article, we identified topics from the full text and from its abstract to compute three counts:

hits : number of topics that are identified from full papers and also identified from their abstracts.

mistakes : number of topics that are identified from full papers but are not identified from their abstracts.

misses : number of topics that are not identified from full papers but are identified from their abstracts.

We then borrowed two measures from Information Retrieval:

Recall : hits/(hits + misses)

Precision : hits/(hits + mistakes)

The closer these two measures are to unity, the better the algorithm's performance. We randomly selected 100 papers for testing and achieved values of 0.52 and 0.58 in Recall and Precision respectively. When we added the keyword provided by the authors to the set of keywords extracted using tf $\star$ idf, the accuracy was much improved: 0.82 in Recall and 0.87 in Precision.

4.1.6 Computing the Weight of Mentioning a Topic in a Given Year

The relevance of the document d_j to the topic t_i is computed as:

$$r(i,j) = \frac{\text{Count}(t_i, d_j)}{\sum_{t_k \in T} \text{Count}(t_k, d_j)} \tag{4.3}$$

To determine how often the topic t_i is mentioned in the k^{th} year, we sum up all relevances of documents published in the k^{th} year to t_i:

$$t_i^k(1) = \sum_{\text{year}(d_j)=k} r(i,j) \tag{4.4}$$

4.2 Citation Type Detection

Citations appear very frequently in scientific articles and most of digital libraries now organize their papers in the structure of citation indexes [Sma73]. By examining the citations inside an article, we can reveal relationships between articles, draw attention to important corrections of published work and identify significant improvements or criticisms of earlier work [LGB99, KdRH+01]. However, this is still very difficult for researchers because the large and increasing number of articles prevents them from reading everything in the published literature. There is a clear need for new tools to identify the types of citation relationships that indicate the reasons for citation in a human-understandable way [Gev02].

The purpose of identifying the reasons for citations (citation type detection - CTD) varies according to the main objective of each research. The method of Nanba and Okumura [NKO00] uses an heuristic sentence selection and pre-defined cue phrases to classify citations into three categories for supporting a system of automatic review articles. To extend the usage of linguistic patterns, Teufel [Teu99] uses formulaic expressions, agent patterns and semantic verb classes instead of cue phrases to determine

the corresponding class for a sentence. Although both these works show the usefulness of linguistic patterns in citation type detection, the manual construction of linguistic patterns is obviously a rather time-consuming task. It also involves some conflicts that are difficult to be resolved. For example, the method of Pham and Hoffmann [PH03] has to eliminate such conflicts and send to human experts for providing rules that resolved them.

The available methods do not appear to be integrated into an ETD process because of two main limitations: the first is their definitions of citation types are not appropriate for evaluating the interest and utility of topics; the second reason is the manual construction of linguistic patterns must depend on the corpus. This makes the detection process become inflexible when applying to other corpora. We want to develop an automatic method for detecting citation types, that have a clear definition for citation types which support the detection of emerging trends by tracing the development of a topic and clarify the relationship between articles. In addition, the method must be able to detect citation types without any need for user-interactions or explicit knowledge about linguistic patterns as were required in [NKO00, Teu99, PH03].

In the following sections, we will describe our definition of citation types. After briefly summary the basic concepts of two kinds of finite-state machines (FSM): hidden Markov models (HMMs) and maximum-entropy Markov models (MEMMs), we will introduce the method for evaluating sentences and classification of citing area. Experiments and evaluations are given in the last section.

4.2.1 Definition of Citation Types

Given a paragraph containing citations (we call this paragraph the citing area), we want to detect why the cited paper is mentioned according to the authors. It is well known that there are many reasons for citations (citation types). To classify citing areas using citation relationships, we also have to consider the citation types. For example, in [Wei71], Weinstock proposed 15 categories for the common reasons of

citations, to build a system for the automatic generation of review articles, Nanba and Okumura [NKO00] classified the reasons for citations into three categories while Pham and Hoffmann [PH03] used four types of citations for building a citation map between articles.

In order to support researchers in tracing the development of a topic over time as well as clarify the relationship between articles, we classified citation types into the following six main categories (or classes), which are important for emerging trend detection:

Type I: The paper is based on the cited work; it means that the citation shows other researchers' theories or methods as the theoretical basis for the current work. (corresponding to Nanba's type B)

Type II: The paper is a part of the cited work

Type III: The cited work supports this work

Type IV: The paper points out problems or gaps in the cited work (corresponding to Nanba's type C, Pham's type Limitation)

Type V: The cited work is compared with the current work

Type VI: Other citations

Note that these classes are overlapping, meaning that a citation area may belong to two or more classes. We will choose the most suitable class label for a citation area and also measure the likelihood of each citing area on a class.

4.2.2 Hidden Markov Models

A hidden Markov model (HMM) is a finite-state automaton with stochastic state transitions and observations whereby a sequence of observations is emitted along the transitions of states over time [Rab89]. A HMM $\lambda = (A, B, \Pi)$ is defined on a set of n states $S^\lambda = (s_1, s_2, \ldots, s_n)$, a set of possible observations O^λ and three probability

distributions: a state transition probability to $s_j \in S^\lambda$ from $s_i \in S^\lambda$: $a_{ij} = P(s_j|s_i)$; an observation probability distribution $b_j(o) = P(o|s_j)$ for $o \in O^\lambda$, $s_j \in S^\lambda$; and an initial state distribution for each state $s_i \in S^\lambda$: $\pi_i = P(q_1 = s_i)$.

Although initially introduced and studied in the late 1960s and early 1970s, statistical methods of Markov source or hidden Markov modeling have become increasingly popular in the last several years. There are two strong reasons why this has occurred. First the models are very rich in mathematical structure and hence can form the theoretical basis for use in a wide range of applications. Second the models, when applied properly, work very well in practice for several important applications.

The process of observation emission in a HMM λ can be described as follows: First, the model choose an initial state $q_1 = s_{i_1} \in S\lambda$ according to the initial state probability π_{i_1}. From this state, an observation symbol o_1 is randomly emitted according to the observation probability distribution $b_{i_1}(o_1)$. After that, a new state $q_2 = s_{i_2} \in S\lambda$ is selected according to the transition probability $a_{i_1 i_2}$ and the observation emission process is repeated. By following this process up to the time $t = T$, the observation sequence $O = (o_1, o_2, \ldots, o_T)$ is generated by the hidden process of transiting model state along the sequence $Q = (q_1, q_2, \ldots, q_T)$.

Computing the probability of generating an observation sequence

Given a time-length T, A HMM λ can produce any observation sequence with different probabilities. The problem of computing the probability that the observation sequence was produced by the model can be viewed as the problem as one of scoring how well a given model matches a given observation sequence. This viewpoint is extremely useful, for example, if we consider the case in which we are trying to choose among several competing models, the solution to this problem allows us to choose the model which best matches the observations. One of effective solutions for this problem is the Forward-Backward procedures which were shown in Figure 4.2.

Forward, Backward procedures

Input: A HMM λ and an observation sequence $O = (o_1, o_2, \ldots, o_T)$.
Output: The probability of O given the model λ.

1. Define forward variables and backward variables:

$$\alpha_t(i) = P(o_1, o_2, \ldots, o_t, q_t = s_i|\lambda) \quad (4.5)$$
$$\beta_t(i) = P(o_{t+1}, o_{t+2}, \ldots, o_T|q_t = s_i, \lambda) \quad (4.6)$$

2. Initialization:

$$\alpha_1(i) = \pi_i.b_i(o_1), \quad (1 \leq i \leq n) \quad (4.7)$$
$$\beta_T(j) = 1, \quad 1 \leq j \leq n \quad (4.8)$$

3. Induction:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{n} \alpha_t(i).a_{ij}\right].b_j(o_{t+1}), \quad (1 \leq j \leq n, 1 \leq t \leq T-1) \quad (4.9)$$
$$\beta_t(i) = \sum_{j=1}^{n} a_{ij}.b_j(o_{t+1}).\beta_{t+1}(j), \quad (1 \leq i \leq n, T-1 \geq t \geq 1) \quad (4.10)$$

4. Termination:

$$P(O|\lambda) = \sum_{i=1}^{n} \alpha_T(i) \quad (4.11)$$
$$= \sum_{j=1}^{n} \pi_j.b_j(o_1).\beta_1(j) \quad (4.12)$$

Figure 4.2: Computing the probability of an observation sequence

The Viterbi algorithm

The viterbi algorithm is a dynamic programming algorithm for finding the most likely sequence of hidden states known as the Viterbi path that result in a sequence of observed events, especially in the context of hidden Markov models. The forward algorithm is a closely related algorithm for computing the probability of a sequence of observed events. Its a subset of a wider topic known as information theory.

Given a HMM λ and an observation sequence $O = (o_1, o_2, \ldots, o_T)$, the Viterbi algorithm [Vit67] is used for finding the single best state sequence $Q^O = \left(q_1^O, q_2^O, \ldots, q_T^O\right)$ for the sequence O:

$$Q^O = \arg\max_Q P\left(O, Q|\lambda\right) \tag{4.13}$$

Details of the Viterbi algorithm is shown in Figure 4.3.

The Viterbi Training Algorithm

Traning a HMM is one attempt to optimize the model parameters so as to best describe how a given observation sequence comes about. The observation sequence used to adjust the model parameters is called a training sequence since it is used to train the HMM. The training problem is the crucial one for most applications of HMMs, since it allows us to optimally adapt model parameters to observed training data-i.e., to create best models for real phenomena.

The standard method to train HMMs is the EM algorithm, also known in HMM context as the Baum-Welch algorithm [Rab89]. However, we use the Viterbi training (VT) algorithm instead of EM to avoid expensive computation in practice. The VT algorithm just takes the single most likely path and maximize the probability of emitting the observation sequence along its corresponding path. This can be described as in Figure 4.4.

The Viterbi algorithm

Input: A HMM λ and an observation sequence $O = (o_1, o_2, \ldots, o_T)$.
Output: The state sequence Q^O that maximized $P\left(O, Q^O|\lambda\right)$.

1. Definition:

$$\delta_t(i) = \max_{q_1, q_2, \ldots, q_{t-1}} P(q_1, q_2, \ldots, q_{t-1}, q_t = s_i, o_1, o_2, \ldots, o_t|\lambda) \tag{4.14}$$

2. Initialization:

$$\delta_1(i) = \pi_i.b_i(o_1), \quad 1 \leq i \leq n \tag{4.15}$$
$$\psi_i(i) = 0 \tag{4.16}$$

3. Recursion:

$$\delta_t(j) = \max_{1 \leq i \leq n} (\delta_{t-1}(i).a_{ij}).b_j(o_t), \quad (2 \leq t \leq T, 1 \leq j \leq n) \tag{4.17}$$
$$\psi_t(j) = \arg\max_{1 \leq i \leq n} (\delta_{t-1}(i).a_{ij}), \quad (2 \leq t \leq T, 1 \leq j \leq n) \tag{4.18}$$

4. Termination:

$$P^*(O|\lambda) = \max_{1 \leq i \leq n} (\delta_t(i)) \tag{4.19}$$
$$k_T = \arg\max_{1 \leq i \leq n} (\delta_t(i)) \tag{4.20}$$

5. Back tracing:

$$k_t = \psi_{t+1}(k_{t+1}), \quad (T - 1 \geq t \geq 1) \tag{4.21}$$

6. Finalization:

$$q_t^O = s_{k_t}, \quad (1 \leq t \leq T) \tag{4.22}$$

Figure 4.3: The Viterbi algorithm

The Viterbi training algorithm

Input: n sentences $O^1, O^2, ..., O^\theta$, in which $O^k = \left(o_1^k, o_2^k, \ldots, o_T^k\right)$, where each o_t^k is a word or a citation
Output: The model parameters

1. Initialization: Randomly assign a label $l \in \{1, 2, ..., n\}$ for each observation symbol occurred in training sequences.
2. Definition:

$$c_t^k(i) = \begin{cases} 1, \text{if the label } i \text{ is assigned to } o_t^k \\ 0, \text{otherwise.} \end{cases} \tag{4.23}$$

3. Estimate model parameters:

$$\pi_i = \frac{\sum_{k=1}^{\theta} c_1^k(i)}{\theta}, (1 \le i \le n) \tag{4.24}$$

$$a_{ij} = \frac{\left\|\left\{(t,k) \left| c_t^k(i) = c_{t+1}^k(j) = 1 \right.\right\}\right\|}{\left\|\left\{(t,k) \left| (t < T) \wedge \left(c_t^k(i) = 1\right) \right.\right\}\right\|}, (1 \le i, j \le n) \tag{4.25}$$

$$b_j(o) = \frac{\sum_{(t,k):o_t^k = o} c_t^k(j)}{\sum_{(t,k)} c_t^k(j)} (1 \le j \le n), o \in O \tag{4.26}$$

4. Re-alignment: For each observation sequence O^k, find the corresponding state sequence $Q\left(O^k\right) = (q_1, q_2, ..., q_T)$ by Viterbi algorithms, then re-assign the label i to O_t^k if $q_t = s_i$
5. If there is any change in step 4, go back to step 2, otherwise stop

Figure 4.4: An outline of the Viterbi training algorithm

4.2.3 Maximum-Entropy Markov Model

The structure of maximum-entropy Markov models (MEMMs) is similar to that of hidden Markov models, but instead of transition and observation probabilities, we have only one single function $P(s|s', o)$ which provides the probability of the current state s given the previous state s' and the current observation o. This complex function is often separated into $\|S\|$ transition functions $P_{s'}(s|o)$. In contrast to HMMs, in which the current observation only depends on the current state, in MEMMs, the current observation may also depend on the previous state. It means the observations is associated with state transition rather than with states [MFP00].

In MEMMs, each transition function $P_{s'}(s, o)$ is often represented in the exponential form:

$$P_{s'}(s, o) = \frac{1}{Z(o, s')} \exp\left(\sum_a \lambda_a f_a(o, s)\right) \tag{4.27}$$

where f_a is a feature, λ_a is a parameter to be learned and $Z(o, s')$ is the normalizing factor that makes the distribution sum to one across all next state s.

The Viterbi algorithm in context of MEMMs

Despite the differences between MEMMs and HMMs, there is still an efficient dynamic programming solution to the classic problem of identifying the most likely state sequence given an observation sequence. In context of MEMMs, we redefine $\delta_t(i)$ to be the probability of being in state s_i at time t given the observation sequence up to time t. The recursive step in the Viterbi algorithm is then:

$$\delta_t(j) = \max_{1 \le i \le n} \left(\delta_{t-1}(i) . P_{s_i}(s_j|o_{t-1})\right), (2 \le t \le T, 1 \le j \le n) \tag{4.28}$$

$$\psi_t(j) = \arg\max_{1 \le i \le n} \left(\delta_{t-1}(i) . P_{s_i}(s_j|o_{t-1})\right), (2 \le t \le T, 1 \le j \le n) \tag{4.29}$$

Parameter Estimation by Generalized Iterative Scaling

Generalized Iterative Scaling (GIS) [DR72] is an iterative algorithm for finding the λ_a values that form the maximum entropy solution for each transition function (Equation

Input: Training sequences: There are n observation sequences $O^1, O^2, ..., O^\theta$
Output: The model parameters

1. Start with an arbitrary model structure
2. For each training sequence $O = (o_1, o_2, ..., o_T)$, determine the state sequence $Q(O) = (q_1, q_2, ..., q_T)$ associated with observation sequence O using Viterbi algorithm
3. For each state-observation pairs $(s_t,\ o_t)$, deposit it into their previous state s_{t-1} as training data for the transition function $P_{s'}(s|o)$
4. Find the maximum entropy solution for each state transition function $(P_{s'})$ by GIS.

Figure 4.5: An outline of the training algorithm of a maximum-entropy Markov model

4.27).

To train a MEMM, we first split the training data into (state-observation) pairs relevant to the transitions from each state s', then apply the Generalized Iterative Scaling method (GIS) [DR72] to estimate the transition function for state s' (f'_s), Details of the training algorithm for MEMM are described as in Figure 4.5:

4.2.4 Citation Type Detection Using Finite-State Machines

In this section, we describe the method for detecting citation types. The detection process can be described as follows: Given a citing area consisting of several sentences, we apply finite-state machines to compute the likelihood of each sentence on each class. After that, we evaluate the importance of each sentence and combine these values to identify the corresponding class for this citing area. We present here two methods to evaluate the above likelihood using hidden Markov models and maximum-entropy Markov models, after that we will introduce the sentence-weighting strategy to identify class label for a given citing area.

Sentence Evaluation

In most text-processing tasks using HMMs and MEMMs, people often use word-based models, i.e, each word (or n-gram) is one observation. The main drawback of these methods is the machine cannot accept unknown observation symbols or accepts them with a very low probability of emission functions. For example, if we consider each English word as an observation, the model trained by the sentence "*The man walks so fast*" may produce 0 or a small value depending on the training algorithm when computing the likelihood of the sentence "*The man goes so fast*" even though the meaning of the second sentence can be implied from the semantics of the trained sentence. This problem occurs not only with finite-state machines, but also with all word-based methods.

One solution to this problem is enlarging the training set so as to cover all possible cases of synonymy and hyponymy. However, it is difficult to build a large training set and it also increases the complexity of training phase. For example, the method using cue phrases [NKO00] has to construct a very long list of cue phrases; the rule-based method [PH03] has to add many rules to the rule set in order to achieve high accuracy.

To overcome the drawback of the aforementioned solution, we still use word-based models, but after the training phase, we do some post-processing on model parameters for dealing with the problem of synonymy and hyponymy.

For a HMM, we re-adjust the emission functions to:

$$\bar{b}_j(o) = \max_{o' \subseteq o} b_j(o') \tag{4.30}$$

where $o' \subseteq o$ means the word o' is a hyponym or synonym of the word o.

For a MEMM, we first organize all word concepts in a concept hierarchy, in which each node in the hierarchy consists of a word and its synonyms and a sub-concept is represented by a descendant of its parent concepts. The synonymy and hyponymy

relationships between words are represented by feature functions of MEMMs:

$$f_{(c,q)}(w,s) = \begin{cases} 1, \text{if } (s=q) \wedge (w \in c) \\ 0, \text{otherwise} \end{cases} \tag{4.31}$$

where c represented for a concept, w is a word and $w \in c$ means the concept c accepts the word w as its synonym or hyponym.

There are six finite-state machines are used for evaluating sentence, each machine represents for a citation type, it accepts the set of English words including "\cite" as the its observations. We have a number of training sentences for each class. These sentences are used as the input of the training algorithm (Viterbi training algorithm for HMMs and Generalized Iterative Scaling algorithm for MEMMs) to estimate model parameters.

Given an unknown sentence O and six trained FSMs corresponding to six classes, we compute how well the sentence O matches these FSMs by calculating the probability of sentence O along its best path on each machine:

$$P^*(O|\lambda) = \arg\max_Q P(O,Q|\lambda) = P\left(O, Q^{(O)}|\lambda\right) \tag{4.32}$$

where $Q^{(O)}$ is the state sequence found by the Viterbi algorithm.

Weighting Sentences and Classification of Citing Areas

Consider a kind of finite-state machine, HMM or MEMM. We have a total of six machines $\{\lambda_i\}_{i=1}^6$ corresponding to six classes. Given an unknown sentence O, we find the best state sequence Q_i^O corresponding to O in each machine λ_i and compute the likelihood $P^*(O|\lambda_i) = P\left(O, Q_i^O|\lambda_i\right)$ to measure how closely the sentence O matches the machine λ_i.

A citing area might consist of many sentences; each sentence can match all six machines with different levels. We need to combine these likelihoods in order to determine which class is suitable for the entire citing area. To this end, we want to determine the

importance of each sentence in evaluating the citing area.

Given a sentence O, and a finite-state machine λ_i, we compute $P^*(O|\lambda_i)$ and define:

$$P^{(O)}(\lambda_i) = \frac{P^*(O|\lambda_i)}{\sum_{j=1}^{6} P^*(O|\lambda_j)} \tag{4.33}$$

as the probability of selecting the model λ_i given the sentence O. The entropy of this probability distribution is:

$$H^{(O)} = -\sum_{i=1}^{6} P^{(O)}(\lambda_i) \log_2 P^{(O)}(\lambda_i) \tag{4.34}$$

As the entropy $H^{(O)}$ becomes larger, the chance of selecting the model corresponding to sentence O becomes more uncertain, and the the role O plays in determining class label for the citing area becomes less important. Thus, we can weight each sentence O in the citing area by

$$Weight(O) = \frac{\log_2 6 - H^{(O)}}{\log_2 6}; (0 \leq Weight(O) \leq 1) \tag{4.35}$$

If the citing area C consists of m sentences: $O^1, O^2, \ldots O^m$. The corresponding citation type for this citing area is:

$$Type(C) = \arg\max_{1 \leq i \leq 6} \sum_{j=1}^{m} Weight\left(O^j\right) . P^*\left(O^j|\lambda_i\right) \tag{4.36}$$

To use citation types more flexibly, instead of assigning a class label for a given citing area, we can compute how closely a given citing area matches a category i by measuring the likelihood:

$$L(C|i) = \frac{\sum_{j=1}^{m} Weight(O^j) . P^*(O^j|\lambda_i)}{\sum_{i'=1}^{6} \sum_{j=1}^{m} Weight(O^j) . P^*(O^j|\lambda_{i'})} \tag{4.37}$$

Making a model that analyzes the entire citing area requires many complicated computations and a very large training set. Like other methods, our method segments the citing area into sentences and classifies it by evaluating the sentences. However, instead of selecting only one sentence for evaluating the whole citing area, we evaluate the likelihood of each sentence on each class, and use the weight of each sentence to combine these likelihoods in a reasonable way.

From theoretical viewpoint, it is worth noting that our method can be extended to deal with more citation types. It takes into account the problem of word synonymy and hyponymy, allows overlapping between classes and works without any user-interactions or pre-defined linguistic patterns. That can be viewed as a significant difference between our citation type detection method and previous works.

4.2.5 Experiments

We designed two experiments for two purposes: first, we want to evaluate if the model using FSMs is more appropriate than other methods using linguistic patterns in the task of detecting citation types; secondly we want to compare two methods using HMMs and MEMMs and discuss the advantages and drawbacks of each model in practice.

In the first experiment, we used the data set provided by Nanba and Okumura [NKO00] consisting of 282 citing areas for training and 100 citing areas for testing. Because the work of Nanba and Okumura used different class labels for citations, we redesigned our citation types so as to match their three types and make the comparison. The evaluation criterion is the accuracy of the classification result produced by each method.

In the second experiment, we manually labeled 811 citing areas from our collection of scientific papers. For a limited number of sentences in the training set, we manually selected training sentences from the dataset and did the experiment 10 times before taking average of accuracy.

The set of concepts is extracted from an external knowledge taxonomy: WordNet

	Nanba				HMMs				MEMMs			
	C	B	O	(%)	C	B	O	(%)	C	B	O	(%)
16 citations type C	12	0	4	75.0	14	0	2	87.5	14	0	2	87.5
32 citations type B	2	25	5	78.1	0	25	7	78.1	0	26	6	81.3
52 citations type O	1	5	46	88.5	3	1	48	92.3	1	1	50	96.1

Table 4.1: The accuracies of Nanba and Okumura's method, HMMs, and MEMMs

[Wor]. These experiments used HMMs and MEMMs with 25 states (This is the average of number of words in each sentence). Increasing the number of states may improve the classification results, but requires longer computational time in the training and testing phases.

Experiment 1

This experiment is used to evaluate if our method achieves higher accuracy compared to Nanba and Okumura's method when running in the same conditions. The data set provided by Nanba and Okumura in [NKO00] consists of 282 citing areas for training and 100 citing areas for testing. We use the same definition of citation types as they defined: B, C and O and select training sentences according to their sentence selection strategy. Table 4.1 shows the accuracy of Nanba and Okumura's method compared to our methods.

Running under the same conditions, our method using HMMs and MEMMs based on concept-representation achieve higher accuracy than Nanba's method. Although the set of cue phrases is well designed for this dataset, Nanba's method still has the problem of synonymy and hyponymy, that why our method using concept-representation can result in higher accuracy.

Experiment 2

This experiment is used to compare the performance of two methods using HMMs and MEMMs. To this end, we collect 9000 papers from two main sources: ACM Digital Library and Science Direct, and randomly select 811 citing areas for this experiment.

Number of training sentences	HMMs (%)	MEMMs (%)
100	60.1	61.4
200	67.1	67.2
300	72.6	73.8
400	79.9	79.6
500	84.9	86.6
600	90.4	91.8
700	95.2	95.9
800	99.5	99.7
811	100.0	100.0

Table 4.2: The accuracies of two methods using HMMs and MEMMs

We randomly selected a varying number of sentences for training from these 811 citing area and run the experiment 10 times before taking an average of accuracy. Table 4.2 shows the detection accuracies of the methods using HMMs and MEMMs when testing with all 811 citing areas.

The method using MEMMs produced slightly better result than HMMs as shown in Table 4.2. In addition, the method using MEMMs requires lower computation time for the training phase: it takes 7918 seconds for training MEMMs with 800 sentence compared to 20168 seconds taken by the VT algorithm. The main reason is not only the different characteristics of HMM training and MEMM training algorithms, but also because we must re-distribute the emission functions of HMMs to deal with the synonymy and hyponymy relationships between words while we can model these relations by feature functions of MEMMs.

4.3 Computing the Influence

Let S be a subset of T, t_i is a topic in T but not belonging to S. We want to compute the influence of the topic t_i on topics in S. First, we define $P(S)$ $(P(\overline{S}))$ is the probability of any topic in S being (not being) mentioned in an article. If the total number of articles is n and the number of articles mentioning any topic in S is m, then $P(S) = \frac{m}{n}$ and $P(\overline{S}) = \frac{n-m}{n}$.

The entropy of the occurrence of any topic in S is:

$$H(S) = -P(S)\log P(S) - P\left(\overline{S}\right)\log P\left(\overline{S}\right) \tag{4.38}$$

The entropy of the occurrence of any topic in S under the condition that the occurrence of topic t_i is known – $H(S|t_i)$ – can be calculated as follows:

$$H(S\,|t_i) = -\sum_{x=S,\overline{S}}\sum_{y=t_i,\overline{t}_i} p(x,y)\log p(x\,|y) \tag{4.39}$$

Now, we consider the mutual information:

$$I(S;t_i) = H(S) - H(S\,|t_i) \tag{4.40}$$

which reflects the reduction in uncertainty about S due to the the occurrence of t_i is known. The greater $I(S;t_i)$ is, the more influence of t_i on other topics in S.

Because the set of articles used to compute the occurrence of a topic changes overtime, we normalize the influence of t_i to the interval $[0,1]$:

$$t_i^k(4) = \frac{I(T\backslash\{t_i\};t_i)}{H(T\backslash\{t_i\})} \tag{4.41}$$

4.4 Summary

In this chapter, we have proposed several methods to extract features associated with each topic. A topic hierarchy and a topic counting strategy are used for identifying topics of a documents in order to measure the weight of mentioning a topic in a given year. Our citation type detection method using finite-state machines is more appropriate and accurate compared to other works.

The influence of a topic on other topics is also formulated and computed. Automatic methods for weighting author reputations and sources are under construction. Currently, the author reputations is simply assigned by the number of papers published

by each author and the weight of journal/proceedings are manually assigned for testing the model.

Chapter 5

Topic Verification and a Prototype System

5.1 Topic Verification

In our ETD model, the topic verification module takes a set of topics with theirs features as the input and produces the set of emerging trends. The interest and utility functions are integrated into this module for evaluating input topics.

Due to the difficulty of the topic verification task, existing work on emerging trend detection usually detect topic areas that have grown in size and variety at an increasing rate over time. We want to evaluate the interest and utility of each topic separately using these two measures in order to make the topic verification method more reasonable in classification of emerging trend.

5.1.1 The Measure of Growth in Interest

The interest is the power of attracting or holding one's attention because it is unusual or exciting. A research topic is said to be interest if it is novel and attractive. Therefore, it is recently mentioned many times by influential people, in important journals/conferences.

Evaluating the interest is to combine these above criteria in to on measure. That

why we represent topics in such a way we can efficiently evaluate the interest. By analyzing the time-series of features associated with each topic, we can track the novel of a topic as well as determine how often a topic is mentioned in the trial period, who mentioned it and where published it. In other words, to evaluate the growth in interest of a topic t_i in a given period, these following features must be considered:

- $\left\{t_i^k\,(1)\right\}_{k=1}^{\Delta}$: $t_i^k\,(1)$ determines how often t_i is mentioned in the year k^{th}. The change in value of $\left\{t_i^k\,(1)\right\}_{k=1}^{\Delta}$ along the time-series can be used for evaluating the change in attractiveness of researchers on t_i due to its significance, novelty, or challenge.

- $\left\{t_i^k\,(3)\right\}_{k=1}^{\Delta}$: $t_i^k\,(3)$ specifies the number of citations in the k^{th} year to the topic t_i. This can be viewed as another measure for research attractiveness. However, a topic having many citations might not be novel. It's attractive because it provided theoretical background and techniques supporting for later works, or specified problems or gaps in research context that need to be overcome.

- $\left\{t_i^k\,(5)\right\}_{k=1}^{\Delta}$: $t_i^k\,(5)$ is the weight of author reputations of the topic t_i in the year k^{th}. Actually, people cannot collect all papers talking about a topic, this feature enables us to evaluate the novelty and attractiveness of a topic by human experiences.

- $\left\{t_i^k\,(6)\right\}_{k=1}^{\Delta}$: $t_i^k\,(6)$ is the weight of sources (journals/proceedings) talking about t_i in the k^{th} year. As the same as the weight of author reputations, this feature is integrated into the model for evaluating the interest by explicit knowledge.

5.1.2 The Measure of Growth in Utility

In social problems, utility is a measure of the happiness or satisfaction gained from a product or service. Utility was originally viewed as a measurable quantity, so that it would be possible to measure the utility of each individual in the society with respect to each product or service. By adding individual utilities together to yield the total utility of all people with respect to all products or services, society could then aim to

maximize the total utility, or equivalently the average utility per person.

In context of research literature, researchers often view the utility as the measure for the importance and usefulness of a topic: How importantly and usefully a topic is used in later works, how much it influences other works and how wide its applications is in real life.

The formulation of the utility measure cannot be given if we consider on each individual topic. Each topic can be viewed in different level of utility depending on authors' opinions. Constructing the utility measure is to combine all of these information into one quantitative evaluation to yield the general utility. These following features are used:

- $\left\{t_i^k\left(2\right)\right\}_{k=1}^{\Delta}$: $t_i^k\left(2\right)$ is the weight of citations in the k^{th} year to t_i, in which t_i is cited for referring to a theoretical basis, using methods or making comparison. Using citation information is a reasonable way to evaluate the importance and usefulness of a certain topic to other topics. However, not all citations reflects the importance and usefulness, we want to evaluate the importance and usefulness by weighting only "positive" citations, i.e. citations type I, III, and V.

- $\left\{t_i^k\left(4\right)\right\}_{k=1}^{\Delta}$: $t_i^k\left(4\right)$ is the influence of t_i on other topics in the k^{th} year. This can be viewed as a measure for importance of a topic, which reflects the impact of a topic on the research context.

- $\left\{t_i^k\left(5\right)\right\}_{k=1}^{\Delta}$ and $\left\{t_i^k\left(6\right)\right\}_{k=1}^{\Delta}$: As the same as in evaluation of interest measure, the weight of author reputations and sources are used for evaluating the utility by explicit knowledge while the collection of papers was not complete.

5.1.3 Formulation of Interest and Utility Measures

To evaluate the growth in interest and utility of a topic, we consider on all six time-series $\left\{t_i^k\left(j\right)\right\}_k$ $\left(1 \leq j \leq 6\right)$, normalize and evaluate the growth in value of each one. Define $Growth\left(t_i, j\right)$ as the growth in value of the time-series $\left\{t_i^k\left(j\right)\right\}_k$ along the time-axis. The growths in interest and utility are then computed by taking average of the

growths of corresponding features:

$$\text{Measure of growth in interest: } f(t_i) = \frac{1}{4} \sum_{j \in \{1,3,5,6\}} Growth(t_i, j) \tag{5.1}$$

$$\text{Measure of growth in utility: } g(t_i) = \frac{1}{4} \sum_{j \in \{2,4,5,6\}} Growth(t_i, j) \tag{5.2}$$

Another problem now presents itself: How to evaluate the growth in value of a time-series?

One possible solution is to evaluate the speed and acceleration of growth at a specific point. To this end, we first interpolate a time series $s = (s_1, s_2, \ldots, s_\Delta)$ by a continuous, smoothing function

$$\begin{aligned} \varphi &: [1, \Delta] \to \mathbb{R} \\ \text{st} &: \varphi(i) = s_i, (1 \leq i \leq \Delta) \end{aligned} \tag{5.3}$$

and compute the speed and acceleration of the time-series s at the time $x = t$

$$Speed(t) = \frac{d\varphi}{dx}(t) \tag{5.4}$$

$$Acceleration(t) = \frac{d^2\varphi}{dx^2}(t) \tag{5.5}$$

Speed and acceleration are then combined to evaluate the level of growth in interest and utility of each topic at a specific point in time. Based on this evaluation, we can classify topics in different ways according to their interest and utility: a topic growing fast in both interest and utility with high speed and high acceleration can be considered an emerging trend; a topic growing fast in interest but having small utility may be a new attractive research topic, and so on.

However, this is only a local evaluation, meaning that this method lack the evaluation for the tendency of a time-series in the trial period. To make a global evaluation for the tendency of a time-series, we uses inference to predict the dependence of the value on time: By considering each pair $(time, value)$ as a data point, we use regres-

sion analysis to predict the dependence of values on the time. The simplest way is to apply linear regression on all data points and use the slope co-efficient of the regression equation to evaluate the global tendency of the time-series.

5.1.4 Classification of Emerging Trends

Based on the interest and utility measures, we classify topics into the following groups:

1. **Emerging Trends:** A topic growing in interest in utility over time is an emerging trend. In our model, a topic t_i is an emerging trend if and only if $f(t_i) > 0$ and $g(t_i) > 0$

2. **Potentially Emerging Trends:** A topic, that is growing fast in interest but has small value in utility (i.e. $f(t_i) >> 0$ and $g(t_i) \leq 0$), is considered as a candidate for further emerging trends (we called it an potentially emerging trend). For example, a topic is recently attractive, but too novel to have citations and influence.

3. **Creative Trends:** A topic, that is growing fast in utility even though it has no growth in interest (i.e. $f(t_i) \leq 0$ and $g(t_i) >> 0$), is called a creative trends. This topic might not be novel, but it is important and useful for other works. By applying the theories and techniques it provided to other research area, people might create emerging topics or wide-spread applications.

4. **Obsolete Trends:** A topic, that is not growing in both interest and utility, is obsolete (i.e $f(t_i) \leq 0$ and $g(t_i) \leq 0$). For example, a research issue that was completely solved or a method that was superseded by other advanced methods.

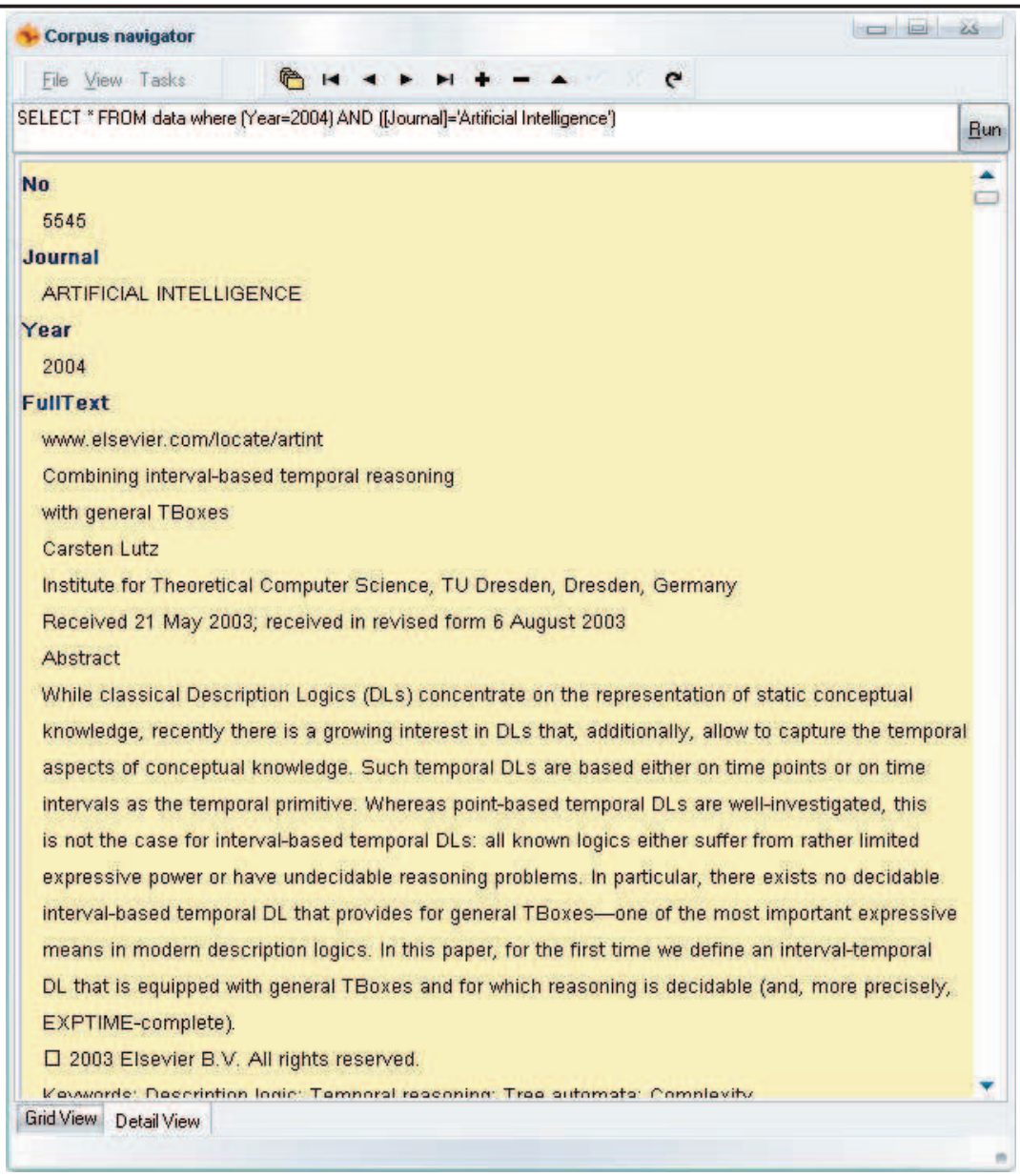

Figure 5.1: User-interface of the prototype system

5.2 Implementing a Prototype System

5.2.1 The Scientific Corpus

We collected 15,000 papers related to the domain Artificial Intelligence from 1998 to 2005. All of them are originally in Portable Document Format (.PDF) published in 20 journals:

1. Advanced Engineering Informatics
2. Artificial Intelligence
3. Artificial Intelligence in Engineering
4. Artificial Intelligence in Medicine
5. Cognitive Science
6. Cognitive Systems Research
7. Data and Knowledge Engineering
8. Electronic Commerce Research and Applications
9. Engineering Applications of Artificial Intelligence
10. Expert Systems with Applications
11. Fuzzy Sets and Systems
12. Information Sciences
13. Information Systems
14. International Journal of Approximate Reasoning
15. International Journal of Electrical Power and Energy Systems
16. International Journal of Human-Computer Studies
17. Knowledge-Based Systems
18. Neural Networks
19. Neurocomputing
20. Robotics and Autonomous Systems

5.2.2 Pre-Processing

Because the original papers is .PDF files, we implement the Acrobat Application Programming Interface (Acrobat API) into the system to extract contents of papers. The text data are then indexed and citation links between papers are identified. Figure 5.1 shows the user-interface of the system and an example of a paper.

5.2.3 Experiments

We have designed an experiment to test the model, in which we we choose the period from 1998 to 2005 as the trial period. The system will examine all documents in the trial period to extract features and evaluate trends.
The following topics are examined:

- Machine Learning
  - Kernel Methods
  - Conditional Random Fields
  - Neural Networks
  - Decision Tree
- Text Mining
  - Information Extraction
  - Information Retrieval
- Reasoning
  - Planing
  - Scheduling
  - Decision Making
  - Belief Revision

- Knowledge Representation
  - Fuzzy Logic
  - Fuzzy Modeling
- Rule-based Systems
  - Expert Systems
- Multi-agent Systems
- Natural Language Processing
  - Computational Linguistics
  - Natural Language Understanding
- Speech Processing
  - Speech Synthesis
- Computer Games
- Genetic Algorithms

The level of growths in interest and utility are computed by the prototype system as shown in Table 5.1. Look at this table, we can see how the prototype system classify topics. For examples, "Kernel methods", a promising technique in statistical learning that attracts much interest in the research community, has wide-spread applications, and also strongly impacts on other research topics, is classified into class 1 (an emerging trend). The topic "Conditional Random Fields", a powerful technique to analyze sequential data, has much interest in recent years, but due to its novelty, we could not find any citation to this topic in the database, we can only compute a small value in influence on other topic. Other examples are text processing techniques, such as "Information Extraction" and "Information Retrieval" have quickly grown in both interest and utility because of the explosion of textual data in the WEB.

Topic	Growth in Interest	Growth in Utility	Class
Belief Revision	-0.1361	-0.0172	4
Computational Linguistics	+0.1138	+0.2371	1
Computer Games	-0.1874	-0.0715	4
Conditional Random Fields	+0.3533	+0.0012	1
Decision Making	-0.3228	-0.4689	4
Decision Trees	-0.0500	+0.0042	3
Expert Systems	-0.5096	-0.7401	4
Fuzzy Logic	-0.1187	+0.2001	3
Fuzzy Modeling	-0.0457	+0.0048	3
Genetic Algorithms	+0.0896	+0.0739	1
Information Extraction	+0.0240	+0.1651	1
Information Retrieval	+0.1195	+0.0337	1
Kernel Methods	+0.2801	+0.2588	1
Knowledge Representation	-0.1117	-0.0313	4
Machine Learning	+0.0087	-0.0109	2
Multi-agent Systems	+0.0176	+0.0310	1
Natural Language Understanding	-0.0235	-0.1186	4
Neural Networks	-0.1082	-0.0556	4
Planing	+0.0167	-0.0753	2
Reasoning	-0.3813	-0.2457	4
Rule-based Systems	-0.4508	-0.4734	4
Scheduling	+0.0992	+0.0449	1
Speech Processing	+0.1286	+0.2116	1
Speech Synthesis	+0.1147	-0.1929	2
Text Mining	+0.2555	+0.1467	1

Table 5.1: Evaluating the level of growths in interest and utility

It is difficult to make an comparative evaluation, because most of existing methods in ETD do not make any decision on the output topics [PD95, APL98]. The final decision on emerging trends is often left to users [RGP02, PY01]. The other reason is existing methods do not make clearly the definition of the interest and utility measures, most of them are based on frequencies to visualize the output topics without any classification of emerging trends.

However, we can evaluate if our method can represent topics more reasonable and our topic representation module is more effective in the task of distinguishing emerging and non-emerging trends. To this end, we drop some features that do not exists in other methods and compare with the previous result. Table 5.2 shows the result of the classification without using citation information. In which the method classified "Information Extraction" and "Information Retrieval" into class 4 (non-emerging), while it assigns "Neural Networks" into the class of emerging trends.

In conclusion, our method can classify emerging trends more precisely because it uses a reasonable topic representation method and classifies topics using two separated measures. This also makes the method more flexible when adding some more features extracted from the corpus.

Topic	Growth in Interest	Growth in Utility	Class
Belief Revision	-0.1835	+0.1934	3
Computational Linguistics	+0.0793	+0.4389	1
Computer Games	-0.2287	+0.1315	3
Conditional Random Fields	+0.3240	+0.0024	1
Decision Making	-0.3682	-0.2200	4
Decision Trees	-0.0577	+0.2398	3
Expert Systems	-0.5157	-0.5382	4
Fuzzy Logic	-0.1449	+0.4210	3
Fuzzy Modeling	-0.0612	+0.2128	3
Genetic Algorithms	+0.0764	+0.3138	1
Information Extraction	-0.0178	-0.3697	4
Information Retrieval	-0.0995	-0.2340	4
Kernel Methods	+0.2517	+0.5080	1
Knowledge Representation	-0.1371	+0.1905	3
Machine Learning	-0.0096	+0.2262	3
Multi-agent Systems	-0.0203	+0.2485	3
Natural Language Understanding	-0.0518	+0.1033	3
Neural Networks	+0.1237	+0.1554	1
Planing	+0.0004	+0.1371	1
Reasoning	-0.3867	-0.0308	4
Rule-based Systems	-0.4850	-0.2457	4
Scheduling	+0.0831	+0.2639	1
Speech Processing	+0.0966	+0.4237	1
Speech Synthesis	+0.0745	+0.0459	1
Text Mining	+0.2545	+0.3832	1

Table 5.2: Evaluating the level of growths in interest and utility without citation information

Chapter 6

Conclusions

6.1 Summary and Contributions of the Thesis

Our research objective is to build a model for emerging trend detection in scientific corpora. In other words, the main goal of this research is to overcome the gap of existing ETD models when dealing with an important kind of textual databases: scientific text corpora.

We recognized that the main drawback of existing models lays on their model structures where research topics are not well represented, extracted and evaluated. Therefore, we proposed a more appropriate model that enables us to develop a fully-automatic emerging trend detection method. The key idea is to view each topic as a time-series associated with as many as possible useful features extracted from text and to avoid the use of manual processes as much as possible.

In our model, each topic is represented by a set of temporal features which are commonly provided in scientific papers, this allows our model to adapt to different kinds of scientific corpora and also can be efficiently modified according to the needs of users.

We have developed several methods for extracting features associated with topic. In our experiments, the methods for topic identification and citation type detections achieved impressive results compared to other works. It is worth noting that these

methods do not require user-interactions and their flexibility allows them to be extended.

Finally, the construction of interest and utility measures is a significant contribution of our work. By evaluating the growth in interest and utility separately, we can also classify emerging trends by different criteria as well as clarify the development of research topics in the published literature.

6.2 Future Works

While our methods for topic representation, identification and verification described in Chapter 3, 4, and 5 are interesting, none of them are the last word on the subject. Many extensions, variations, and improvements are possible. It is a rich area for further studies of which we will outline some of immediate extensions that could be performed on each method.

Finding Richer Representation for Topics

Finding more features to represent topic is one possible improvement. For example, our model can represent a topic in the relationships with other topics, but it evaluate each topic individually. However, the developments of related topics may affect the interest and utility of a topic. Representing some features that reflect the development in the whole research context may enable us to detect potential emerging trends. That could be very interest and useful for researchers.

Tracing Development along Citation Links and Citation Types

The work presented in this thesis uses citation types for weighting only. However, if we trace backward following citation links, citation types can also help us to draw the development of a topic from original ideas to recent development with improvements, modifications or simplifications. In context of emerging trend detection research, a method to trace backward in time along citation link and use citation type to analyze

the development of a topic is very useful and could be improved to be a new stand-alone emerging trend detection method.

Improve the Interest and Utility Measures

Almost existing ETD methods leave the final decision of emerging trends to users. Our method has built these two measures in an attempt at developing an automatic topic verification method. However, these measures should be verified and evaluated in order to identify emerging trends more precisely and reasonably.

Use of Web Resources

Some components of our prototype system is under construction. The original idea for this prototype system is to evaluate the model with full-text data. Since the Web information proliferation provides huge dynamically changing textual data online freely. Detecting emerging research trends from World Wide Web has an opportunity to be "emerged" in context of emerging trend detection and textual data mining.

Publications

1. Minh-Hoang Le, Tu-Bao Ho, Yoshiteru Nakamori. A method of detecting emerging trends in a large repository of scientific documents. In *Proceedings of the 5th Symposium on Knowledge and System Science*, pp.243-248, 2004.

2. Minh-Hoang Le, Tu-Bao Ho, Yoshiteru Nakamori. Detecting Emerging Trends from Scientific Corpora. In *Proceedings of 69th Japanese Society for Artificial Intelligence knowledge based system workshop*, pp.45-50, Awazi, Japan, 2005.

3. Minh-Hoang Le, Tu-Bao Ho, Yoshiteru Nakamori. Detecting Citation Types using Finite-State Machines. In *Proceedings of 10th Pacific-Asia Conference on Knowledge Discovery and Data Mining, PAKDD*, 2006 (to appear).

4. Minh-Hoang Le, Tu-Bao Ho, Yoshiteru Nakamori. Detecting Emerging Trends from Scientific Corpora. In *International Journal of Knowledge and Systems Sciences*, 2006 (to appear).

5. Minh-Hoang Le, Tu-Bao Ho, Yoshiteru Nakamori. A Model for Detecting Emerging Trends from a Large Collection of Scientific Papers. Submitted to the *International Journal of Data and Knowledge Engineering*, 2006.

Bibliography

[ABC+95] J. Allan, L. Ballesteros, J. Callan, W. Croft, and Z. Lu. Recent experiments with inquery. In *4th Text Retrieval Conference (TREC-4)*, 1995.

[APL98] James Allan, Ron Papka, and Victor Lavrenko. On-line new event detection and tracking. In *Research and Development in Information Retrieval*, pages 37–45, 1998.

[BP00] Fabien Bouskila and William M. Pottenger. The role of semantic locality in hierarchical distributed dynamic indexing. In *Proceedings of the International Conference on Artificial Intelligence*, 2000.

[Bri92] Eric Brill. A simple rule-based part-of-speech tagger. In *Proceedings of ANLP-92, 3rd Conference on Applied Natural Language Processing*, pages 152–155, 1992.

[BY83] Gillian Brown and George Yule. *Discourse Analysis*. Cambridge University Press, Cambridge, UK, 1983.

[CC99] Chaomei Chen and Les Carr. A semantic-centric approach to information visualization. In *International Conference on Information Visualization (IV'99)*, pages 18–23, 1999.

[CL92] H. Chen and K. J. Lynch. Automatic construction of networks of concepts characterizing document databases. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(5):885–902, 1992.

[COM] COMPENDEX. COMPENDEX®. Available from World Wide Web: http://www.uspto.gov/main/patents.htm.

[DDL+90] Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.

[DeJ82] G. DeJong. An overview of the frump system. In W. G. Lehnert and M. H. Ringle, editors, *Strategies for Natural Language Processing*, pages 149–176. Erlbaum, 1982.

[DHJ+98] George S. Davidson, Bruce Hendrickson, David K. Johnson, Charles E. Meyers, and Brian N. Wylie. Knowledge mining with vxinsight: Discovery through interaction. *Journal of Intelligent Information Systems*, 11(3):259–285, 1998.

[DR72] J.N. Darroch and D. Ratcliff. Generalized iterative scaling for log-linear models. *The Annals of Mathematical Statistics*, pages 1470–1480, 1972.

[FD95] Ronen Feldman and Ido Dagan. Knowledge discovery in textual databases (KDT). In *Knowledge Discovery and Data Mining*, pages 112–117, 1995.

[FSM+95] David Fisher, Stephen Soderland, Joseph McCarthy, Fangfang Feng, and Wendy Lehnert. Description of the umass system as used for muc. In *Proceedings of the 6th Message Understanding Conference (MUC-6)*, pages 127–140, 1995.

[Gev02] David R. Gevry. Detection of emerging trends: Automation of domain expert practices, 2002.

[Hea94] Marti A. Hearst. *Context and Structure in Automated Full-text Information Access*. PhD thesis, University of California at Berkeley, USA, 1994.

[HHWN02] Susan Havre, Elizabeth Hetzler, Paul Whitney, and Lucy Nowell. Themeriver: Visualizing thematic changes in large document collections. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):9–20, 2002.

[INS] INSPEC. INSPEC®. Available from World Wide Web: http://www.iee.org.uk/Publish/INSPEC.

[Jon88] Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Document retrieval systems*, pages 132–142, 1988.

[KdRH+01] Ronald N. Kostoff, J. Antonio del Rio, James A. Humenik, Esther Ofilia Garcia, and Ana Maria Ramirez. Citation mining: integrating text mining and bibliometrics for research user profiling. *Journal of the American Society for Information Science and Technology*, 52(13):1148–1156, 2001.

[KGP+03] April Kontostathis, Leon Galitsky, William M. Pottenger, Soma Roy, and Daniel J. Phelps. A survey of emerging trend detection in textual data mining. In Michael Berry, editor, *A Comprehensive Survey of Text Mining*, chapter 9. Springer-Verlag, 2003.

[LAS97] Brian Lent, Rakesh Agrawal, and Ramakrishnan Srikant. Discovering trends in text databases. In David Heckerman, Heikki Mannila, Daryl Pregibon, and Ramasamy Uthurusamy, editors, *Proceedings of 3rd International Conference on Knowledge Discovery and Data Mining, KDD*, pages 227–230. AAAI Press, 1997.

[LDC] LDC. Linguistic data consortium. Available from World Wide Web: http://www.ldc.upenn.edu.

[Leh82] W. G. Lehnert. Plot units: A narrative summarization strategy. In W. G. Lehnert and M. H. Ringle, editors, *Strategies for Natural Language Processing*, pages 375–414. Erlbaum, 1982.

[Ley02] L. Leydesdorff. Indicators of structural change in the dynamics of science: Entropy statistics of the sci journal citation reports. *Scientometrics*, 53(1):131–159, 2002.

[LGB99] Steve Lawrence, C. Lee Giles, and Kurt Bollacker. Digital libraries and autonomous citation indexing. *IEEE Computer*, 32(6):67–71, 1999.

[LM92] Elizabeth D. Liddy and Sung-Hyon Myaeng. Dr-link's linguistic-conceptual approach to document detection. In *TREC*, pages 113–130, 1992.

[LSL+00] V. Lavrenko, M. Schmill, D. Lawrie, P. Ogilvie, D. Jensen, and J. Allan. Mining of concurrent text and time-series. In *ACM KDD Text Mining Workshop*, 2000.

[Luh57] H.P Luhn. A statistical approach to mechanized encoding and searching of literary information. *IBM Journal*, pages 309–317, 1957.

[Luh58] H.P Luhn. A statistical approach to mechanized encoding and searching of literary information. *IBM Journal*, pages 159–165, 1958.

[MFP00] Andrew McCallum, Dayne Freitag, and Fernando Pereira. Maximum entropy Markov models for information extraction and segmentation. In *Proceedings of the 17th International Conference on Machine Learning*, pages 591–598, 2000.

[NFH+96] Lucy T. Nowell, Robert K. France, Deborah Hix, Lenwood S. Heath, and Edward A. Fox. Visualizing search results: Some alternatives to query-document similarity. In *SIGIR*, pages 67–75, 1996.

[NKO00] Hidetsugu Nanba, Noriko Kando, and Manabu Okumura. Classification of research papers using citation links and citation types: Towards automatic review article generation. In *Proceedings of the American Society for Information Science (ASIS)*, pages 117–134, 2000.

[PD95] A.L. Porter and M.J. Detampel. Technology opportunities analysis. *Technological Forecasting and Social Change*, 49:237–255, 1995.

[PFL+00] Alexandrin Popescul, Gary Flake, Steve Lawrence, Lyle Ungar, and C. Lee Giles. Clustering and identifying temporal trends in document databases. In *Advances in Digital Libraries, ADL 2000*, pages 173–182, Washington, DC, 2000.

[PH03] Son Bao Pham and Achim G. Hoffmann. A new approach for scientific citation classification using cue phrases. In *Australian Conference on Artificial Intelligence*, pages 759–771, 2003.

[PMS+98] Catherine Plaisant, Richard Mushlin, Aaron Snyder, Jia Li, Dan Heller, and Ben Shneiderman. Lifelines: Using visualization to enhance navigation and analysis of patient records. In *Proceedings of the American Medical Informatic Association Annual Fall Symposium*, pages 76–80, 1998.

[PY01] William M. Pottenger and Ting-Hao Yang. Detecting emerging concepts in textual data mining. *Computational information retrieval*, pages 89–105, 2001.

[Rab89] Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, volume 77:2, pages 257–286. IEEE, 1989.

[RGP02] Soma Roy, David Gevry, and William M. Pottenger. Methodologies for trend detection in textual data mining. In *Proceedings of the Textmine '02 Workshop, Second SIAM International Conference on Data Mining*, 2002.

[RL94] Ellen Riloff and Wendy Lehnert. Information extraction as a basis for high-precision text classification. *ACM Transactions on Information Systems*, 12(3):296–333, 1994.

[RT01] Kanagasabai Rajaraman and Ah-Hwee Tan. Topic detection, tracking and trend analysis using self-organizing neural networks. In *Proceedings of the Fifth Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'01)*, 2001.

[SA96] Ramakrishnan Srikant and Rakesh Agrawal. Mining sequential patterns: Generalizations and performance improvements. In Peter M. G. Apers, Mokrane Bouzeghoub, and Georges Gardarin, editors, *Proceedings of 5th International Conference on Extending Database Technology, EDBT*, volume 1057, pages 3–17. Springer-Verlag, 1996.

[SA00] Russell Swan and James Allan. Automatic generation of overview timelines. In *SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 49–56, New York, NY, USA, 2000. ACM Press.

[SC73] Gerard Salton and C.S.Yang. On the specification of term values in automatic indexing. *Journal of documentation*, 29:351–372, 1973.

[Sit] US Patent Site. US patent site. Available from World Wide Web: http://edina.ac.uk/compendex.

[Sma73] H. Small. Co-citation in the scientific literature: A new measure of the relationship between two documents. *Journal of the American Society of Information Science*, 24:265–269, 1973.

[Tar72] Robert E. Tarjan. Depth first search and linear graph algorithms. *SIAM Journal of computing*, 1:146–160, 1972.

[Teu99] Simone Teufel. *Argumentative Zoning: Information Extraction from Scientific Text*. PhD thesis, University of Edinburgh, 1999.

[Vit67] Andrew. J. Viterbi. Error Bounds for Convolutional Codes and an Asymptotically Optimal Decoding Algorithm. *IEEE Transactions on Information Theory*, pages 260–269, 1967.

[Wei71] Melvin Weinstock. Citation indexes. *Encyclopedia of Library and Information Science*, 5:16–41, 1971.

[Wor] WordNet. A lexical database for the english language. Available from World Wide Web: http://wordnet.princeton.edu.