

Detection of Emerging Trends: Automation of Domain Expert Practices

David R. Gevry
CSE Department Technical Report
Lehigh University
LU-CSE-02-001

Table of Contents

Abstract.....	1
1.0 Introduction.....	2
2.0 Motivation and Related Work	4
3.0 Approach.....	7
3.1 Citation-based Emerging Trend Detection	8
3.1.1 Tracing a Trend via Citation Linkages	9
3.2 Web-based Emerging Trend Detection.....	20
3.2.1 Identification of emerging trends using web resources.....	21
3.2.2 Case Study on the topic of Object Databases	25
3.3 Combination of Methodologies for Emerging Trend Detection.....	29
3.3.1 Verification Step for Combined Methodology.....	30
3.4 Partial Automation of Combined Methodology.....	32
4.0 Implementation	33
4.1 Database Implementation.....	33
4.1.1 Database Schema Development.....	34
4.1.2 Database Implementation of the Schema	41
4.2 Script Design	47
4.2.1 Automatic Extraction and Aggregation of Terms	48
4.2.2 Term Extraction Interface	49
4.2.3 Automatic Aggregation of Verification Metrics	50
4.2.4 Verification Interface	55
5.0 Experiments and Analysis of Results	56

5.1 Methodology of Evaluation.....	57
5.1.1 System Performance	57
5.1.2 Validation Methodology for Database Correctness.....	58
5.1.3 Evaluation of the Combined Methodology.....	60
5.2 Results	61
5.2.1 Results of Performance Study.....	61
5.2.2 Usability Study.....	62
5.2.3 Results of Combined Methodology Evaluation.....	64
5.2.4 Discussion of Results	66
6.0 Conclusions and Future Work	66
References.....	69
Appendix I	75
Appendix II.....	78
Appendix III.....	89

List of Figures

Figure 1: Inductive Decision Tree Document Frequency	15
Figure 2: Inductive Decision Tree Repeated Authors	16
Figure 3 Inductive Decision Tree New Venues	17
Figure 4: Fuzzy Decision Tree Document Frequency.....	18
Figure 5: Fuzzy Decision Tree Repeat Authors	19
Figure 6: Fuzzy Decision Tree New Venue Inclusion	20
Figure 7: Web Mining Algorithm.....	24
Figure 8: Term Extraction Interface	50
Figure 9: Document Count Table	52
Figure 10:Unique Author Table	53
Figure 11: Unique Co-Author Table	54
Figure 12:Unique Venue Table	54
Figure 13:Document Title Table	55

List of Tables

Table 1: Helper Terms	22
Table 2: INSPEC Search: Object Databases.	26
Table 3: INSPEC database search:	29
Table 4: Initial Database Schema	34
Table 5: Dublin Core Schema.....	35
Table 6: Resource Descriptive Formats for Four Online Sources	38
Table 7: Final Schema for Database Storage	40
Table 8: Record Table.....	41
Table 9: Author Table.....	42
Table 10: Reference Table.....	43
Table 11: Conference Table.....	43
Table 12: Full Text Table	44
Table 13: ACM Digital Library Table.....	44
Table 14: US Patent Table	45
Table 15: IEEE Explore Table.....	45
Table 16: INSPEC Table	45
Table 17: Treatment Table.....	46
Table 18: Performance Statistics	61
Table 19: Metrics of Usability Evaluation	62
Table 20: Statistics of Usability Evaluation	62
Table 21: Answer Ranking List.....	63
Table 22: Metrics to Question Associations.....	63
Table 23: Analysis of Group A and Group B precision results	64

Table 24: Lower Tail Test Results 65

Abstract

The automatic detection of emerging trends is an important research area in the field of textual data mining. Explosive performance increases in technology require the development of precise forecasting tools to drill down into the textual artifacts of various research communities and reveal emerging innovations to technology planners and investors. This report presents two manual methodologies that have been developed in the study of approaches to the task of emerging trend detection. These methods are then integrated together in order to improve the overall precision of each. The performance of this combined methodology is evaluated using the standard metric of precision and it is shown with a confidence of 95% that the usage of this methodology improves precision for the detection of emerging trends.

The overall goal of this research is the automation of various domain expert trend detection practices and integration of these automated modules into a fully automated system. The first two methods of this automation are presented and their usability and performance are evaluated. We show that these tools aid in increasing the efficiency of the task of emerging trend detection and further propose improvement for these tools as well as future plans for the full automation of the presented methodology.

1.0 Introduction

Emerging trend detection is an exciting area of research in text mining. An emerging trend is a topic area for which one can trace the growth of interest and utility over time. An example of such a trend is XML, a technology that emerged in the mid 1990's.

The necessity for automated methods for detecting emerging trends has grown with the increasing availability of digital information. What makes it difficult is that it is not only based on data collected / explored but also on the experience or domain expertise of the person involved in the detection process. Currently too much data is available for a human expert to examine manually and not risk missing some vital piece of information. Trending of this nature is thus primarily based on human-expert analysis of sources (e.g., patent, trade, and technical literature) combined with bibliometric and text mining techniques that employ both semi [1] and fully automatic methods [3, 6, 11].

With the continued increases in the performance of computational technologies, more aggressive implementations of trend detection methodologies are becoming possible. This has spurred research into the development of more sophisticated methodologies for the detection of emerging trends. Such emerging trends are defined as topic areas that have grown in size and variety at an increasing rate over time. Specifically, incipient emerging trends (trends that occur for the first time) are the main focus of this work.

This report describes two methodologies that are being developed for the detection of emerging trends and then demonstrates how ideas from both methodologies have been integrated to form a new approach. The first methodology (Section 3.1) uses citation information to form a set of documents related to a candidate emerging trend. Thresholds are then applied to determine whether the candidate emerging trend, represented by the set of documents, has a sufficient foothold in the research community to be considered an emerging trend. The second methodology (Section 3.2) uses a web-based approach to gather candidate emerging trends in a main topic area (an area that has progressed from an emerging trend to a recognized area of research or study). Candidate emerging trends are then verified through an abstract database search of recent years. Finally a combined approach is described (Section 3.3) where key pieces of the threshold approach for verification from the first methodology are integrated with the web-based research algorithm of the second methodology. The focus of this report is on the partial automation of this combined approach, its usability, and performance in a multimedia learning environment. This semi-automated approach will be an integral part of the CIMEL (Collaborative Constructive Inquiry-based Multimedia E-Learning) project [21] for promoting inquiry-based learning.

Chapter 2 provides an overview of similar work currently being done in the field of data mining.

Chapter 3 discusses the two initial methodologies and their combination, as well as the proposed areas of automation.

Chapter 4 discusses the implementation of aspects of the automation process and the incorporation of two automatic methods into the CIMEL multimedia environment. As part of the process of automating the trend detection, the design of a database schema for a literature repository is discussed in Section 4.1. Next the implementation of the tools that are used to automatically generate statistical information for the web search algorithm and verification stage of the combined methodology is presented (Section 4.2). Finally the user interfaces for these tools are presented and the reasoning behind their design is explained.

Chapter 5 provides a description of the methodology for evaluation for this implementation as well as the overall evaluation. The performance of the tools for automation is presented followed by a description of the CIMEL system's validation methodology. Next the usability of the system is tested by a group of students who were given the task of finding incipient emerging trends in the main topic area of "inheritance and object-oriented programming." This group of students was split into a control group and a test group to test the increase in precision of the combined methodology for emerging trend detection. The control group viewed a CIMEL multimedia module on inheritance, while the test group viewed the same module as well as a second CIMEL multimedia tutorial explaining the combined methodology (Section 3.3) for detecting emerging trends.

Chapter 6 presents conclusions as well as the project's future plans.

2.0 Motivation and Related Work

In previous work, [3, 9, 10], we examined the usage of various linguistic and statistical features to track trends across time. The HDDITM system [4,14] 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 rate of change in the size of clusters and in the frequency and association of features is used as input to machine learning techniques to classify topics as emerging or non-emerging.

However, a domain expert does not use linguistic features exclusively to detect an emerging trend. The research in this report is motivated by the desire to better characterize a domain expert approach to the detection of emerging trends. Through this research we aim to identify features and methods to enhance the automatic detection of emerging trends.

Several research projects are exploring solutions to the detection of emerging trends. ThemeRiver [5] enables users to visualize trends and detect emerging trends. It is a prototype (mock-up) that visualizes thematic variations over time across a collection of documents. As it flows through time, the river changes width to depict changes in the thematic strength of temporally collocated documents. The river is within the context of a timeline and a corresponding textual representation of external events.

Another project [6] presents a method of tracking sequential patterns across time. This method extracts content bearing words from the corpus it is using and generates sequential patterns within a selected time-interval based on a minimal support threshold between content bearing words. The authors also present a system for visualizing these patterns.

The Envision system [7] allows users to explore trends in digital library metadata (including publication dates) graphically 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 TDT project [8] is an ‘Event Tracking’ mechanism, which tracks topical information in a stream consisting of news stories using speech processing technology. The goal of [8] is essentially to detect changes in topics – disruptive events exhibiting discontinuities in semantics in localized data sources such as newscasts. Our research [3, 4, 9, 10] focuses on integrative or non-disruptive emergence of topics that build on previously existing topics. There is a significant difference in the goal of these research projects: unlike the TDT research, our goal is to detect novel trends that are globally incipient in a given domain.

TimeMines [11] is an automated system that generates overview timelines for topics in free text news corpora. These timelines are used to indicate the key topics involved in the corpora and their coverage with a ranking function of how important a particular topic is within that area. In contrast, our research goal is not to identify all topics that are important but rather identify selected emerging trends that are incipient.

TOAS [1] extracts information about particular emerging technologies through a process of search and retrieval from abstract databases (e.g., INSPEC, Medline, etc.) with manually generated queries. Following this initial data collection, various data processing techniques are

used to generate reports on the topic of the search. TOAS incorporates the ideas of ‘Monitoring’ and ‘Bibliometrics’ in a complementary fashion for the detection of emerging trends. Monitoring involves tracking of data for a specific purpose, the implication of which will subsequently be interpreted by a domain expert. On the other hand, bibliometrics uses counts of citations in publications, patents or citations to measure and interpret scientific and technological advances [2]. This is the first step towards a fully automatic approach to emerging trend detection.

In [19] a discussion of studies of patterns in citations concludes that active research fronts develop in citations between recent years. This is an important characteristic that can be leveraged to enhance our fully automatic approach to emerging trend detection. Additionally [20] discusses a method for soft-clustering documents using citation patterns in a database (CiteSeer: www.csindex.com). The method centers on the assumption that scientific disciplines form around key papers in scientific literature. Using this assumption, a clustering algorithm is developed to track the changes in scientific disciplines using the most highly cited papers in each year to form discipline clusters. This approach is similar to our citation-based methodology, however the focus appears to be at a lower level of granularity as it looks for broader trends/disciplines. In our research we are looking for the emergence of new trends or topics rather than disciplines. Additionally, in our method, the selection of documents into a set representing a trend is based on heuristics that determine documents’ similarity and community relationship to the trend. Overall the vision of this work is to develop a precise method for the automatic detection of emerging trends.

3.0 Approach

This the material presented in Section 3.0 of this report represents the joint work between Soma Roy, another student in our lab and the author of this report. Contents of this section can be found verbatim in Chapter 3.0 of Soma Roy's thesis [23] due to the co-authoring of [27].

The problem of trend detection is approached from two different perspectives. The first methodology uses citations as well as author relationships to generate a document/trend set for a selected topic. This set is then analyzed to detect the initial emergence point of a trend. The second methodology uses web resources to identify candidate emerging trends. Domain knowledge is used to validate potential incipient trends as emerging. Finally a combined methodology improves on the second methodology by integrating features of the first methodology.

The initial development of the presented methodologies represents the joint effort of the emerging trend project staff for the CIMEL project [21] at Lehigh University (www.lehigh.edu/~cime1). The goal of this work is to explore different methods to increase the precision of existing software for the detection of emerging trends through the examination of manual methods using domain knowledge to perform this task. This report focuses on the automation of key steps in the combined methodology and lays the ground work for fulfillment of the ultimate goal of a fully automated emerging trend detection system.

3.1 Citation-based Emerging Trend Detection

Citation linkages can be used to trace the development of a trend across time. Through this tracing a community of authors can be assembled for a given trend. This methodology uses citation linkages to assemble a document set for a chosen trend and verifies it as an emerging trend based on its foothold in the research community.

3.1.1 Tracing a Trend via Citation Linkages

In what follows outline the seven steps in this first methodology.

1. Determination of a potential trend and/or selection of a topic of interest.

A topic of interest is selected and is characterized in one or two descriptive sentences. This description is used for comparison to the documents retrieved in later steps. Various sources are used to retrieve recent documents on the topic (e.g., Citeseer: www.csindex.com). This allows initial validation of a topic's worth based on document and citation counts. Additionally, key authors on a topic can be identified based on counts of citations to their work.

The documents retrieved from this step are examined to verify that they discuss the topic of interest and are then used to determine keywords related to the topic.

2. Initial Citation Traversal Backward in Time

The references of the retrieved papers from various sources are examined, and from these references a subset is selected based on the titles and the author names that appear more frequently in the papers. In this case any repeated reference or repeated author is considered significant due to the limited amount of citation information available. This differs from step

four which selects the papers based on the number of papers that cite a particular author or reference. Once the papers are selected, citation link information is used to retrieve the abstract of this set of papers from various online resources (e.g., Science Citation Index). The abstracts are then examined for relation to the topic, and those papers that do not discuss the topic are pruned. This comparison is accomplished by examining the title of the document and the abstract for relation to the description of the topic formed in step one. One method that may enhance this process is finding the subject sentence in the abstract. This sentence usually starts with particular catch phrases (e.g., “This paper”, “We present”, “The author discusses”, etc.) This helps determine whether the citation should be used for examination of the trend. If an abstract is not available from online sources then the reference is used conditionally to trace citations forward in time.

3. Tracing Citations Forward in Time

Citations to the papers retrieved from the initial traversal are looked up (e.g., with Science Citation Index). This search returns a large number of documents, which requires initial pruning steps. First it is assumed that pruning based on venue will yield a viable set of documents that is representative of the topic. Thus venue pruning rejects all documents published in venues outside of a selected domain. This pruning action helps in reducing the total number of documents retrieved and also restricts the documents retrieved to a particular area of interest.

The next pruning step examines the title and keywords of the documents for similarity to the topic description sentence formed in step 1. Finally, the abstracts are examined as in step 2 to

determine whether to include or exclude a given document from the trend set (the set of documents related to the topic description sentence which represent the trend). These last two steps can be combined if there is a small enough collection of documents that cited a particular source. If the documents from step 2 that were used conditionally to trace citations forward in time do not yield useful documents, they are pruned from the trend set.

4. Tracing Backwards in Time

The references for the papers obtained in steps 2 and 3 are examined and a subset of these references is formed. Each set of papers is handled separately, with the set from step 3 being examined first. First the author names are examined for the set. Author names that are referenced by three or more documents are selected. Next the repeated references for articles written by one of the selected authors are obtained. New papers that were not previously found are selected. If the abstracts or titles are obtainable, this set is pruned based on similarity as in step 2. If there is not an abstract available for a paper it is conditionally added. This process is repeated for the set of papers from step 2.

5. Set Improvement

Online repositories with citation linkage information (e.g., Science Citation Index) are queried with terms from the topic description to determine if there are additional documents missed by the citation tracing. The results are pruned on similarity to the topic and duplicates are removed. If there are remaining papers, these are added to the trend set and their references are examined to identify potential matches with the set obtained so far. The citation information of the retrieved documents is combined with that of the trend set. The process ends with a final query

to additional online sources (e.g., INSPEC, Compendex and www.csindex.com) with terms from the topic description sentence. This final search retrieves documents that are not covered by the sources that contained the necessary citation information for the previous steps.

6. Identification of Emergence Time

Upon completion of the previous step duplicate documents are identified and removed from the trend set. The document frequency, number of repeated authors, and number of new venues is then graphed by year. We then select the years with an overall higher document frequency. It is our premise that these ‘candidate years’ have a higher likelihood of being points where the trend is emerging.

7. Thresholds for Emerging Trend Detection

Using candidate years as an emergence point for the trend, we then apply a series of thresholds. These thresholds represent heuristics derived from case studies of emerging trends that we have conducted. Two such case studies are considered in the following Section 3.1.2.

1. A document frequency of five or greater is required for the candidate year. This is used to prune out candidate years where a trend has not developed to the point of emergence.
2. The candidate year is required to be the largest document year in all years prior to the candidate. The candidate year should represent the largest amount of work to date on a trend. Therefore we prune out candidate years that do not exceed the years prior to them in document frequency.

3. The candidate year is required to contain 20% of all documents in the trend set, prior to and including the candidate. The candidate year should have a high level of representation for the work to date on a topic.
4. The candidate year is required to contain 10% of all documents in the trend set for all the years studied. If the candidate year being examined is not the current year (in present time) then this threshold is used to assert the overall importance of the candidate year.
5. 25% of all documents in prior years must occur in the three years prior to the candidate year. The trend should have an increase over a short period of time to be considered emerging. Additionally, the majority of documents in a trend that is emerging should occur close to the emergence point.
6. Venue variety increases in the candidate year. This increase in venue variety indicates an increase in the activity of a trend.
7. There should be at least one repeated author present in the trend. The trend needs to have the beginnings of a community of authors.
8. There should be at least 10 venues present in the trend. A core set of venues is required for the trend to be considered emerging.

3.1.2 Case Studies

The citation linkage methodology is used to trace the following trends and verify them as emerging or non-emerging trends. These trends were selected from the field of data mining for their impact and usage in the field.

Selection of Decision Trees:

The main topic of our case studies is decision trees. From this topic two sub-topics are selected; Inductive Decision Trees and Fuzzy Decision Trees. The topic of decision trees is selected due to the attention it has received in data mining and machine learning literature and research [e.g.12, 15, 16, 17, 18].

Inductive Decision Tree Case Study:

Our first case study considers the trend of Inductive Decision Trees in the domain of Data Mining. Since the trend of Inductive Decision Trees has emerged already this gives us a good starting point to examine the patterns surrounding its emergence. The methodology is followed to yield a trend set of documents. Then thresholds are examined to determine when the initial point of emergence occurs. Figure 1 shows the document frequency for the Inductive Decision Tree trend set across time.

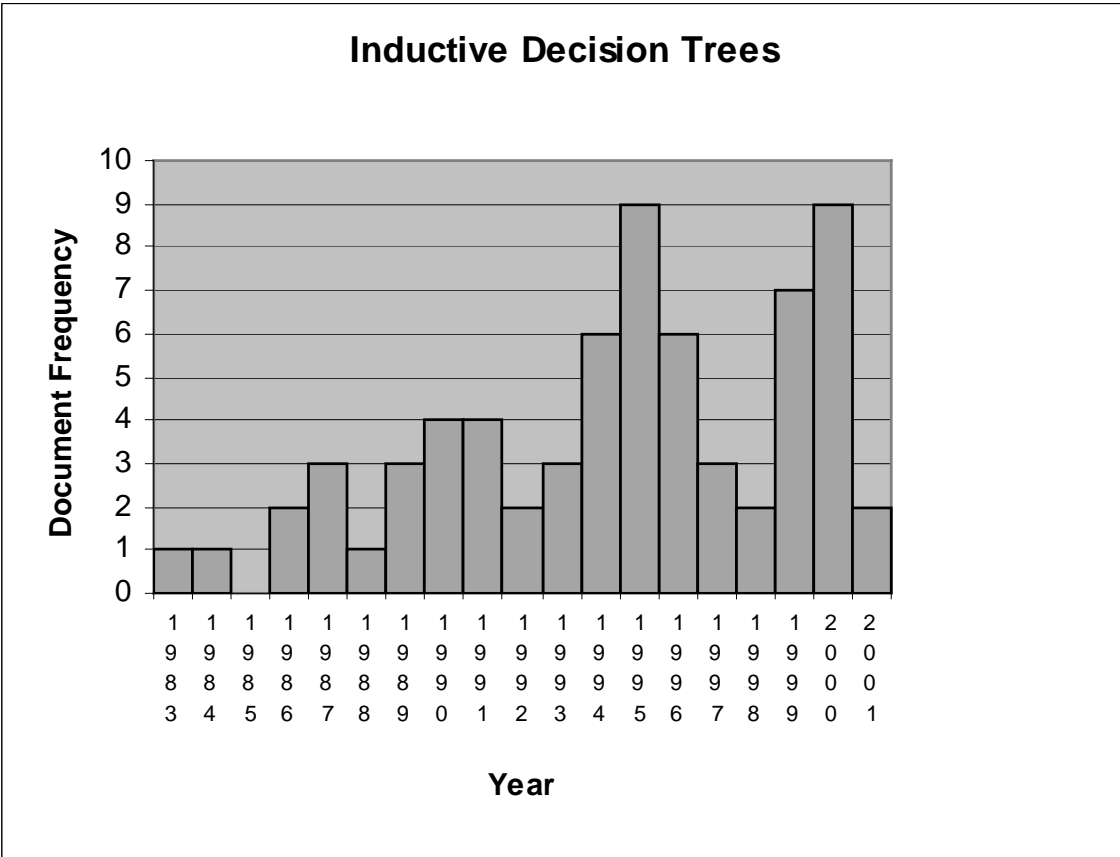


Figure 1: Inductive Decision Tree Document Frequency

From this graph the years 1991, 1995, and 2000 are selected as potential candidate years of emergence due to their document frequency in respect to previous years. However, the year 1991 is excluded due to not meeting the document frequency threshold of at least 5 documents. Next, the year 2000 is removed from the candidate year set because it does not have the required 20% of previous documents. These thresholds are used to maintain a level of representation in the candidate year. The rationale behind this is to prevent a candidate year from being identified as emerging when the bulk of the documents occur in prior years. Similarly the threshold for the past three years (threshold 5 in step 6 in Section 3.1) is used to maintain a larger percentage of the documents close to the candidate year.

The next threshold we apply required that all candidate years contain at least one repeated author. In order for a trend to be considered emerging it has to have the beginnings of an author base or community. Figure 2 shows the growth of the number of repeated authors over time.

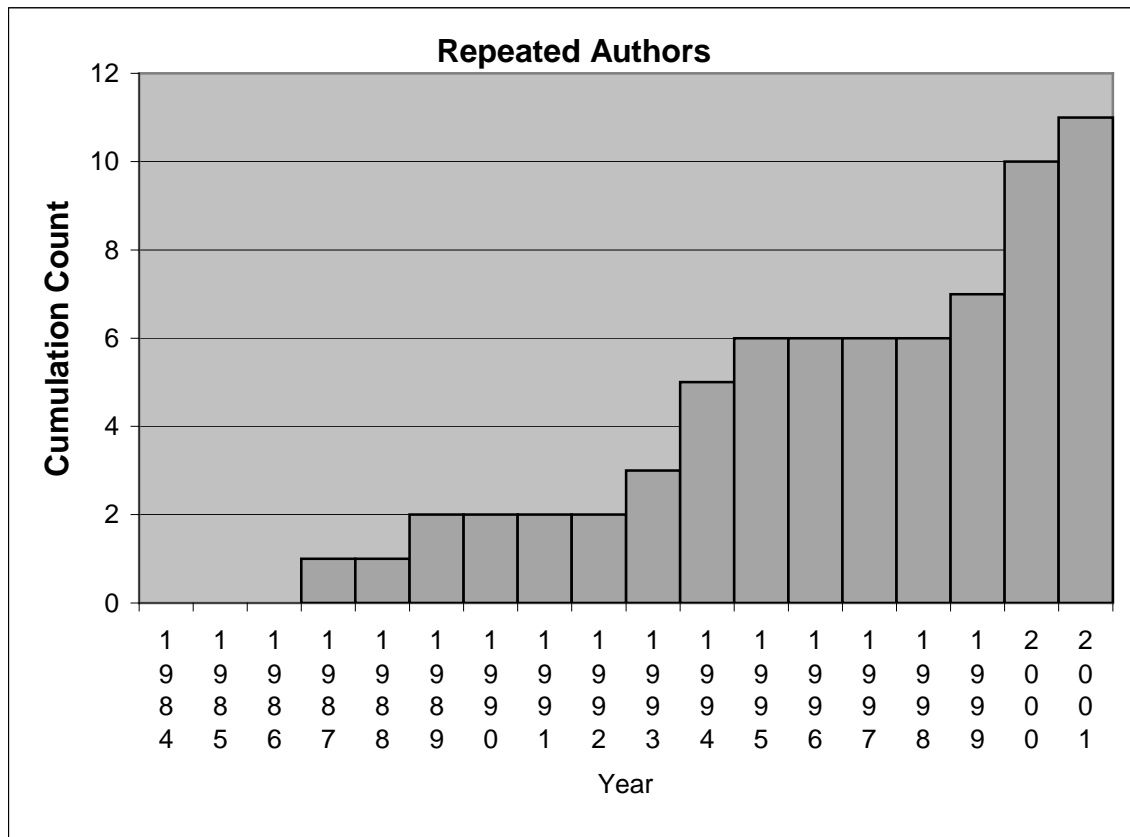


Figure 2: Inductive Decision Tree Repeated Authors

Finally the restriction of an increase in the number of new venues is applied to determine how active the trend is in the domain. There should be a core-venues-set that the topic occurs in to guarantee that the topic has a foothold in a domain before it is considered emerging. Additionally, there should be a relative growth within the domain for the topic. The number of

new venues a topic acquires in the trend set represents this growth. Figure 3 shows the inclusion of new venues into the trend set of induction decision trees in relation to time.

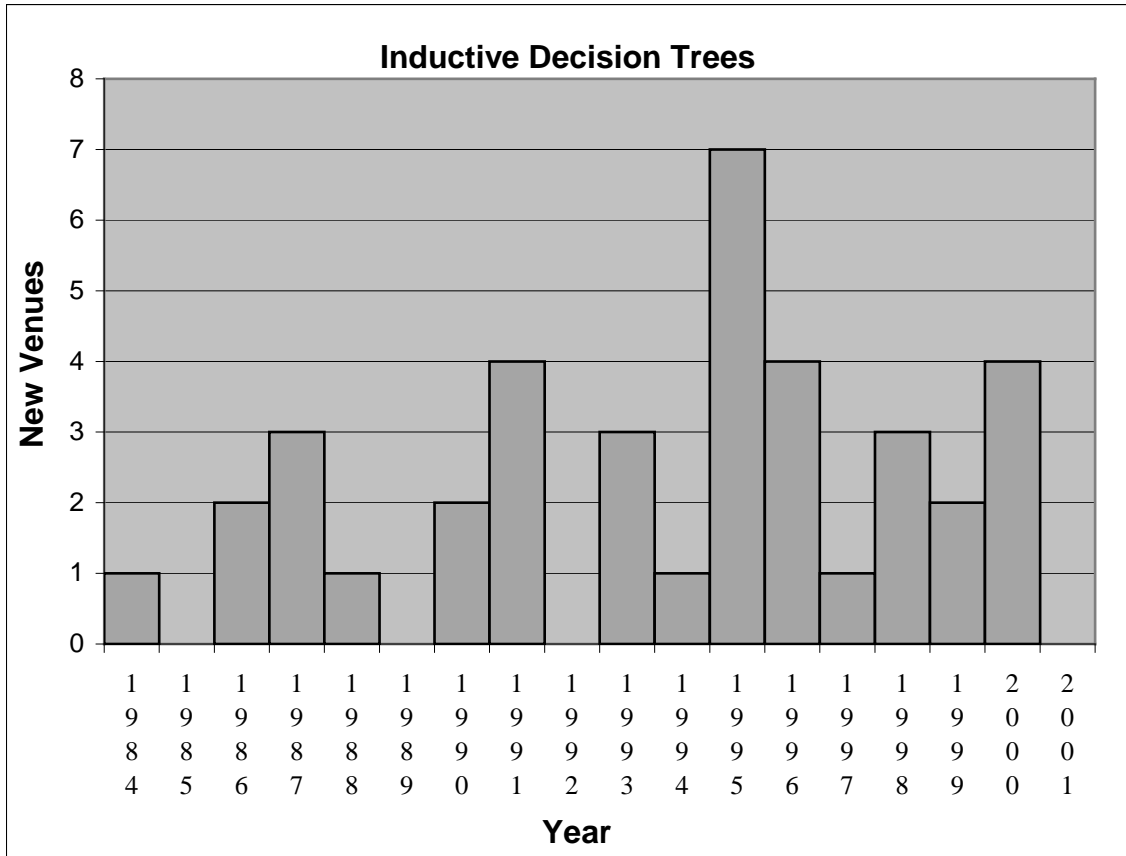


Figure 3 Inductive Decision Tree New Venues

The candidate year 1995 is selected as the year of emergence of the inductive decision tree trend by these threshold metrics.

Fuzzy Decision Tree Case Study:

Following the same methodology, the document set for the topic of fuzzy decision trees is generated. Figure 4 shows the document frequency for the trend.

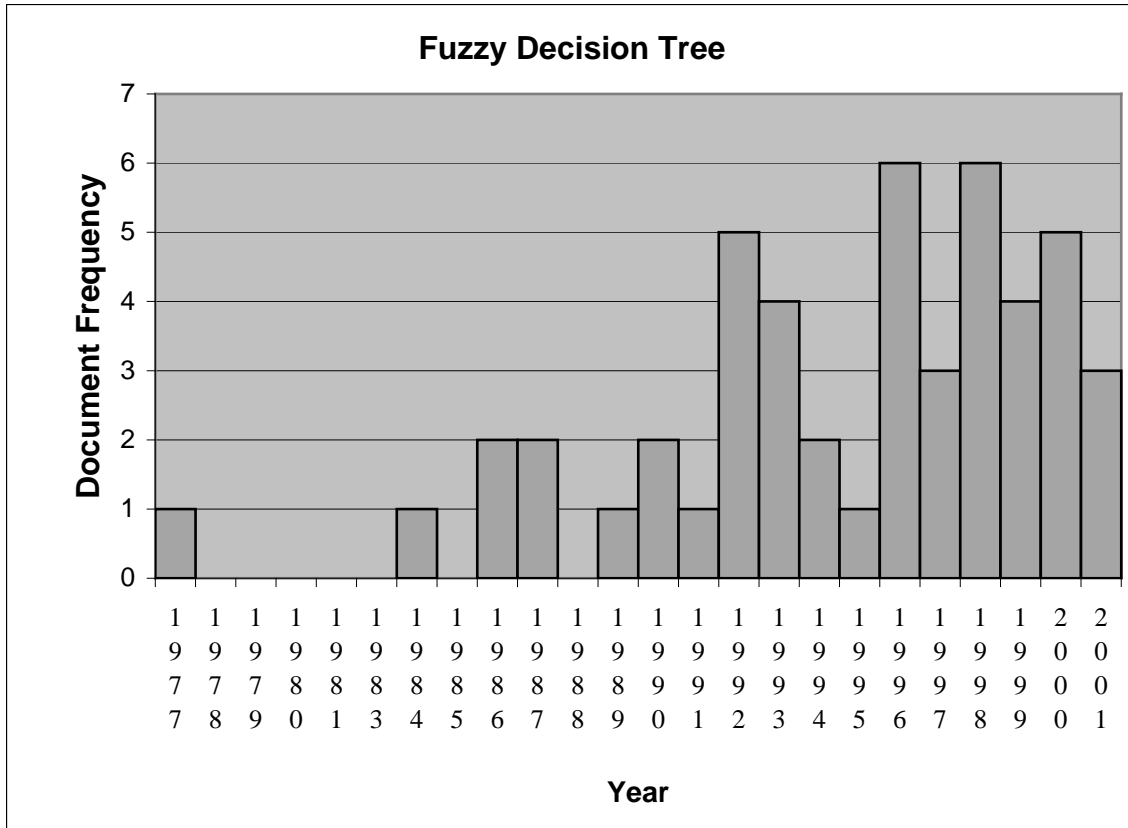


Figure 4: Fuzzy Decision Tree Document Frequency

The candidate years for this topic are 1992, 1996, 1998, and 2000. The years 1998 and 2000 are pruned by threshold (2) because they are not the largest years for document frequency.

All candidate years contain at least one repeated author showing that the trend is beginning to receive attention from a group of authors. Figure 5 shows the increase of repeated authors with relation to time.

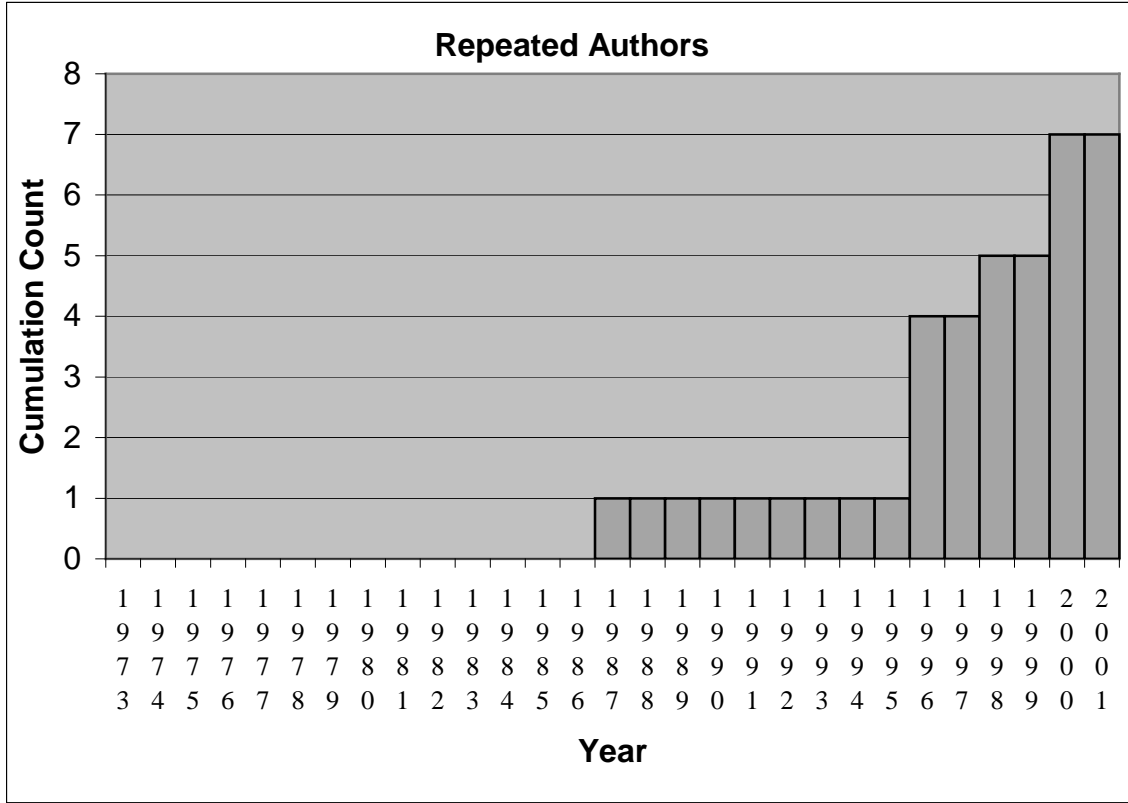


Figure 5: Fuzzy Decision Tree Repeat Authors

The year 1992 is removed from the set of candidate years because it does not reach the threshold for venues included in the trend. There are only seven different venues present by 1992. The year 1996 does, however, contain the required number of venues and has an increase in the number of new venues over the prior year. Therefore the year 1996 is selected as the year of emergence for the trend fuzzy decision trees.

Our goal is to develop manual methods for detecting emerging trends with higher precision than our previous work [3, 4, 9, 10, 14], and then to automate these methods. Currently we have not been able to formally evaluate this first methodology due to difficulty in obtaining abstracts with

citation information. However, as seen in [20] (discussed in Section 2) the usage of citation information to track trends in a repository of research documents is feasible. As part of our future work we plan to obtain access to a repository containing citation information and use this data to aid in further development and validation of citation based techniques.

In what follows we present a second manual methodology for the trend detection, and conclude with a combined approach that incorporates elements of both methodologies. This combined approach is evaluated formally in Section 5.

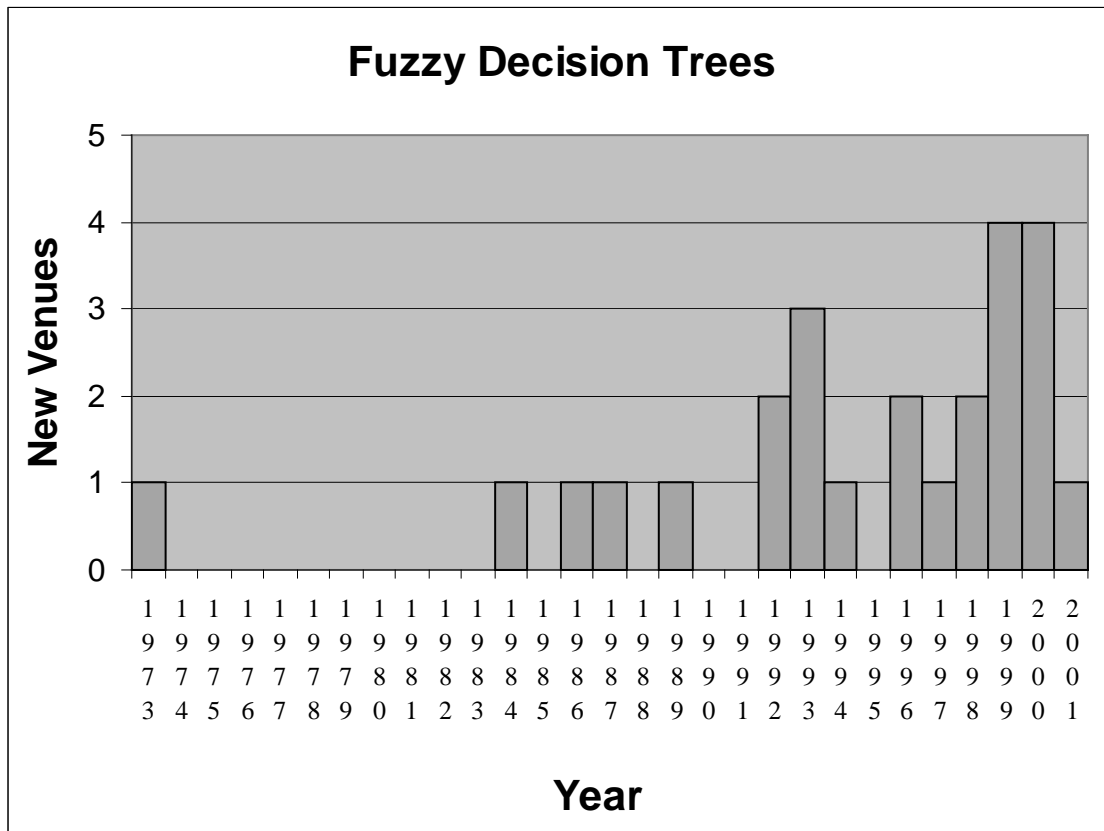


Figure 6: Fuzzy Decision Tree New Venue Inclusion

3.2 Web-based Emerging Trend Detection

The following methodology uses web-based research to detect candidate emerging trends and then verifies these trends by examining their representation in a research abstract database. This method represents the work of Soma Roy, a member of our project group, and can be found verbatim in Chapter 3.0 of [23]. This method focuses particularly on incipient emerging trends, trends that are in the beginning stages of emergence.

3.2.1 Identification of emerging trends using web resources

Following is the second blueprint for a methodological approach to the detection of emerging trends. We present a case study of this approach in section 3.2.2.

1. Selection and validation of main topic area.

Detection of emerging trends starts with the selection of a main topic area. Knowledge in this area is required as the use of domain knowledge at various stages of identification of emerging trends is necessary. The objective is to discover emerging trends in the area of interest.

An INSPEC Database search on the chosen main topic is done to confirm it as a possible area of research. INSPEC is a well known scientific abstract database which houses many research abstracts in the areas of Computer Science, Electrical Engineering, as well as Physics. We choose it because of its wide area of coverage in the Computer Science research literature community.

2. Search for candidate emerging trends.

Recent conferences and workshops are searched for discussion on the main topic area giving special attention to workshop websites and technical papers for possible emerging trends (i.e., topics within the domain of the main topic area).

3. Candidate emerging trend verification.

A web search engine (e.g., Google, Yahoo, etc.) is used to find additional trends and uncover further evidence of references to the candidate trends.

Following is a list of words associated with emerging trends (also called “helper” terms):

most recent contribution	recent research	a new paradigm	hot topics
Emergent	newest entry	cutting edge strategies	first public review
Novel	new approach	proposed	current issues
Future	recent trend	Next generation	

Table 1: Helper Terms

Two possible scenarios are:

a) If step 2 identified any candidate emerging trends, a search is made using any of the popular web search engines like Yahoo or Google using a candidate trend <and> any of the helper terms from the above list (where <and> indicates a logical AND of search terms).

Otherwise,

b) If step 2 did not identify any candidate emerging trends, a search is made using any of the popular web search engines like Yahoo or Google using main topic area <and> any of the helper terms from above list.

The algorithm in Figure 7 is followed at this stage. In the algorithm, “main topic” should be read as “candidate emerging trend” if case 3(a) applies. In completing this step several candidate emerging trends are identified. Also, further references to candidate emerging trends identified in step 2 can be found.

Input: Search Engine retrieved pages
(Search term: (Candidate Emerging Trend <OR> Main Topic) <AND> helper term)

Output: List of additional Candidate Emerging Trends

```
algorithmicWebMining() {
    Make an empty list L2; // will be used to store candidate emerging trends
    Define Topic = Candidate Emerging Trend <OR> Main Topic;
    Define m = frequency of Topic in page;
    Define n = frequency of helper term;
    m = 0;
    n = 0;
    Click on link = 1;

    While (pages to inspect) { // search engine retrieved pages
        If (year of page == recent) { // with in 2 to 3 years of current year (e.g., 2002)
            Count m, n; // Count the number of occurrences of the Topics m and Helper Term n in the page
            If (m > 1) {
                If (n > 1)
                    Accept the Page;
                    Process_page();
                Else
                    n = frequency of any other helper term
                    If (n > 1)
                        Accept the Page;
                        Process_page();
                    Else
                        n = 1; // Remember the search engine retrieved pages found at least one
                        occurrence of the helper term in this page.
                        If helper term co-occur with Topic
                            Accept the Page;
                            Process_page();
                        else
                            Reject Page;
            }
        } // end of m check
    } // end of year check
    Link ++; // click on next link
} // end of while
} // end of algorithmicWebMining

Process_page()
{
    Add Topic to L2;
    List frequency of all words / phrases in the page;
    Phrases or words with higher frequency (ignore general words / phrases) are given higher weights;
    Phrases / Words co-occurring with helper terms are given higher weights;
    Add phrases / words with higher weights to L2;
}
```

Figure 7: Web Mining Algorithm

4. Verification of Algorithmic Results

An INSPEC Database search is performed using “main topic area <and> newly found candidate emerging trend” from the year of origin of the main topic area to the current year. If the frequency of documents referencing the search terms increases over the years, the candidate emerging trend is confirmed as a bona fide trend with respect to the main topic.¹

If few documents appear in different years (say one or two), the authors of the articles are also investigated. If the same author is writing (may be as a follow up thesis to recent research, etc.), it’s NOT an emerging trend.

5. Additional Trends

Steps 3 and 4 are repeated with combinations of other helper terms and/or other candidate emerging trends until all the desired emerging trends are found.

Alternative approach

If an emerging trend has already been predicted based on some previous research or using domain expertise, step 2 of the methodology can be skipped. Thus after step 1, step 3 would be followed using the “predicted trend <and> helper term(s)”.

3.2.2 Case Study on the topic of Object Databases

¹ Note: One objective here is to find the year of origin of a candidate emerging trend within the main topic.

In this section a common example of how the above methodology would be used is presented. For this example the main topic area is chosen to be “Object Databases”.

1. Selection and validation of main topic area.

First following step 1 of the methodology, a main topic area is chosen which in this case is Object Databases.

Following this selection, an INSPEC database search is performed to verify the selected topic area for its potential to contain emerging trends.

1988	1989	1990	1991	1992	1993	1994
10	13	28	38	24	41	73
1995	1996	1997	1998	1999	2000	2001
36	46	47	54	39	36	16

Table 2: INSPEC Search: Object Databases.

The above table shows the document counts by year from an INSPEC query on Object Databases. The coverage of this topic over recent years suggests that new innovative enhancements to Object Databases are being researched. Therefore Object Databases is a valid main topic area for the identification of emerging trends.

2. Search for candidate emerging trends.

The next step is to explore workshops and conferences that are related to the discussion of this main topic area. This is initiated by examining the OOPSLA website (oopsla.acm.org). Through the examination of OOPSLA’s 2001 conference as well as additional sources, “XML Databases”

is found to be a candidate emerging trend. Following are a few excerpts from key conference and workshop papers, that were used in reaching this conclusion. Phrases in the relevant portions of the excerpts are highlighted (bold) to provide a visual cue to the information used in the detection of XML Databases as a candidate emerging trend.

i.) (oopsla.acm.org/oopsla2001/fp/workshops/17.html)

“During the past few years, there has been a **considerable interest and growth in a number of new and emerging technologies, such as XML**. For many organizations already using object-orientation with database management systems, XML data adds a new dimension that brings considerable flexibility and promise, ... The **recent trend towards XML servers, native XML databases** and support for XML in existing relational databases is a testimony to the importance of this issue for the vendor community as well.”

ii) EDBT 2002 Workshop (XMLDM)

“... As database systems increasingly start talking to each other over the Web, there is a **fast growing interest in using the eXtensible Markup Language (XML) as the standard exchange format**. As a result, many relational database systems can export data as XML documents and import data from XML documents. XML is on its way to becoming the communication standard of the Web. Moreover, there is an **increasing trend to store XML-data in database systems** and, by this, make it easier to access and maintain.”

iii) 1st ECOOP Workshop (XOT)

“XML has many similarities with object-oriented data models and languages. However, whereas the object-oriented technology has reached a great level of maturity, **XML is still in its infancy.**”

Additional Sources that were examined include:

iv) Web Databases 2001

v) WebDB Workshops

vi) ACM SIGIR 2000 Workshop on XML and Information Retrieval

The way in which XML Databases is referred to in the above excerpts identifies it as a candidate emerging trend in the area of Object Databases.

3. Candidate emerging trend verification.

A web search engine (e.g. Google, Yahoo, etc.) is used to find additional trends and to find further evidence of the candidate trends that are found in the previous step (in this case only XML Databases). A query is formed which combines “Object Databases” <and> various helper terms (where <and> indicates a logical AND of search terms), which are listed in the methodology. The methodology does not require step 3(b) to be followed due to the detection of a candidate emerging trend in step 2 of the methodology, however it can be followed to identify additional candidate trends and to assist in the validation of candidates found in the previous steps.

The algorithm in Figure 7 is followed and the candidate emerging trend (XML Databases) is found as an emerging trend that is widely referred to in recent research work.

4. Verification of Algorithmic Results

An INSPEC Database search using the “Object-oriented <and> XML Database” is performed to verify the algorithmic results.

1988	1989	1990	1991	1992	1993	1994
0	0	0	0	0	0	0
1995	1996	1997	1998	1999	2000	2001
0	0	0	0	5	11	5

Table 3: INSPEC database search:

Object-oriented <and> XML Database²

The frequency of documents referencing the search term increases over the years; hence we conclude that XML Databases is an emerging trend with reference to Object Databases in Object Oriented Software Engineering.

3.3 Combination of Methodologies for Emerging Trend Detection

In this section the two methodologies presented for detecting emerging trends are integrated to improve the precision over the individual methodologies. The main area of the methodologies that is combined is the verification portion. The usage of authors and venues in verification is added to the second methodology to strengthen its ability to capture a developing research

² Note: The frequency of documents in 2001 (i.e., current year) is not complete.

community. Using the incipient or web-based emerging trend detection methodology we form the initial portion of the combined methodology.

3.3.1 Verification Step for Combined Methodology

The verification step occurs after candidate emerging trends are identified, by way of steps 1-3 (see incipient emerging trend methodology (Section 3.2) steps 1-3). Similar to both original methodologies, a repository search is performed to assist in the verification of the candidate emerging trend. For the experiments performed in this report we retrieve a subset of the INSPEC repository that focuses on “inheritance” and “object oriented programming.” This repository and its generation is discussed later in the implementation section (Section 4.0). The benefit of the generation and local storage of this repository is two-fold. The first benefit is that the documents contained in the repository are all closely related to the topic of “inheritance” which is the main topic area for both experiments. Secondly it allows for the efficient automation of data mining solutions to improve the usage of the methodology.

The document frequency of the trend is examined first to determine the likelihood that it is an emerging trend. Document frequency, number of documents per year and overall for a trend, plays an important role at identifying an incipient emerging trend in both minimum and maximum levels. A trend needs to have a minimal representation in the area of “inheritance” to be considered an incipient emerging trend. Therefore it must have a presence in the repository that is represented by the metric of document frequency. The current threshold that we heuristically choose for verification is a minimum bound on document frequency of greater than one document. If there is only one document present we cannot consider the candidate to be

incipiently emerging because there is no indication of its increasing representation in the research community.

We are still researching the upper bound for document frequency, however our initial assumption is that this bound is found within the range of 10-20 documents. Our premise is that after a certain level of acceptance is reached the trend emerges. Therefore there is a point where a trend can no longer be considered an incipient emerging trend yet can still be considered to be emerging. Further research into the distribution of document frequency will yield a better indication as to the range of this upper bound.

An additional requirement for the candidate emerging trend is that the documents include the current year as well as additional prior years. In order to be considered an emerging trend there needs to be a distinct presence in the current and recent years as well as in the web community. This is to ensure that the trend has been around long enough to receive the attention of the research community, as well as to yield an indication of where the trend is heading in the research community.

In addition to these metrics the user is provided with specific document information on the items retrieved for the trend. This information includes the unique author set for the documents, unique venues, unique co-author sets, the titles by year, and the abstract for each document. This information provides the user with wide range of additional information from which he/she can gauge a trend's validity as an incipient emerging trend. For example, the user can use this information to determine if there is an acceptable author base to consider the trend to be

emerging. If the document distribution is sufficient per the above criteria, then it is selected as an incipient emerging trend. However, the user can use their domain knowledge and the additional author and venue information to mark this as a non-emerging trend if the user feels the document base is invalid. Examples of this would be a particular group of authors who seem to be affiliated and a single venue in which they publish with only two to three documents in the set. This would indicate that the topic was only being examined within a single institution and may not have the attention needed to be considered an incipient emerging trend. Usage of the author and venue information stems from the venue and author thresholds in the citation-based verification process discussed in Sections 3.1.1 and 3.1.2.

3.4 Partial Automation of Combined Methodology

The focus of this work is the partial automation of the combined methodology presented above. This automation targets two key areas of the methodology that require excessive data aggregation for their processing. While the ultimate goal of the work in emerging trend detection is the full automation and refinement of this methodology, for this set of experiments we implement two of the most time consuming parts of the methodology. The first part that is implemented is the term extraction part of the web-based algorithm. To facilitate this term extraction we provide a tool to the users of this methodology to automatically extract the frequency of occurrence of the candidate emerging trend as well as the helper terms. The second part of the methodology that is implemented is the aggregation of data for the verification step described in the last section.

The tool that is used for the first implementation allows the user to enter a URL and a candidate emerging trend. The user is returned the frequency of occurrence of the candidate emerging trend and each of the helper terms. This gives the user a direct indication of the usefulness of the provided URL and increases the efficiency of the web search algorithm. Currently, for the purpose of this work, this tool is implemented to support normal text/html pages as well as PDF files.

The other part of the methodology that we implement is the verification step. The user is provided with a query interface to search our repository for a candidate emerging trend. Sections 3.3 and 4.1 describe this repository and how it is generated from the INSPEC abstract database. The documents that are selected from this database are in the area of “inheritance” and these documents and their related information were inserted into a relational database. The user can query this database through the interface that we provide and they are returned a set of tables that contain the aggregated information on the candidate emerging trend as well as links to the individual related abstracts. This provides the user with an efficient display of the metrics that are involved in verifying the candidate emerging trend.

Both of these tools and their interfaces are discussed in more detail in Section 4.

4.0 Implementation

4.1 Database Implementation

In the following section we discuss the implementation of the literature repository. First the database schema development for the database repository is presented. Next the implementation of this schema for the literature repository is discussed.

4.1.1 Database Schema Development

The first and crucial part of the implementation of the verification step is the development of the database schema for the repository. This schema is designed for not only the CIMEL repository but also to be a lasting database schema for later data mining research projects. Therefore in developing this schema close attention was paid towards its later flexibility for the addition of new document sources. The development of a schema for the storage of data in the CIMEL repository starts with an initial set of fields that are required for the immediate data needs of the system. This set is listed below:

ID	numeric
Authors	Text
Editors	Text
Venue	Text
Publication Date	Date
Type of Venue	Char
Title	Text
Abstract	Text
Reference Information	Text

Table 4: Initial Database Schema

However, to expand the later usability and flexibility of this data schema a variety of other data schemas are examined to determine other important fields that are not present in the original

schema. The goal is to allow a wide variety of sources to be added using this data schema as well as increase later data mining benefits of the repository itself.

The first schema for resource description that was examined is the Dublin Core. Additional information can be found on this at dublincore.org. The Dublin Core is comprised of a set of 15 elements:

Content	Intellectual Property	Instantiation
Coverage	Contributor	Date
Description	Creator	Format
Type	Publisher	Identifier
Relation	Rights	Language
Source		
Subject		
Title		

Table 5: Dublin Core Schema

The initial schema that we developed mirrors some of these elements quite closely. Of these elements we identify Title, Source, Creator, Contributor, Date, Identifier, Type, and Description as items that the current schema encompasses (Title, Venue, Authors, Editors, Publication Date, Reference Information, Type of Venue, and Abstract respectively). The Resource Description Framework (RDF)³ used by the University of Illinois in the Grainger Digital Library (dli.grainger.uiuc.edu) employs a selection of Dublin Core. In comparison the RDF format used by the Grainger Digital Library uses 12 of the 15 elements of the Dublin Core (Coverage,

³ Resource Description Framework (RDF) is an XML ontology for the representing semantics describing a resource in a standardized machine-readable format. (www.w3.org/RDF/)

Contributor, and Subject are not included in this format). Coverage is a difficult element to integrate into a data schema. The description of coverage given at dublincore.org is shown below:

Element Description: The extent or scope of the content of the resource. Coverage will typically include spatial location (a place name or geographic co-ordinates), temporal period (a period label, date, or date range) or jurisdiction (such as a named administrative entity). Recommended best practice is to select a value from a controlled vocabulary (for example, the Thesaurus of Geographic Names [[Getty Thesaurus of Geographic Names, http://shiva.pub.getty.edu/tgn_browser/](http://shiva.pub.getty.edu/tgn_browser/)]) and that, where appropriate, named places or time periods be used in preference to numeric identifiers such as sets of co-ordinates or date ranges.

Currently none of the available sources include coverage or a field similar to this element. One possible field that could relate to coverage is the country of publication. However this does not describe the extent or scope of the publication so there is only a weak relationship between the two. Nonetheless country of publication as well as language of publication should be added to the schema for their ability to group different sets of documents in the repository. This may prove useful for future data mining projects.

The RDF format of the Grainger Digital Library includes a field for the element Relation. This is implemented in the form of a list of the document's references. Currently none of the data we have obtained contains reference information; however we will likely obtain information of this

nature in the near future. Therefore the inclusion of reference information is an important addition to the data schema. Additionally the element Relation in the RDF format contains links to the full text version of the document. This is also important to include in the data schema, because we plan to obtain the full text of documents when possible.

The final two elements of the Dublin Core that the RDF format of the Grainger Digital Library implements which our initial data schema did not include are Format and Publisher. Format can be implemented to describe the status of the full text link. This will allow the determination of validity of the link and the document format that is in use. The publisher should be included to maintain the information retrieved with documents that include a publisher in their fields.

Finally two important additions that were added to the schema are a source identification field and a unique storage identification field. The source identification allows backtracking to the initial repository from which the document was retrieved. The unique storage id allows backtracking to the archived version of the document as well as serving as the primary key for many of the database tables.

Presented below are the resource description formats of INSPEC (axiom.iop.org), the US Patent Office (www.uspto.gov), the Delphion patent database (www.delphion.com), and the ACM Digital Library (portal.acm.org). These are a few of the sources that we use in data mining tasks.

INSPEC:	ACM Digital Library:	US Patent Office:	Delphion:
Data source:			United States Patent
Record type:			
INSPEC Abstract number:			
Accession number:			
Title:		Title	Title:
Author(s):	Authors	Inventors:	Inventor(s):
		Primary/Assistant Examiners:	Primary/Assistant Examiners:
		Assistant Examiner:	
Author affiliation:		Assignee:	Applicant/Assignee:
Journal title:			
Conference Title:			
Volume and Issue Number:			
Volume:			
Issue:			
Inclusive page numbers:	Pages:		
Start Page:			
Publication date:		Filed:	Issued/Filed Dates:
Publication year:	Year of Publication:		
CODEN:			
ISSN:	ISSN:		
SICI:			
Country of publication:			
Place of publication:			
Material identity number:	DOI:	Appl. No.:	Application Number:
Publisher:	Publisher		
Conference date:			
Conference location:			
Number of references:			
Language:			
Treatment:			
Abstract:	ABSTRACT	Abstract	Abstract:
		Claims	First Claim:
Controlled indexing:	Subject Descriptors:	Field of Search:	Field of Search:
Uncontrolled indexing:	General Terms:		
Classification codes:	Primary Classification:	Current U.S. Class:	Current Class
		Intern'l Class:	IPC Class:
			Original Class
			Priority Number(s):
			Other Abstract Info:
	References	Other References	Other References:
	Citings		
	Review		
	Keywords:		
	Additional Classification:		

Table 6: Resource Descriptive Formats for Four Online Sources

Using these fields as a template combined with our initial schema and the elements that are found in the Dublin Core and University of Illinois Grainger Digital Library formats, a more comprehensive and flexible data schema was formed. Certain document-type-specific fields were generalized to reduce the total number of fields. Primary Authors and Additional Authors/Investigators generalized the US Patent data down to two fields. Source-specific-identification information can be stored in a delimited string format to reduce the number of fields required. This allows future data mining without sacrificing the flexibility or generality of the schema. Additionally, these strings can be later expanded into additional tables through the processing of these delimited strings as is shown with the tables presented in the next section. Another set of fields that can be generalized are the classification code fields. Future research of classification codes from various sources will help in conversion of these codes into a general classification schema for the repository.

The research of these various resource description schemas yielded the following schema. This schema is subdivided into logical tables to represent the various aspects of a document in the next section.

Schema	Data Type
Data source	varChar
Data Source Information	Text
Record/Venue type	varChar
Title	varChar
Primary Creator(s)	varChar
Additional Creator(s)	varChar
Creator affiliation	varChar
Venue title	varChar
Volume	varChar
Issue	varChar
Inclusive page numbers	varChar
Publication date	Datetime
Publication year	Datetime
CODEN	varChar
ISSN	varChar
SICI	varChar
Country of publication	varChar
Place of publication	varChar
Material Identification	varChar
Publisher	varChar
Conference date	Datetime
Conference location	varChar
Number of references	int
Language	varChar
Treatment	varChar
Abstract	Text
Claims	Text
Subject Descriptors: Controlled indexing	Text
General Terms: Uncontrolled indexing	Text
Classification/Field of Search	Text
References	Text
Citings	Text
Other Abstract Info	Text
REVIEW	Text
Full Text Link	Text
Format	varChar
Unique Storage ID	varChar
Reference To Document	varChar

Table 7: Final Schema for Database Storage

4.1.2 Database Implementation of the Schema

The database schema aims at flexibly holding a description for a document from any data source. To facilitate this we created the Record Table to hold only the fields that we would expect a large number of sources to contain. The Record Table therefore contains a single entry for all documents stored in the repository. The Record Table is responsible for providing answers to questions relating to where the document came from and the contents of the document itself.

Record Table		
Variable	Data Type	Description
Unique Storage ID	varChar	Identifies the Record in the DB
Data Source	varChar	Source of the Record
Record/ Venue Type	varChar	Restricted by a code table, describes the type of record
Title	varChar	The title of the Record
Venue Title	varChar	The title of the Venue in which the Record occurs
Volume	varChar	The Volume in the Venue
Issue	varChar	The Issue in the Venue
Inclusive page numbers	varChar	The pages in the Venue in which the Record occurs
Publication date	Datetime	The date of publication
PublicationDateString	varChar	A string representation of the publication date or date range
Country of Publication	varChar	Restricted by a code table.
Place of Publication	varChar	
Material Identification	varChar	A bibliographic reference to the record
Publisher	varChar	The publisher of the Record
Number of references	int	
Language	varChar	Restricted by a code table
Abstract	text	
Claims	text	For US patent
Other Abstract Info	text	For US patent
Review	text	If present

Table 8: Record Table

Overall the Record Table is the central part of any item that is stored in the repository. The Unique Storage ID is used to relate all other tables in the system to the corresponding Record Table entry. This ID is created with a source specific identification as well as a prefix identifying the source as well as the format of the item.

The next logical separation in the resource description fields is that of authors. A document or item has a one-to-many relationship with authors. Additionally an author can serve a variety of roles, primary and secondary author, as well as editor. The author table is created to better represent each author's role and characteristics in relation to the corresponding document.

Author Table		
Variable	Data Type	Description
Unique Storage ID	varChar	ID's the record
Author Name	varChar	An author of the record
Role	varChar	the role the author had in the paper restricted by a code table
Affiliation	varChar	Institution or organization of association in any
Homepage	varChar	If available

Table 9: Author Table

The Author Table holds relevant information on the author and associates it with individual items in the repository.

The next table maintains relationships between documents based on citations. Currently we do not have any citation information for the documents in our repository but we plan on obtaining and using this information in our future work.

Reference Table		
Variable	Data Type	Description
Unique Storage ID	varChar	Pointer to the Record
Referenced Storage ID	varChar	ID of the Record if present in the Repository

Table 10: Reference Table

The Reference Table works by linking one Unique Storage ID to another in the repository. If the referenced document is not present, a Record Table entry is generated to represent the referenced document. In the event that the document is later entered into the repository its entry will replace the temporary referencing entry.

The next table that is distinct from the main Record Table is the Conference Table. This table describes the conference a document is published in.

Conference Table		
Variable	Data Type	Description
Unique Storage ID	varChar	
ConferenceDateString	varChar	A string representation of the date range of a conference
ConferenceDate	Datetime	A place holder for later representation of the date of a conference
Conference location	varChar	Location where the conference took place

Table 11: Conference Table

The usage of a string and a date data type to represent the conference date enables representation of the range of time for a conference as well as a comparable date in the system.

The Full Text Table holds a link to a copy of the full document when available. Currently the system contains only abstracts.

Full Text Table		
Variable	Data Type	Description
Unique Storage ID	varChar	
Format	varChar	Format of the Record
Full Text Link	Text	Link to the Record

Table 12: Full Text Table

The following tables represent the individual sources we plan to use in the repository for our data mining research. However, additional sources maybe added later as they become available.

ACM Digital Library Table		
Variable	Data Type	Description
Unique Storage ID	varChar	
Subject Descriptors	varChar	ACM specific classifications
General Terms	varChar	Further generalization may combine
Primary Classification	varChar	Keywords with indexing terms
Keywords	varChar	
Additional Classification	varChar	Source specific classification
DOI	varChar	Source ID
ISSN	varChar	Standardized ID

Table 13: ACM Digital Library Table

US Patent Table		
Variable	Data Type	Description
Unique Storage ID	varChar	
Field of Search	varChar	US patent information
Current US Class	varChar	Source information
International Class	varChar	Source information
IPC Class	varChar	Source information
Original Class	varChar	Source information
Priority Number(s)	int/varChar	Source information

Table 14: US Patent Table

IEEE Explore Table		
Variable	Data Type	Description
Unique Storage ID	varChar	
IEEE Catalog Number	varChar	Record Identification numbers
ISBN	varChar	Source information
INSPECAccessionNumber	varChar	Source information

Table 15: IEEE Explore Table

INSPEC Table		
Variable	Data Type	Description
Unique Storage ID	varChar	
Controlled Indexing	Text	Source expert indexing
Uncontrolled Indexing	Text	Source automatic indexing
Classification Codes	Text	Source information
INSPEC Abstract Number	varChar	Source ID
Accession Number	varChar	Source ID
Material Identification Number	varChar	Source ID
CODEN	varChar	Standardized ID
SICI	varChar	Standardized ID
ISSN	varChar	Standardized ID

Table 16: INSPEC Table

Each of the respective source tables is responsible for the source specific data. The INSPEC source data also is contained in the Treatment Table.

Treatment Table		
Variable	Data Type	Description
Unique Storage ID	varChar	
Treatment	varChar	Restricted by code table

Table 17: Treatment Table

This table holds the individual Treatment field values for each item. This is an INSPEC specific classification of the document type/presentation format. This allows for mining of INSPEC documents based on the values that are contained in the Treatment Field.

The initial data stored in this repository came from the INSPEC database and is restricted to include the terms “inheritance” and “object-oriented.” This restriction is designed to obtain abstracts that are related to the topic area chosen for our experimental evaluation of the methodology. For our experiments with the combined methodology, inheritance in object-oriented programming was selected as the main topic area.

This data was placed into an intermediate XML data schema. Utilizing this data schema and an in-house data manipulation tool the INSPEC data was cleaned. This tool can be found in Appendix III. The data cleaning process of the Inheritance data proceeded with the removal of duplicate documents based on document title followed by various data correction and null value removal operations. To standardize the Country of Publication field all values were examined and abbreviations expanded to their full country names. Full country names are the most

commonly used representation and the logical choice for later data mining. There are various cases where the type of document does not contain a venue title. The number of these cases however is relatively small, and the null value is replaced with the Missing Value identifier (UNKOWN) so that the correct value could be added later when the document retrieval system is refined. After the data was cleaned a Unique Storage ID was assigned to each of the documents.

After the data cleaning and ID assigning phase, the documents were inserted into the database tables using the bulk copy utility for Microsoft SQL Server 7.0 [25]. This utility processes delimited text files that were created by a C++ text management program that extracts the documents from the archived XML data format and forms entries for all valid tables. The functionality of the bulk copy utility could be replicated for another RDBMS by creating a program that uses ODBC or database specific drivers and reads the text files containing each table's information.

The reason we choose to use MS SQL Server 7.0 is due to its Full Text Indexing support. The Full Text Indexing engine is used to search the documents for candidate emerging trends in the verification step. This is described in more detail in the Script Design Section 4.2.

4.2 Script Design

The tools that are used for the automation of the methodology were designed in Perl for a variety of reasons. The first reason is that Perl is an ideal language for the processing of text and string manipulation due to its strong text manipulation capabilities and sophisticated pattern matching

features. Additionally SQL database connection libraries, as well as web service request libraries are readily available. It is also ideal for web-based data presentation through the usage of CGI scripting. The scripts themselves appear in Appendix II.

4.2.1 Automatic Extraction and Aggregation of Terms

The first automation tool that was developed is the term extraction tool. This automated the algorithmic searching of a webpage for the candidate emerging trend frequency as well as the helper term frequency. The input is sent from the CIMEL Flash multimedia environment with a CGI POST. This is handled with the CGI.pm library provided with Perl. Once the URL and candidate emerging trend are received by the server, the URL is checked for the occurrence of an http:// prefix. In the absence of the http:// prefix one is added to provide an absolute path for the URL (this is a requirement of the LWP::UserAgent object). The Perl script also contains the helper terms in a list so that they can be matched as well.

After the input is correctly formatted, the URL is loaded into an LWP::UserAgent object first with the appropriate MIME types that can be accepted by the system. Next an HTTP request is sent to the web server containing the URL by way of an HTTP request object that processes the LWP::UserAgent object and returns the resulting page or file. The page headers are extracted from the result object and the type of the request is determined. Currently, the two accepted types are text/html and application/PDF. If the requested page is a PDF file then the contents of the page are assigned to a string and written to a temporary file using the FILE:TEMP tempfile() function. This allows for the clean conversion of the PDF type file to text through the usage of

“pdftotext” a PDF converter freely available from www.foolabs.com/xpdf/. The temporary file is opened using this program and its output is piped directly into a new string. The text received is then processed with a few text conversion and cleaning substitutions. These cleaning methods replace extra whitespace with single spaces and remove special characters that could not be read. Additionally words are dehyphenated if they cross a line boundary, and the ‘\n’ character is replaced with a single space. Upon completion of this process the temporary file is closed and automatically deleted.

The matching step is the same for the converted PDF files and the text/html pages that are contained in the string variable. First the trend is searched for in the text. Upon completion of this step all helper terms are searched for in the text. Finally the user is presented with the counts of the candidate emerging trend and the present helper terms. In the event that no terms are present, the user receives a message indicating this.

In the event that the user tries to examine a non-supported file type, the script returns a message indicating the supported types. The user interface section that follows details the layout of the results.

4.2.2 Term Extraction Interface

The design of the term extraction interface is focused on simplicity. The information that the user needs to perform the algorithm are the counts of the candidate emerging trend as well as the helper terms. Therefore a tabular approach to this is used starting with the trend count (m) and

followed by the individual helper term counts (n). This interface is labeled with the current trend that the user is examining. Additionally, a link to the URL that is supplied to the tool is also presented to provide an easy mechanism for the user to take a closer look at the documents that are of interest. The interface is shown below in (Figure 25).

TREND	aspect oriented
URL SEARCHED	http://www.pscit.monash.edu.au/~kendall/evolve2000.pdf

Trend	Count (m)
aspect oriented	4
Helper Term	Count (n)
proposed	2
recent research	1

Figure 8: Term Extraction Interface

4.2.3 Automatic Aggregation of Verification Metrics

As with the previous Perl script the input variables are transferred from the CIMEL Flash multimedia environment [21] through the CGI.pm interface. These variables include the user's ID, the candidate emerging trend, and login information for the database. The CGI script connects to a MS SQL Server 7.0 database through the DBI (Data Base Interface) library for Perl using the DBD::ODBC driver. The script starts by initializing its environment variables and taking a timestamp of the user's entry into the system. Next the parameters are decoded from the CGI query.

After decoding the user's request to the system, a database connection is established and the user's requested trend is searched for in the database. This search examines both the title and the abstract of each document using the Full Text Indexing service support of MS SQL Server 7.0. The result set of the search is then stored to a temporary table that is used to generate the aggregated information for the user. This table contains the field of UniqueStorageID, Title, VenueTitle, Abstract, and PublicationDate (refer to the tables and variables in Section 4.1.2).

The first set of information that is generated for the user is the document frequency by year. As in the previous script, an HTML document is generated dynamically for the user, and each table includes only the relevant information based on the context. The document frequency table is generated through an SQL query that counts the documents based on their publication date. These counts are then stored into an associative map variable with keys corresponding to their publication year. An HTML table starting with the maximum year and descending to the minimum year in the set is then generated and the total of the document frequencies is displayed as a caption to this table. The name of the trend in the search is listed above this initial table. Figure 9 presents the document frequency display for the trend “Aspect Oriented”.

Current Trend: Agent Oriented

YEAR	Document Count
2001	0
2000	0
1999	2
1998	3
1997	2
1996	1

Total Document Frequency: 8

Figure 9: Document Count Table

Next the list of Unique Authors is selected from the AuthorTable (Section 4.1.2). This list (Figure 10) corresponds to the authors of the documents present in the temporary table.

Unique Authors
Xu Dian-Xiang
Zheng Guo-Liang
Crnogorac, L.
Ramamohanarao, K.
Georgeff, M.
Rao, A.
Dianxiang Xu
Fan Xiao-Cong
Xiaocong Fan
Jiang Hui
Guoliang Zheng
Rao, A.S.
Xie Xiren
Kinny, D.
Poggi, A.
Lin Dong
Hou Jian-Min

Total: 17

Figure 10:Unique Author Table

Similarly the list of Unique Author Sets (Figure 11) is generated from the AuthorSetTable in the database. The number of authors and author sets is given after each respective table.

Unique Co-Author Sets
Crnogorac, L. : Ramamohanarao, K. : Rao, A.S. :
Crnogorac, L. : Rao, A.S. :
Dianxiang Xu : Guoliang Zheng : Xiaocong Fan :
Fan Xiao-Cong : Hou Jian-Min : Xu Dian-Xiang : Zheng Guo-Liang :
Georgeff, M. : Kinny, D. : Rao, A. :
Jiang Hui : Lin Dong : Xie Xiren :
Poggi, A. :

Total: 7

Figure 11: Unique Co-Author Table

The list of Unique Venues (Figure 12) is presented in a similar manner and is aggregated directly from the temporary table with a Select Distinct statement.

Unique Venues
Agents Breaking Away. 7th European Workshop on Modelling Autonomous Agents in a Multi-Agent World, MAAMAW `96. Proceedings
Australian Computer Science Communications
Chinese Journal of Computers
Information and Software Technology
International Journal on Artificial Intelligence Tools (Architectures, Languages, Algorithms)
Journal of Software
Proceedings of 15th International Joint Conference on Artificial Intelligence. IJCAI 97
Proceedings Technology of Object-Oriented Languages and Systems. TOOLS 31

Total: 8

Figure 12: Unique Venue Table

The final table that is presented by this script is the titles of the documents by year (Figure 13). These titles are selected along with their UniqueStorageID and PublicationDate from the temporary table and are presorted by the database itself.

YEAR	Document Titles
1999	Agent Class Methodology: a new kind of autonomous object generation methodology
1999	Research on inheritance of software agent
1998	The syntax and semantics of SPLAW-an agent specification and programming language
1998	Integrating object-oriented and rule-based programming for building agent systems
1998	A logic based language for networked agents
1997	Inheritance by extensions and restrictions in agent systems
1997	Analysis of inheritance mechanisms in agent-oriented programming
1996	A methodology and modelling technique for systems of BDI agents

Figure 13:Document Title Table

The titles are links to buttons that can be used to obtain a reference to the document as well as the abstract. The buttons themselves are linked to a JavaScript function that fills and submits a hidden form that loads the selected abstract display. This script logs the users request with the user name, UniqueStorageID, the action of getting an abstract, and a timestamp.

Finally after all tables are generated, the user's request is logged into the database with start and completion times. This log stores the userID, type of action, the trend requested, number of documents retrieved as well as the start and end time of the script.

4.2.4 Verification Interface

The design of the interface for the verification step is based closely on the key metrics that are used by the combined methodology in verifying an emerging trend. The document frequency, unique authors, unique co-author sets, unique venues, titles and abstracts of the candidate emerging trend's document set need to be presented to the user. These values give the users the

information they need (after completing their literature search) to verify whether the trend is indeed emerging in the main topic area.

There are a number of issues that were considered when designing this interface. The first is whether the interface should be linked directly with flash or appear in HTML loaded into a spawned browser window. One of the important goals for this interface is linking it smoothly with the CIMEL multimedia courseware environment. The issue with Flash however is that we cannot create dynamically the tables that are needed for the display of the verification data. The solution we choose to handle this problem and to maintain a smooth transition from the courseware environment to the web browser is to have the user enter the query in the flash environment and have the results displayed in the web browser. This ties the two interfaces together while resolving the issues that would result from using only one interface to handle the verification step. A sample of the interface results is shown in the previous section (Section 4.2.1) in Figures 8-13.

5.0 Experiments and Analysis of Results

In the following sections we discuss the testing and validation of the automated tools as well as the combined methodology itself as discussed in Sections 3.3-3.4. First the system performance testing methods are presented followed by the results of a statistical analysis of the usage logs for the system. Next we present the validation testing that was performed to evaluate the correctness of the database repository. Following this the usability for the term extraction and database

retrieval tools is discussed and the result of our user evaluation of these tools is presented. Finally the effectiveness of our combined methodology is evaluated.

5.1 Methodology of Evaluation

5.1.1 System Performance

System performance is measured using the average execution time for the term extraction and the database retrieval tool. The data for these averages comes from the experiment that is described in Section 5.3 for the evaluation of the effectiveness of the combined methodology. It is important that the average execution time is low because these tools are interfaced to web users through CGI. In future versions that target research users, the amount of data that needs to flow to and from the database could likely increase. Therefore the performance tests assist in evaluating the needs of future development in emerging trend detection.

The performance of this system is measured by the metric of average execution time. This metric is evaluated for each action of the system. These actions consist of term extraction, database retrieval, and abstract retrieval. There are three scripts present in the systems tools that perform these actions. The term extraction tool uses one script to perform the task of retrieving a web page and counting the words of interest. This execution time depends partially on the web server containing the page and the network response time.

The database retrieval tool however is composed of two scripts. The first script makes the initial database query and aggregates the data for the dynamically generated tables. This results in the

dynamic writing of links, for each document containing the chosen terms, to the second script. The second script retrieves the abstract as well as a bibliographic reference for corresponding documents.

The next section will summarize the statistical analysis of the average execution time for these tools.

5.1.2 Validation Methodology for Database Correctness

In generating the repository numerous data validation tests were performed to ensure that the data transfer from the collected data into the database tables was correct. Additionally many data cleaning tasks were performed on the collected data before insertion to prevent later problems in using the database for data mining tasks.

The first issue that was handled was a check to ensure all fields have valid values in the original XML data collection. The issue was that two different collections were extracted separately and needed to be merged together. The first collection was generated through manual extraction from the INSPEC repository into flat text files. Each file contained one hundred or fewer abstracts with indexing information. These files were combined and converted to the XML format of the second collection. After conversion the fields were checked and where data was missing, a missing value marker was inserted as appropriate. The second collection was extracted using an automated extraction tool, discussed in [24], that stored its results into an

XML file. The second collection contained NULL entries due to connection errors in the extraction tool. These were also removed.

The next step involved the combination of the two collections. The combined collection was then tested for duplicate documents based on the Document Title. Duplicates were removed from the collection by a data-cleaning program designed for this task. Finer grained cleaning was necessary for a few cases where the titles were the same except for a misspelling. After this the country of publication was normalized to have unabbreviated names to assist later data mining on this field.

After the data were cleaned, it was converted into input text files for each table using an in-house C++ table generation program. This program pulled the necessary data for each column and printed it to a delimited file for database insertion. Additional data cleaning was performed in this process as well as generation of a bibliographic reference. This program generated a unique ID for each abstract based on its source, data format, and a source indexing number. These changes were written to a new archive file for later usage.

The database itself was tested after insertion of the table to verify correct insertion. This process assisted in the identification of a wide number of data cleaning issues that were missed due to the size of the collection, or due to the misinterpretation of the table-forming program. These tests include browsing of the data in each table manually, as well as examination of query results for errors.

5.1.3 Evaluation of the Combined Methodology

The combined methodology, as described in Section 3.3, was tested in a controlled experiment in an undergraduate programming languages course at Lehigh University (CSC 262). The students in this course were assigned the task of identifying two emerging trends in the main topic area “Inheritance in Object Oriented Programming.” This main topic area was chosen due to its relevance to the course material as well as to simplify the evaluation of the experimental results. The students were evaluated on their precision of completing the assigned task. Precision is a standard metric of evaluation in the field of text mining. It is measured as the fraction of correctly retrieved items vs. the total number of retrieved items. Therefore for this experiment the number of correctly retrieved emerging trends vs. the total number of trends that a student retrieved formed the metric of precision.

For this experiment, the participating students were randomly divided into the two groups. The method used for this random division was to flip a coin for each student to choose their group. Each group contained 15 students at the start of the experiment. The second group, Group B, received an in-class lecture on inheritance as well as a multimedia lecture on inheritance and a multimedia tutorial of the combined methodology. The first group, Group A, received everything except the multimedia tutorial of the combined methodology and therefore used their own intuition in discovering emerging trends in this area.

The multimedia tutorial was designed to explain the concept of emerging trend detection as well as the usage of the combined methodology. A detailed description of its design and implementation can be found in [23]. The automated tools presented in this report are integrated

with an assignment that was presented in the tutorial. The tools were used in the assignment as part of the experiment to evaluate the efficiency of the combined methodology.

5.2 Results

5.2.1 Results of Performance Study

The following section presents the results of our performance studies on the automated tools. Table 18 shows the average execution time as well as the minimum and maximum execution times. The data for this study came from the usage logs we have gathered for each tool. These logs contain the user ID, which tool was used, and the start and end times.

	Term Extraction (secs)	Database Search (secs)	Return Abstract (secs)
Mean (Avg Execution Time)	2.14	0.55	0.14
Standard Error	0.43	0.10	0.05
Standard Deviation	3.49	1.33	0.35
Sample Variance	12.18	1.77	0.12
Minimum (Execution Time)	0	0	0
Maximum (Execution Time)	17	14	1
Sample Size	65	195	57

Table 18: Performance Statistics

On average, each of the tools has an excellent execution and response time. Clearly, there is room for improvement in terms of the maximum execution times. However since this is dependent on the size of the input as well as the network and server response times, this is the best handled in the user interface by introducing a mechanism to communicate estimated time to completion. The maximum execution time of 17 seconds was most likely due to network and server delays. This was a normal HTML file and on repeating the same term extraction an execution time of 1 second was observed.

5.2.2 Usability Study

The usability of the automated tools was studied to determine problem areas with the interface as well as the instructions and explanation that is provided with the tools. This study was handled with a statistical analysis of a user survey (completed by students in Group B of the combined methodology experiment that is presented in Section 5.3). These students used the methodology with the automated tools to detect emerging trends that is described in Section 3.3 as well as in the multimedia tutorial that is presented in [23].

The metrics that we addressed in our analysis were the ease of use and clarity of each tool's interface, as well as the user's view of the instructions provided for the tool in the tutorial. The section of the usability survey related to the automated tools can be found in the appendix of this report. Tables 19-20 show the results of this usability study for the tools that are presented in this report.

Metrics
i. The term extraction tool was useful.
ii. The term extraction tool was easy to use.
iii. The database search tool was useful.
iv. The database search tool was easy to use.
v. The instructions for the tools were clear.

Table 19: Metrics of Usability Evaluation

	<i>i</i>	<i>ii</i>	<i>iii</i>	<i>iv</i>	<i>v</i>
Mean	67.50%	47.50%	72.50%	65.00%	55.00%
Standard Deviation	33.44	34.26	18.45	26.87	24.44
Sample Variance	1118.06	1173.61	340.28	722.22	597.22
Minimum	12.50%	0.00%	25.00%	0.00%	12.50%
Maximum	100.00%	100.00%	87.50%	100.00%	87.50%
Sample Size	10.00	10.00	10.00	10.00	10.00

Table 20: Statistics of Usability Evaluation

The questions are ranked on percentage scale:

Agree Strongly	100%
Agree Somewhat	75%
No Opinion	50%
Disagree Somewhat	25%
Disagree Strongly	0%

Table 21: Answer Ranking List

Each metric is evaluated based on the user responses to one or more of the questions in the survey in Appendix I. The breakdown of the association of questions to metrics is seen in Table 22.

Metric	Questions
i	1, 3
ii	4
iii	6, 7
iv	2
v	5

Table 22: Metrics to Question Associations

The results from this usability study indicate that the automated tools are useful in completing the task of detecting emerging trends. The ease of use and the explanation of their usage are, however, in need of refinement.

One improvement that was suggested is the separation of the query boxes from the Flash multimedia. Some users of slower machines experienced problems when running the Flash interface and browsing the web for trends. Also, alternating between web browsing and the tutorial confused some users, therefore it seems that there should be a finer division between the actual assignment and the explanations of functionality of the tools and the assigned task. [23] describes the multimedia tutorial in greater detail.

Another important suggestion is the desire for an interface that would allow the user to select additional helper terms to use in a term extraction search. Additionally there was a desire to have a query interface to the database search script that mirrors web search engine functionality.

5.2.3 Results of Combined Methodology Evaluation

This section presents the evaluation of student precision for the task of emerging trends detection. Some students in group A reported more than the assigned two trends. A summary of the statistical evaluation is shown in Tables 23-24.

	Group A	Group B
Mean	21.43%	50%
Standard Deviation	26.5	47.14
Sample Variance	702.02	2222.22
Minimum	0%	0%
Maximum	66.67%	100%
Count	14	10
Confidence Level (95.0%)	15.3	33.72

Table 23: Analysis of Group A and Group B precision results

Hypothesis: Group B (with the methodology) will perform significantly better than Group A (without the methodology) in terms of precision.

Lower Tail test

A Lower Tail t-test is selected to evaluate the students' results. We are interested in determining if there is a difference between the mean precision of Group A and Group B. Additionally the Lower Tail t-test is chosen over the two tail test because we are interested in determining whether the mean precision of Group B is greater than Group A.

Population 1 sample corresponds to Group A (without methodology); Population 2 Sample corresponds to Group B (with methodology).

Null Hypothesis: (Mean precision of sample 1) – (Mean precision of sample 2) ≥ 0

Hypothesized Difference	0.00
Level of Significance	0.05
<hr/>	
Population 1 Sample	
Sample Mean	21.43
Sample Size	14
Sample Standard Deviation	26.50
<hr/>	
Population 2 Sample	
Sample Mean	50.00
Sample Size	10
Sample Standard Deviation	47.14
Population 1 Sample Degrees of Freedom	13
Population 2 Sample Degrees of Freedom	9
Total Degrees of Freedom	22
Pooled Variance	1324.04
Difference in Sample Means	-28.57
<i>t-Test Statistic</i>	-1.90
<hr/>	
Lower-Tail Test	
Lower Critical Value	-1.72
p-Value	0.04
Reject the null hypothesis	
<hr/>	

Table 24: Lower Tail Test Results

As shown in table 24, the difference in sample means between sample 1 (without methodology) and sample 2 (with methodology) is less than 0 (-28.57) with a confidence level of 95%. Thus by the Lower Tail t-test sample 2 precision results (with the methodology) are significantly greater than sample 1 (without the methodology) and the null hypothesis is rejected.

5.2.4 Discussion of Results

The results of the evaluation of the combined methodology are promising. The students with the methodology (Group B) and automated tools out-performed the students without the methodology (Group A) in terms of precision.

From the initial participants there are a number of students who did not return the assignment before the deadline. This subject loss however does not affect the randomness of the selection process for the two groups [26]. Group A contained 14 students, while Group B contained 10 after this loss of subjects.

6.0 Conclusions and Future Work

This report summarizes the work that is in progress for the full automation of a combined manual methodology for the detection of emerging trends. The combined methodology represents the work of myself and another student in characterizing the manual task of emerging trend detection. This methodology is shown to improve precision in the detection of emerging trends with a confidence of 95%.

The ultimate goal of this research is to develop a precise fully automatic system by examining how domain experts validate emerging trends. In order to improve the efficiency of the presented methodology the term extraction and database search systems were developed. The evaluation of these tools show that they improve the usability of the methodology and aid the task of emerging trend detection. However, the instructions for these tools need to be improved, and users expressed a desire for increased flexibility.

Our current plans for improving this system are to expand the Term Extraction tool to gather the first one hundred hits from a search engine in response to a user-entered query and to dynamically count the terms of interest. Using the returned counts the documents will be ranked and the results presented in a ranked listing. This will speed the usage of the methodology by removing some of the tediousness of web search. However there will need to be some work involved with implementing a dynamically updating interface to improve the response time of such a system. This would allow the user to immediately begin the evaluation of these links.

Additionally functionality could be implemented to allow the user to select the number of links to be processed to override the default of the first one hundred links. This would allow the user to tailor the search to a narrower or broader range of links. The term extraction itself could be improved by enabling the user to specify multiple trends or additional helper terms to be identified in the search of the links. Finally the user could be presented with the option to preprocess the list of links by checking those that are of interest to the user.

An improvement on the efficiency of the system would be to process sets of links in parallel. Given the appropriate bandwidth this could reduce the overall time needed to process the required number of links.

Another future goal we have is to improve our own data repository by adding other subject areas as well as improving on our current areas. This will involve the integration of an improved data collection tool that will be able to link directly to the database and update the repository on a continual basis as new documents become available. Additionally the number of sources can be increased, and the tool could provide a flexible means of identifying the fields for a source dynamically.

Concurrently with these other goals we are working to acquire citation information for the documents we currently have available from a comprehensive electronic source of this information. Citations are semantically rich in information related to the flow of a trend over time. Gaining this information will allow us to explore topic and author relationships in more depth. Using these relations we will be better able to characterize a trend automatically.

Additionally, in automating this system there needs to be a Conference/Workshop Web Crawler developed. It is important that we automate the search for candidate emerging trends in a dynamic web resource environment. This will require the implementation of domain expertise into a automatic system to identify key terms in the assigned resources to be used for the detection of emerging trends.

To make emerging trend detection a realizable goal more work needs to be done in the area of trend characterization. We are still looking into different ways to express a trend. The current term-based approach works reasonably in most cases. However it is our intuition that this does not fully capture the semantics involved with describing the real underlying innovation. We plan to focus, in our research, on the semantic matching of terms and linguistic features in order to use the meaning of a term to match on more than words alone. For a more precise verification we need to incorporate such capabilities with our database search tool to provide a semantic search utility. Efficient emerging trend detection needs to surpass term/word matching and match the underlying semantics of a trend to the semantics of a document. This will require sophisticated matching algorithms and would benefit greatly from a full text repository to search on. What we envision is a method of assigning semantic descriptors to a candidate emerging trend as well as documents and abstracts.

Further development of web crawling utilities will enable larger scale data mining and trend detection software. A version of the emerging trend detection software may be integrated with a multilevel system that gathers topic information from institutional web-pages, conferences and workshops, as well as online abstract repositories. This will allow the mapping of topics and trends between each of these sources and facilitate many useful search strategies. The utility of being able to link a topic to researchers and institutions as well as related documents is of tremendous benefit to accelerating the speed at which knowledge is discovered.

References

- [1] Alan L. Porter and Michael J. Detampel. Technology Opportunities Analysis. *Technological Forecasting and Social Change*, Vol 49, 237-255, 1995.
- [2] H. D. White and K. W McCain. *Bibliometrics*. Annual Review of Information Science and Technology, Elsevier, Amsterdam, Vol 24, 119-186, 1989.
- [3] William M. Pottenger and Ting-hao Yang. *Detecting Emerging Concepts in Textual Data Mining*. Computational Information Retrieval, Michael Berry, Ed., SIAM, Philadelphia, PA, August 2001.
- [4] Fabien Bouskila, William M. Pottenger. The Role of Semantic Locality in Hierarchical Distributed Dynamic Indexing. *In Proceedings of the 2000 International Conference on Artificial Intelligence (IC-AI 2000)*, Las Vegas, Nevada, June 2000.
- [5] Susan Havre, Beth Hetzler and Lucy Nowell. ThemeRiverTM: In Search of Trends, Patterns, and Relationships. Battelle Pacific Northwest Division. Presented at *IEEE Symposium on Information Visualization*, InfoVis '99, San Francisco CA, October 25-26 1999.
- [6] Pak Chung Wong, Wendy Cowley, Harlan Foote, Elizabeth Jurus, Jim Thomas. Visualizing Sequential Patterns for Text Mining. Pacific Northwest National Laboratory. *In Proceedings of IEEE Information Visualization 2000*, October 2000.

- [7] L. T Nowell, R. K France, D. Hix, L. S Heath and E. A Fox. Visualizing Search Results: Some Alternatives to Query-Document Similarity. In *Proceedings of SIGIR'96*, Zurich, 1996
- [8] Y. Yang, J. Carbonell, R. Brown, T. Pierce, B.T Archibald, X. Liu. Learning Approaches for Detecting and Tracking News Events. *IEEE Intelligent Systems: Special Issue on Applications of Intelligent Information Retrieval*, Vol 14(4), 32-43, 1999.
- [9] T. Yang, *Detecting Emerging Contextual Concepts in Textual Collections*. M.S. Thesis, Department of Computer Science at the University of Illinois at Urbana-Champaign, 2000.
- [10] L. Zhou, *Machine Learning Classification For Detecting Trends In Textual Collections*. M.S. Thesis, Department of Computer Science at the University of Illinois at Urbana-Champaign, 2001.
- [11] Russel Swan and David Jensen. TimeMines: Constructing Timelines with Statistical Models of Word Usage. In *The Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2000.
- [12] S.K. Murthy. *On growing better decision trees from data*. Doctoral dissertation, University of Maryland, 1997.

- [13] Yumi Jin. *Graphical User Interface and Information Visualization Techniques for Detection of Emerging Concepts*. M.S. Thesis, Department of Computer Science at the University of Illinois at Urbana-Champaign, December 2000
- [14] W. Pottenger, Y. Kim, and D. Meling. *Data Mining for Scientific and Engineering Applications*, chapter HDDI™. Hierarchical Distributed Dynamic Indexing. R. Grossman and C. Kamath and V. Kumar and R. Namburu, Eds., 2001.
- [15] Quinlan, J. R. Induction of decision trees. *Machine Learning*, 1:81--106, 1986.
- [16] Quinlan, J. R. *C4.5: Programs for machine learning*. San Mateo, CA: Morgan Kaufmann 1993.
- [17] Cezary Z. Janikow. Exemplar Learning in Fuzzy Decision Trees. In *Proceedings of FUZZ-IEEE 1996*, pp. 1500-1505.
- [18] Yuan, Y.F. and Shaw, M.J. Induction of Fuzzy Decision Trees. *Fuzzy Sets and Systems*. Vol 69, Iss 2, pp 125-139, 1995.
- [19] Price, D.D. Networks of Scientific Papers. *Science*. Vol. 149, pp 510-515, 1965.

- [20] Alexandrin Popescul, Gary William Flake, Steve Lawrence, Lyle H. Ungar, C. Lee Giles. Clustering and Identifying Temporal Trends in Document Databases, IEEE Advances in Digital Libraries, ADL 2000, Washington, D.C., May 22-24, pp. 173-182, 2000.
- [21] Glenn D. Blank, William M. Pottenger, G. Drew Kessler, Soma Roy, David R. Gevry, Jeff Heigl, Shreeram Sahasrabudhe and Qiang Wang. Design and Evaluation of Multimedia to Teach Java and Object-Oriented Software Engineering. In *2002 ASEE Annual Conference and Exposition*, June 16-19, 2002.
- [22] David R. Gevry. *Detection of Emerging Trends: Automation of Domain Expert Practices*. M.S. Thesis. Department of Computer Science and Engineering at Lehigh University. June 2002.
- [23] S. Roy, *A Multimedia Interface for Emerging Trend Detection in Inquiry Based Learning*. M.S. Thesis, Department of Computer Science and Engineering at Lehigh University, 2002.
- [24] J. Eynon. *A Generic Learning Method For Information Extraction From The World Wide Web*. M.S. Thesis, Department of Computer Science and Engineering at Lehigh University, 2002.
- [25] SQL Server 7.0 Online Documentation, Retrieved on April 22, 2002 from <http://www.microsoft.com/sql/techinfo/productdoc/70/books.asp>

[26] Geoffrey Keppel, William H. Saufley, Jr., Howard Tokunaga. Introduction to Design & Analysis A Student's Handbook. Second Edition. New York: W.H. Freeman and Company, 1992.

[27] Soma Roy, David Gevry, William Pottenger. Methodologies for Trend Detection in Textual Data Mining. *Proceedings of the Textmine '02 Workshop, Second SIAM International Conference on Data Mining*. April, 2002.

Appendix I

Usability Study: Term Extraction Tool

(The tool to lookup Candidate Emerging Trends in a URL in Step 2 of the assignment)

1. The term extraction tool was useful.
 - a. Agree Strongly
 - b. Agree Somewhat
 - c. No Opinion
 - d. Disagree Somewhat
 - e. Disagree Strongly

2. Usage of the term extraction tool was explained well.
 - a. Agree Strongly
 - b. Agree Somewhat
 - c. No Opinion
 - d. Disagree Somewhat
 - e. Disagree Strongly

3. The term extraction tool made the Algorithm (used in Step 2) easier to use.
 - a. Agree Strongly
 - b. Agree Somewhat
 - c. No Opinion
 - d. Disagree Somewhat

- e. Disagree Strongly
4. The web interface for the term extraction tool was intuitive.
- a. Agree Strongly
 - b. Agree Somewhat
 - c. No Opinion
 - d. Disagree Somewhat
 - e. Disagree Strongly
5. When using the term extraction tool to obtain the term counts for html and PDF documents did you experience any problems?
6. What improvements would you like to see for the term extraction tool? (e.g. Document formats, layout changes,...)

Database Tool

(in Step 3 of Assignment)

7. Usage of the database search tool was explained well.
- a. Agree Strongly

- b. Agree Somewhat
 - c. No Opinion
 - d. Disagree Somewhat
 - e. Disagree Strongly
8. The web interface for the database search tool was intuitive.
- a. Agree Strongly
 - b. Agree Somewhat
 - c. No Opinion
 - d. Disagree Somewhat
 - e. Disagree Strongly
9. The document frequency, author, and venue listing helped me in verifying a candidate trend.
- a. Agree Strongly
 - b. Agree Somewhat
 - c. No Opinion
 - d. Disagree Somewhat
 - e. Disagree Strongly
10. The ability to view the abstracts for the documents was beneficial to the verification process.
- a. Agree Strongly

- b. Agree Somewhat
- c. No Opinion
- d. Disagree Somewhat
- e. Disagree Strongly

11. Were there any problems with the database search tool that you encountered?

12. Did the verification step lose you? Was there a point where you were unsure how to proceed?

Appendix II

Database Search Tool

```
#!/usr/bin/perl -w
use DBI;
use DBD::ODBC;
use CGI qw(:all);
sub DOCFreq;
sub UniqueAuthor;
sub UniqueCoAuthorSets;
sub UniqueVenue;
sub Title;
sub ScriptForm;

#Get Start Time
($sec,$min,$hr,$mday,$mon,$year)=localtime();
$year+=1900;
$mon++;
$StartTime=$year."-".$mon."-".$mday." ".$hr.":".$min.":".$sec;

# environment variables
if(param()){

$database = "dbi:ODBC:ETrend";

$login =param('userID');
```

```

$password =param('id');
$username=param('login');
$Term=param('trend');
$UID="N/A";
$action="DisplayTrend";
$DocTotal=0;

if($login eq ""){ $login="test";}
$stable="###".$login.$hr.$min.$sec;

#Set Search String
$search = "INSERT INTO $table
          SELECT UniqueStorageID,Title, VenueTitle, Abstract, PublicationDate
          FROM RecordTable WHERE ";
$titleFront="CONTAINS( Title, \' \'"';
$back="\' \)";
$or=" OR ";
$abstractFront="CONTAINS(Abstract, \' \'"';
$search=$search.$titleFront.$term.$back.$or.$abstractFront.$term.$back;
#Database Connect
$dbh=DBI->connect($database, $username, $password) or die print "Database Unavailable\n";
$dbh->{LongReadLen}=512*1024;
#Database Select
$sth=$dbh->prepare("USE Cime1_E_T_R");
$sth->execute();

#Create Temporary Table for Search
$tableCreateQuery=" CREATE TABLE $table (UniqueStorageID varchar(50),Title text,
          VenueTitle varchar(1000),Abstract text,PublicationDate datetime)";
$sth=$dbh->prepare($tableCreateQuery);
$sth->execute();
#Execute Search
$sth=$dbh->prepare($search);
$sth->execute();

{
local ($oldbar) = $|;
$cfh = select (STDOUT);
$| = 1;
#HTML Header information
print "Context: text/html \n\n";
print "<HTML>\n<HEAD>\n<TITLE>Incipient Emerging Trends</TITLE>\n</HEAD>\n";

#Write JavaScript
ScriptForm();
print "<BODY>\n<p style=\"font-family:Times;font-size:25;color:green\"><B><U>Current Trend:</U>
$Term</B></p>\n";
#Generate Table Results
DOCFreq();
UniqueAuthor();
UniqueCoAuthorSets();
UniqueVenue();
Titles();
#End Html
print "</BODY>\n";

```

```

print "</HTML>\n\n";

$| = $oldbar;
select ($cfh);
}
#Get End Time
($sec,$min,$hr,$mday,$mon,$year)=localtime();
$year+=1900;
$mon++;
$EndTime=$year."-".$mon."-".$mday." ".$hr.":".$min.":".$sec;
#Insert Tracking log and disconnect
$dbh->do("INSERT INTO LogTable VALUES(?,?,?,?,?,DEFAULT)",
        undef, $login, $Term, $action, $UID,$DocTotal,$StartTime,$EndTime);
$dbh->disconnect();

}

else{

print "Context: text/html\n\n<html><body><B>ERROR IN CGI. Please contact
Administrator.</B></body></html>\n";

}

sub DOCFreq{
#Storage and counter variables
my %map;
my $MinYear=5000;
my $Counter=2001;
my $Total=0;
#Selection of Document Frequency
my $DocFrequencyCount="SELECT YEAR(PublicationDate) 'Publication Date', COUNT(UniqueStorageID)
        'Count' FROM $table GROUP BY PublicationDate";

#execution of selection
$sth=$dbh->prepare($DocFrequencyCount);
$sth->execute();
#Retrieval and ordering loop.
while( (my $Year,my $Count)=$sth->fetchrow_array){
    $map{ $Year } = $Count;
    $Total+=$Count;
    #Set Min and Max Years
    if(($Year)<($MinYear)){ $MinYear=$Year;}
    if($Year>$Counter){ $Counter=$Year;}
}
#print Document Frequency Table
print "<P><TABLE BORDER=\\"1\" CELLPADDING=\\"1\">\n";
print "<TR><TH>YEAR</TH><TH>Document Count</TH></TR>\n";
# loop over all years present from max to min year
while(($Counter>$MinYear-1)){
    print "<TR><TD>$Counter</TD><TD>";
    if($map{ $Counter }){
        print "$map{ $Counter}";
    }
}
else{

```

```

    print "0";
    }
    print "</TD></TR>\n";
    $Counter--;
    }

    print "</TABLE></P>\n";
    #Print total frequency
    print "<CAPTION>Total: $Total</CAPTION></P>\n";
    $DocTotal=$Total;
    }

sub UniqueAuthors{
    my $Count=0;

    #Selection of Unique Authors
    my $UAuthorQuery="SELECT DISTINCT A1.Author
        FROM AuthorTable AS A1, $table
        WHERE $table.UniqueStorageID=A1.UniqueStorageID";

    #Execution of Script
    $sth=$dbh->prepare($UAuthorQuery);
    $sth->execute();

    #Retrieval of results
    my @Authors;
    while(my $AuthorName = $sth->fetchrow_array){
        push @Authors, $AuthorName;
    }

    #Printing of Table
    print "<P><TABLE BORDER='1' CELLPADDING='1'\>\n<TR><TH>Unique Authors</TH></TR>\n";

    while($Count< $#Authors+1){
        print "<TR><TD>".($Authors[$Count])."</TD></TR>\n";
        $Count++;
    }
    print "</TABLE>\n";
    print "<CAPTION>Total: $Count</CAPTION></P>\n";
    }

sub UniqueCoAuthorSets{
    my $Count=0;

    #Selection and execution of query
    my $UCASQuery="SELECT DISTINCT A1.AuthorSet
        FROM AuthorSetTable AS A1, $table
        WHERE $table.UniqueStorageID=A1.UniqueStorageID
        ";
    $sth=$dbh->prepare($UCASQuery);
    $sth->execute();

    #Retrieval of Results
    my @Authors;
    while(my $AuthorName = $sth->fetchrow_array){

```

```

    push @Authors, $AuthorName;
}

#Printing of Table
print "<P><TABLE BORDER='1' CELLPADDING='1'>\n<TR><TH>Unique Co-Author
Sets</TH></TR>\n";

while($Count< $#Authors+1){
    print "<TR><TD>$Authors[$Count]</TD><TR>\n";
    $Count++;
}
print "</TABLE>\n";
print "<CAPTION>Total: $Count </CAPTION></P>\n";
}

sub UniqueVenue{
my $Count=0;

#Selection and execution of query
my $UVenueQuery="SELECT DISTINCT VenueTitle
    FROM $table";
$sth=$dbh->prepare($UVenueQuery);
$sth->execute();

#Retrieval of Results
while(my $VenueName = $sth->fetchrow_array){
    push @Venues, $VenueName;
}

#Printing of Table
print "<P><TABLE BORDER='1' CELLPADDING='1'>\n<TR><TH>Unique Venues</TH></TR>\n";

while($Count< $#Venues+1){
    print "<TR><TD>".($Venues[$Count])."</TD><TR>\n";
    $Count++;
}
print "</TABLE>\n";
print "<CAPTION>Total: $Count</CAPTION></P>\n";
}

sub Titles{
my $Count=0;
my @Titles;
my $TitleName;
my @PublicationDates;
my @UniqueIDs;

#Selection and execution of query
my $TitleQuery="SELECT YEAR(PublicationDate)'PublicationDate',Title,UniqueStorageID
    FROM $table ORDER BY PublicationDate DESC";
$sth=$dbh->prepare($TitleQuery);
$sth->execute();

#Retrieval of Results
while((my $PublicationDate,my $Title, my $UniqueID) = $sth->fetchrow_array){

```

```

push @Titles, $Title;
push @PublicationDates, $PublicationDate;
push @UniqueIDs, $UniqueID;
}

#Printing of Table
print "<P><TABLE BORDER=\"1\" CELLPADDING=\"1\">\n<TR><TH>YEAR</TH><TH>Document
Titles</TH></TR>\n";
while($Count< $#Titles+1){
print "<TR><TD>". $PublicationDates[$Count]. "</TD>\n";
$TitleName=$Titles[$Count];

print "<TD><input type=\"Button\" value=\"$TitleName\" onClick=\"GetAbstract('\$UniqueIDs[$Count]\");"
></TD></TR>\n";
$Count++;
}
print "</TABLE>\n";
print "<CAPTION>Total: $Count<br></CAPTION></P>\n";
}

```

```

sub ScriptForm{
#Print JavaScript for Abstract Retrieval
print "<script Language=JavaScript>\n";
print "function GetAbstract(UniqueID){\n";
print "var form=document.ABSform;\n";
print "document.ABSform.user.value=\"$username\";\n";
print "document.ABSform.id.value=\"$password\";\n";
print "document.ABSform.login.value=\"$login\";\n";
print "document.ABSform.UID.value=UniqueID;\n";
print "document.ABSform.Trend.value=\"$Term\";\n";
print "form.submit();\n}\n</script>\n";
print "<form name=\"ABSform\" Method=\"POST\" action=\"\\Scripts\\getABS.pl\">\n";
print "<input type=\"hidden\" name=\"user\" value=\"\">\n";
print "<input type=\"hidden\" name=\"id\" value=\"\">\n";
print "<input type=\"hidden\" name=\"login\" value=\"\">\n";
print "<input type=\"hidden\" name=\"UID\" value=\"\">\n";
print "<input type=\"hidden\" name=\"Trend\" value=\"\">\n";
print "</form>\n";
}
exit;

```

Get Abstract Script

```

#!/usr/bin/perl/
use DBI;
use DBD::ODBC;
use CGI qw(:all);

#Get StartTime
($sec,$min,$hr,$mday,$mon,$year)=localtime();
$year+=1900;
$mon++;
$StartTime=$year."-".$mon."-".$mday." ".$hr.".".$min.".".$sec;

# enviornment variables
if(param()){

```

```

$database = "dbi:ODBC:ETrend";

$login = param('login');
$password = param('id');
$username=param('user');
$UID=param('UID');
$Term=param('Trend');
$action="GetAbs";
$NA=0;
$Count=0;

#Database Connect
my $dbh=DBI->connect($database, $username, $password) or die print "Database Unavailable\n";
$dbh->{LongReadLen}=512*1024;
#Database Select
$sth=$dbh->prepare("USE Cime1_E_T_R");
$sth->execute();

#Get Abstract
$search="SELECT MaterialIdentification,Abstract FROM RecordTable Where UniqueStorageID='".$UID.'"";
$sth=$dbh->prepare($search);
$sth->execute();
@Reference;
@Abstract;
{
    local ($oldbar) = $|;
    $cfh = select (STDOUT);
    $| = 1;
#Print HTML Header
print "Context: text/html\n<html>\n<head><title>Abstract View</title></head>\n";
print "<body>\n<table border='1'\n";
#Handle the case of multiple abstracts
while(($Ref, $Abs)=$sth->fetchrow_array){
    push @Reference,$Ref;
    push @Abstract, $Abs;
}
while($Count< $#Reference+1){
print "<tr><th>Reference:</th><td>$Reference[$Count]</td></tr>\n";
print "<tr><th>Abstract:</th><td>$Abstract[$Count]</td></tr>";
$Count++;
}

#End HTML
print "</table>\n</body></html>\n\n";

$| = $oldbar;
select ($cfh);
}
#Get EndTime
($sec,$min,$hr,$mday,$mon,$year)=localtime();
$year+=1900;
$mon++;
$EndTime=$year."-".$mon."-".$mday." ".$hr.".".$min.".".$sec;

```

```

#Insert Log and disconnect
$dbh->do("INSERT INTO LogTable VALUES(?,?,?,?,?,DEFAULT)", undef, $login, $Term, $action,
$UID,$NA,$StartTime,$EndTime);
$dbh->disconnect();
}
exit;

```

Term Extraction Script

```

#!/usr/bin/Perl -w
#CGI Interface Tool
use CGI qw(:all);
#Database interface Library and Driver
use DBI;
use DBD::ODBC;
#Web interface tools
use LWP::UserAgent;
use HTTP::Request;
use HTTP::Headers;
#temporary file library
use File::Temp qw(tempfile);

#subroutines
sub CountTerms;
sub printHeaders;
sub Connection;

#Environment and Globals
our $types="text/html, application/pdf";
our $PDFTOTEXT="./TMP/pdftotext";
our $Suffix=".pdf";
our $dir="./TMP/";
our $contents="";
our @HelperTerms=("current issues", "cutting edge strategies", "emergent", "first public review",
"future", "hot topics", "most recent contribution", "next generation", "a new paradigm",
"new approach", "newest entry", "novel", "proposed", "recent research", "recent trend" );

$action="UrlStats";
$UID="N/A";
$na=0;
#Get StartTime
($sec,$min,$hr,$mday,$mon,$year)=localtime();
$year+=1900;
$mon++;
$StartTime=$year."-".$mon."-".$mday." ".$hr.":"$.min.".$sec;
$ISPDF=0;

#Database setup variables
$dbase = "dbi:ODBC:ETrend";
$username="";
$password="";

#If Cgi parameters are present in STDIN
if(param()){

```

```

#CGI Parameters
our $TREND=param('Trend');
our $url=param('URL');
our $user=param('userID');
$username=param('Dbuser');
$password=( 'Dbpass');
our $template=$user."XXXX";
$TrendDH=$TREND;
#Remove dashes
$TrendDH=~ s/-/ /g;

#Database connect
$dbh=DBI->connect($database, $username, $password) or die print "Database Unavailable\n";
$dbh->{LongReadLen}=512*1024;
#Database Selection
$sth=$dbh->prepare("USE CimeI_E_T_R");
$sth->execute();

if($user eq ""){ $user="IDnull";}
if($url!~ m/http:\|\|/){ $url="http://".$url;}
#Connect to website
my $res=Connection();
{
local ($oldbar) = $|;
$cfh = select (STDOUT);
$| = 1;

# check the outcome
if ($res->is_success) {

our $head= $res->header('Content-type');
$content=$res->content;
if($head=~ m/application\/pdf/){
$ISPDF=1;
my $temp="";
(my $handle,my $filename) = tempfile($template, SUFFIX=> '.pdf',DIR =>$dir, UNLINK=>1);
print $handle $content;
open(PDF, "$PDFTOTEXT -raw $filename - |") || die "$PDFTOTEXT doesn't want to be opened using
pipe\n";
#Format text from PDF
while (<PDF>) {
while ( m/[A-Za-z\300-\377]-\s*$/) {
$_ .= <PDF>;
last if eof;
s/([A-Za-z\300-\377])-\s*\n\s*([A-Za-z\300-\377])/1$2/s;
}
s/\255/-/g; # replace dashes with hyphens
# replace special characters with a space:
s/[\000-\040]+\s+ /g;
s/\f\n/g;
$temp.=$_;
}
close(PDF);
$content=$temp;

```

```

    CountTerms();
}
elseif($head=~ m/text/html/){
    CountTerms();
}
else {
    print "Content-type: text/html\n\n<HTML><Body><h1 style=\"color:red\">$url</h1><Strong><B>This file
type is not supported, currently only text, html, and pdf file types are
supported.</B></Strong></Body></HTML>\n\n";
}
}
else {
    print "Content-type: text/html\n\n<HTML><Body><Strong><B>Error: " . $res->status_line
."</B></Strong></Body></HTML>\n\n\n";
}
$| = $oldbar;
select ($cfh);
}
}
#Get EndTime
($sec,$min,$hr,$mday,$mon,$year)=localtime();
$year+=1900;
$mon++;
$EndTime=$year."-".$mon."-".$mday." ".$hr.":".$min.":".$sec;
#Insert Tracking Log and Disconnect
$dbh->do("INSERT INTO LogTable VALUES(?,?,?,?,?,DEFAULT)", undef, $user, $TREND, $action,
$url,$na,$StartTime,$EndTime);
$dbh->disconnect();
exit;

#Count the Trend and Helper Terms in the document if supported
sub CountTerms{
    my $iterations=0;
    my $TrendCount=0;
    my @HelperCount;
    my $Count=0;
    my $temp =$contents;
    my $CountsPresent=0;
    my $helperpresent=0;

#remove all dashes
    $temp=~ s/-/ /g;
    if($TREND ne ""){
#Check for the trend, and count occurrences
        while ($temp =~ /($TREND)|($TrendDH)/gi) { $TrendCount++ }
#Check for each helper term and count occurrences
        foreach my $term (@HelperTerms) {
            while ($temp =~ /$term/gi) { ($HelperCount[$iterations])++ }
            $iterations++;
        }
    }
}

#Print Table information
&printHeader();

```

```

print "<BODY><p><Table border=\"1\"><TR><TH style=\"font-family:times;color:green;font-size:24\">TREND</TH>\n<TD style=\"font-family:times;color:green;font-size:24\">$TREND</TD></TR>\n";
print "<TR><TH style=\"font-family:times;color:blue;font-size:20\">URL SEARCHED</TH><TD><a href=$url>$url</a></TD></TR></Table></p>";
$CountsPresent=$TrendCount;

while(($HelperPresent==0)&&($Iterations>0)&&($Count<$Iterations)){

    $CountsPresent+=$HelperCount[$Count];
    $HelperPresent+=$HelperCount[$Count];

    $Count++;
}
if($CountsPresent>0){
print "<Table Border=\"1\" Cellspacing=\"1\"><TR><TH>Trend</TH><TH>Count (m)</TH></TR><TR>\n<TD>$TREND</TD><TD>$TrendCount</TD></TR>\n";
    $Count=0;

    print "<TR><TH>Helper Term</TH><TH>Count (n)</TH><TR>\n";
    while($Count<$Iterations){
        if($HelperCount[$Count]>0){
            print "<TR><TD>$HelperTerms[$Count]</TD><TD>$HelperCount[$Count]</TD></TR>\n";
        }
        $Count++;
    }

    print "</Table>\n";
}
#Print message if no terms are found
else{
    print "\n\n<P style=\"font-family:times;color:red;font-size:12\"><B>No Terms Present or Found in text</B></P>";
    if($ISPDF==1){
        print "<P style=\"font-family:times;color:red;font-size:12\"><B>Due to the nature of PDF files Terms may have been missed</B></P>";
    }
}
#End HTML
print "</BODY></HTML>\n\n";
}

#Setup Database Connection
sub Connection{
    my $ua = LWP::UserAgent->new;
    $ua->agent("$0/0.1 " . $ua->agent);
    my $req = HTTP::Request->new(GET =>$url);
    $req->header('Accept' => $types);

    # send request
    my $res = $ua->request($req);
    return $res;
}
#Print HTML Header
sub printHeader{
    print "Content-type: text/html\n\n";
    print "<HTML><HEAD><TITLE>Url Statistics</TITLE></HEAD>\n";
}

```

```
}
```

Appendix III

```
//Creates the tables based on the document source. In
//this version only INSPEC source from Jeff Eyon's XML layout
//is supported. The Tables created are in a text format for insertion
//by way of the bcp utility into SQL Server 7.0/2000

#include <fstream.h>
#include <stdlib.h>
#include <map>
#include <string>
#include <sstream>
#include <vector>
#include <algorithm>
using namespace std;
#include "XmlExtractor.h" //Generates an STL Map of each field in an Item
#include "trim.h" //General trim function for an STL String

string AcceptResponse();
bool QueryUser( string& );
string GetNextSubTag( istrstream& );
void SetGlobals();
void CreateTableInput(ofstream&);
void CreateRecordTable( map<string,string>, ofstream&, ofstream & );
void CreateTreatmentTable( map<string,string>, ofstream& );
void CreateAuthorTable( map<string,string>, ofstream& );
void CreateINSPECTable( map<string,string>, ofstream& );
void CreateNewRepository( map<string,string>, ofstream& );
void CreateConferenceTable( map<string,string>, ofstream& );
void CreateAuthorSetTable( map<string,string> DocumentMap, ofstream& AuthorTableOut);
void PrintDate( string , map<string,string> , ofstream& , ofstream & );
void TableFieldPrint( string , map<string,string> , ofstream& );
void TableFieldPrintEND( string , map<string,string> , ofstream& );
void outputField( string , string , ofstream & , map<string,string> & );
void FormReference( map<string,string> , ofstream& );
void AddDate( map<string,string> DM, string & Reference);
void AddTitle( map<string,string> DM, string & Reference);
void AddVenue( map<string,string> DM, string & Reference);
void AddPublisher( map<string,string> DM, string & Reference);
void AddVolumeIssueAndPages( map<string,string> DM, string & Reference);
void GetAuthors( map<string,string> DM, string & Reference);
void CountryFix( ofstream& out, map<string,string> DM);
void FixAbstract( ofstream&, map<string,string>);
string RepositoryName="";
string TableName="";
string fieldDelimiter="***F***";
string rowDelimiter="***R***";
string RepositoryType="INSPEC"; // For future multi-source version
string IDPrefix="INSP-TEXT-";
string UniqueID;
map<string,int> IDMAP;
```

```

void main(){
    ofstream error;
    SetGlobals();
    CreateTableInput(error);
    error.close();
}

//Contains all user Global variables to be set.
void SetGlobals(){

    string Response;
    while(!QueryUser( RepositoryName )){ };

    cout<<"Would you like to Reset the delimiters?(***F*** and ***R***)"
        <<"\n Field and Row Delimiters.<Y/N>";
    Response=AcceptResponse();
    while(Response!="Y"&&Response!="N"&&Response!="y"&&Response!="n"){
        cout<<"Please enter Y or N";
        Response=AcceptResponse();
    }
    if(Response=="Y"||Response=="y"){
        cout<<"Please enter the new field delimiter";
        fieldDelimiter=AcceptResponse();
        cout<<"Please enter the new row delimiter";
        rowDelimiter=AcceptResponse();
    }
}

void CreateTableInput(ofstream & error)
{
    //Set Table FileStreams
    error.open("Error.txt");
    ofstream RecordTableOut;
    string FileName="RC-";
    FileName+=RepositoryName;
    RecordTableOut.open(FileName.c_str());
    ofstream TreatmentTableOut;
    FileName="TC-";
    FileName+=RepositoryName;
    TreatmentTableOut.open(FileName.c_str());
    ofstream AuthorSetTableOut;
    FileName="AST-";
    FileName+=RepositoryName;
    AuthorSetTableOut.open(FileName.c_str());
    ofstream AuthorTableOut;
    FileName="AC-";
    FileName+=RepositoryName;
    AuthorTableOut.open(FileName.c_str());
    ofstream INSPECTableOut;
    FileName="IC-";
    FileName+=RepositoryName;
    INSPECTableOut.open(FileName.c_str());
    ofstream ConferenceTableOut;
    FileName="ConF-";
    FileName+=RepositoryName;
}

```

```

ConferenceTableOut.open(FileName.c_str());
ofstream NewRepositoryOut;
FileName="Final-";
FileName+=RepositoryName;
NewRepositoryOut.open(FileName.c_str());

map<string,string> DocumentMap;
bool RetrievedDocument=false;
NewRepositoryOut<<"<DOCUMENT>\n\n";
XMLExtractor RepositoryToNumber(RepositoryName);
DocumentMap=RepositoryToNumber.NextDocument(RetrievedDocument);
while(RetrievedDocument&&DocumentMap.find("ITEM")!=DocumentMap.end()){
    UniqueID=IDPrefix;
    if(DocumentMap.find("<Accession number:>")!=DocumentMap.end()){
        UniqueID+=trim(DocumentMap["<Accession number:>"]);
    }
    else{
        cout<<"Error: Accession Number Missing."<<DocumentMap["ITEMNUMBER"].c_str()<<endl;
        exit(1);
    }

    if(IDMAP.find(UniqueID)==IDMAP.end()){
        IDMAP[UniqueID]=1;
        CreateRecordTable(DocumentMap,RecordTableOut,error);
        CreateTreatmentTable(DocumentMap,TreatmentTableOut);
        CreateAuthorTable(DocumentMap,AuthorTableOut);
        CreateAuthorSetTable(DocumentMap,AuthorSetTableOut);
        CreateConferenceTable(DocumentMap,ConferenceTableOut);
        CreateINSPECTable(DocumentMap,INSPECTableOut);
        CreateNewRepository(DocumentMap,NewRepositoryOut);
        DocumentMap=RepositoryToNumber.NextDocument(RetrievedDocument);
    }
}
NewRepositoryOut<<"</DOCUMENT>";
RecordTableOut.close();
TreatmentTableOut.close();
AuthorTableOut.close();
INSPECTableOut.close();
NewRepositoryOut.close();
ConferenceTableOut.close();
DocumentMap.clear();
}

void CreateRecordTable(map<string,string> DocumentMap,ofstream& RecordTableOut,ofstream & error){

RecordTableOut<<UniqueID.c_str()<<fieldDelimiter.c_str();
TableFieldPrint("<Data source:>",DocumentMap,RecordTableOut);
TableFieldPrint("<Record type:>",DocumentMap,RecordTableOut);
TableFieldPrint("<Title:>",DocumentMap,RecordTableOut);
if(trim(DocumentMap["<Journal title:>"])==""){
    if(trim(DocumentMap["<Conference Title:>"])!=""){
        TableFieldPrint("<Conference Title:>",DocumentMap,RecordTableOut);
    }
}
else{
    RecordTableOut<<"UNKNOWN"<<fieldDelimiter.c_str();
}
}

```

```

    error<<"Error: No Venue Title: "<<UniqueID.c_str()<<" :
"<<DocumentMap["ITEMNUMBER"].c_str()<<endl;
}
}
else {
    TableFieldPrint("<Journal title:>",DocumentMap,RecordTableOut);
}
TableFieldPrint("<Volume:>",DocumentMap,RecordTableOut);
TableFieldPrint("<Issue:>",DocumentMap,RecordTableOut);
TableFieldPrint("<Inclusive page numbers:>",DocumentMap,RecordTableOut);
PrintDate("<Publication year:>",DocumentMap,RecordTableOut,error);
TableFieldPrint("<Publication date:>",DocumentMap,RecordTableOut);
TableFieldPrint("<Country of publication:>",DocumentMap,RecordTableOut);
TableFieldPrint("<Place of publication:>",DocumentMap,RecordTableOut);
FormReference(DocumentMap,RecordTableOut);
TableFieldPrint("<Publisher:>",DocumentMap,RecordTableOut);
TableFieldPrint("<Number of references:>",DocumentMap,RecordTableOut);
TableFieldPrint("<Language:>",DocumentMap,RecordTableOut);
TableFieldPrint("<Abstract:>",DocumentMap,RecordTableOut);
RecordTableOut<<fieldDelimiter.c_str();
RecordTableOut<<fieldDelimiter.c_str();
RecordTableOut<<rowDelimiter.c_str();
}

void FormReference(map<string,string> DM, ofstream & out){
    string Reference="";
    GetAuthors(DM,Reference);
    AddTitle(DM,Reference);
    AddVenue(DM,Reference);
    AddPublisher(DM,Reference);
    AddVolumeIssueAndPages(DM,Reference);
    AddDate(DM,Reference);
    out<<Reference.c_str()<<fieldDelimiter.c_str();
}

void GetAuthors(map<string,string> DM, string & Reference){
    string temp="";
    temp=trim(DM["<Author(s):>"].c_str());
    if(temp!=""){
        if(temp[0]!='<'){
            istringstream tempstream(temp.c_str());
            temp=GetNextSubTag(tempstream);
            while(temp.size()>0){
                Reference+=temp;
                Reference+=" ";
                temp=GetNextSubTag(tempstream);
            }
        }
        else{
            Reference=temp;
            Reference+=" ";
        }
    }
}

void AddDate(map<string,string> DM, string & Reference){
    if(trim(DM["<Publication year:>"])!=""){

```

```

    Reference+=trim(DM["<Publication year:>"]);
    Reference+=".";
}
}
void AddVolumeIssueAndPages(map<string,string> DM, string & Reference){
    if(trim(DM["<Volume:>"])!=""){
        Reference+="Vol ";
        Reference+=trim(DM["<Volume:>"]);
        Reference+=" ";
    }
    if(trim(DM["<Issue:>"])!=""){
        Reference+="Issue ";
        Reference+=trim(DM["<Issue:>"]);
        Reference+=" ";
    }
    if(trim(DM["<Inclusive page numbers:>"])!=""){
        Reference+=trim(DM["<Inclusive page numbers:>"]);
        Reference+=".";
    }
}

void AddPublisher(map<string,string> DM, string & Reference){
    if(trim(DM["<Publisher:>"])!=""){
        Reference+=trim(DM["<Publisher:>"]);
        Reference+=" ";
    }
}

void AddVenue(map<string,string> DM, string & Reference){
    if(trim(DM["<Journal title:>"])==""){
        if(trim(DM["<Conference Title:>"])!=""){
            Reference+=trim(DM["<Conference Title:>"]);
            Reference+=" ";
        }
        else{}
    }
    else {
        Reference+=trim(DM["<Journal title:>"]);
        Reference+=".";
    }
}

void AddTitle(map<string,string> DM, string & Reference){
    Reference+=trim(DM["<Title:>"]);
    Reference+=".\n";
}

void PrintDate(string field, map<string,string> DM,ofstream& out,ofstream &error){
    string datestring="";

    if(trim(DM[field])!=""){
        datestring+="12:00:00:00 1/1/";
        datestring+=trim(DM[field]);
        out<<datestring.c_str()<<fieldDelimiter.c_str();
    }
    else{
        error<<"Error in item:"<<DM["ITEMNUMBER"].c_str()<<" Pub year missing\n";
    }
}

```

```

    out<<fieldDelimiter.c_str();
}
}

void TableFieldPrint(string field,map<string,string> DM,ofstream& out){
    if(trim(DM[field])!=""){
        out<<trim(DM[field]).c_str();
    }
    out<<fieldDelimiter.c_str();
}

void TableFieldPrintEND(string field,map<string,string> DM,ofstream& out){
    if(trim(DM[field])!=""){
        out<<trim(DM[field]).c_str();
    }
    out<<rowDelimiter.c_str();
}

void CreateTreatmentTable(map<string,string> DocumentMap,ofstream& TreatmentTableOut){
    int x=-1;
    string TreatmentString=trim(DocumentMap["<Treatment:>"]);
    string temp="";
    map<string,int> TreatMap;
    while((x=TreatmentString.find(";"))!=-1&&TreatmentString.size()>0){
        temp=TreatmentString.substr(0,x);
        TreatmentString=TreatmentString.substr(x+1,TreatmentString.size()-1);
        temp=trim(temp);
        TreatmentString=trim(TreatmentString);
        if(temp!=""&&(TreatMap.find(temp)==TreatMap.end())){
            TreatMap[temp]=1;
            TreatmentTableOut<<UniqueID.c_str()<<fieldDelimiter.c_str()
                <<temp.c_str()<<rowDelimiter.c_str();
        }
        temp="";
    }
    if(trim(TreatmentString)!=""&&(TreatMap.find(TreatmentString)==TreatMap.end())){
        TreatmentTableOut<<UniqueID.c_str()<<fieldDelimiter.c_str()
            <<TreatmentString.c_str()<<rowDelimiter.c_str();
    }
    TreatMap.clear();
}

void CreateAuthorSetTable(map<string,string> DocumentMap,ofstream& AuthorTableOut){

    string AuthorString=trim(DocumentMap["<Author(s):>"]);
    istream AuthorStream(AuthorString.c_str());
    map<string,int> AuthorMap;
    vector<string> Authors;
    vector<string>::iterator Begin;

    if(AuthorString!=""){
        if(AuthorString[0]!='<'){
            AuthorString=GetNextSubTag(AuthorStream);

            while(AuthorString.size()>0){
                if(AuthorMap.find(AuthorString)==AuthorMap.end()){

```

```

        AuthorMap[AuthorString]=1;
        Authors.push_back(AuthorString);
    }
    AuthorString=GetNextSubTag(AuthorStream);
}

Begin=Authors.begin();
std::sort(Begin,Authors.end());
AuthorString="";
while(!(Begin==Authors.end())){
    AuthorString+=(*Begin);
    AuthorString+=" : ";
    Begin++;
}
}
}
AuthorTableOut<<UniqueID.c_str()<<fieldDelimiter.c_str()
    <<AuthorString.c_str()
    <<rowDelimiter.c_str();

}
}

void CreateAuthorTable(map<string,string> DocumentMap,ofstream& AuthorTableOut){

string AuthorString=trim(DocumentMap["<Author(s):>"]);
istream AuthorStream(AuthorString.c_str());
map<string,int> AuthorMap;
if(AuthorString!=""){
if(AuthorString[0]!='<'){
    AuthorString=GetNextSubTag(AuthorStream);

while(AuthorString.size()>0){
    if(AuthorMap.find(AuthorString)==AuthorMap.end()){
        AuthorMap[AuthorString]=1;
        AuthorTableOut<<UniqueID.c_str()<<fieldDelimiter.c_str()
            <<AuthorString.c_str()<<fieldDelimiter.c_str()
            <<"Author"<<fieldDelimiter.c_str()
            <<DocumentMap["<Author affiliation:>"].c_str()<<fieldDelimiter.c_str()
            <<fieldDelimiter.c_str()<<(UniqueID+AuthorString).c_str()
            <<rowDelimiter.c_str();
    }
    AuthorString=GetNextSubTag(AuthorStream);
}
}
else{
    AuthorTableOut<<UniqueID.c_str()<<fieldDelimiter.c_str()
        <<AuthorString.c_str()<<fieldDelimiter.c_str()
        <<"Author"<<fieldDelimiter.c_str()
        <<DocumentMap["<Author affiliation:>"].c_str()<<fieldDelimiter.c_str()
        <<fieldDelimiter.c_str()<<(UniqueID+AuthorString).c_str()
        <<rowDelimiter.c_str();
}
}
}
}

```

```

}
}

void CreateINSPECTable(map<string,string> DocumentMap,ofstream& INSPECTableOut){

INSPECTableOut<<UniqueID.c_str()<<fieldDelimiter.c_str();
TableFieldPrint("<Controlled indexing:>",DocumentMap,INSPECTableOut);
TableFieldPrint("<Uncontrolled indexing:>",DocumentMap,INSPECTableOut);
TableFieldPrint("<Classification codes:>",DocumentMap,INSPECTableOut);
TableFieldPrint("<INSPEC Abstract number:>",DocumentMap,INSPECTableOut);
TableFieldPrint("<Accession number:>",DocumentMap,INSPECTableOut);
TableFieldPrint("<Material identity number:>",DocumentMap,INSPECTableOut);
TableFieldPrint("<CODEN:>",DocumentMap,INSPECTableOut);
TableFieldPrint("<SICI:>",DocumentMap,INSPECTableOut);
TableFieldPrintEND("<ISSN:>",DocumentMap,INSPECTableOut);

}

void CreateConferenceTable(map<string,string> DocumentMap,ofstream& ConferenceTableOut){
if((DocumentMap.find("<Conference date:>")!=DocumentMap.end())||(DocumentMap.find("<Conference
location:>")!=DocumentMap.end())){
if((trim(DocumentMap["<Conference location:>"])!="")||(trim(DocumentMap["<Conference date:>"])!="")){
ConferenceTableOut<<UniqueID.c_str()<<fieldDelimiter.c_str();
TableFieldPrint("<Conference date:>",DocumentMap,ConferenceTableOut);
ConferenceTableOut<<fieldDelimiter.c_str();
TableFieldPrintEND("<Conference location:>",DocumentMap,ConferenceTableOut);
}
}
}

void CreateNewRepository(map<string,string> DocumentMap,ofstream& NewRepositoryOut){

NewRepositoryOut<<"\t<ITEM:"<<DocumentMap["ITEMNUMBER"].c_str()<<">\n\n";
NewRepositoryOut<<"\t\t<Unique Storage ID>\n\t\t"<<UniqueID.c_str()<<"\n\t\t</Unique Storage
ID>\n\n";
outputField("<Data source:>", "Data source",NewRepositoryOut,DocumentMap);
outputField("<Record type:>", "Record type",NewRepositoryOut,DocumentMap);
outputField("<INSPEC Abstract number:>", "INSPEC Abstract number",NewRepositoryOut,
DocumentMap);
outputField("<Accession number:>", "Accession number",NewRepositoryOut,DocumentMap);
outputField("<Title:>", "Title",NewRepositoryOut,DocumentMap);
outputField("<Author(s):>", "Author(s)",NewRepositoryOut,DocumentMap);
outputField("<Author affiliation:>", "Author affiliation",NewRepositoryOut,DocumentMap);
outputField("<Journal title:>", "Journal title",NewRepositoryOut,DocumentMap);
outputField("<Conference Title:>", "Conference Title",NewRepositoryOut,DocumentMap);
outputField("<Volume:>", "Volume",NewRepositoryOut,DocumentMap);
outputField("<Issue:>", "Issue",NewRepositoryOut,DocumentMap);
outputField("<Inclusive page numbers:>", "Inclusive page numbers",NewRepositoryOut,DocumentMap);
outputField("<Start Page:>", "Start Page",NewRepositoryOut,DocumentMap);
outputField("<Publication date:>", "Publication date",NewRepositoryOut,DocumentMap);
outputField("<Publication year:>", "Publication year",NewRepositoryOut,DocumentMap);
outputField("<CODEN:>", "CODEN",NewRepositoryOut,DocumentMap);
outputField("<ISSN:>", "ISSN",NewRepositoryOut,DocumentMap);
outputField("<SICI:>", "SICI",NewRepositoryOut,DocumentMap);
CountryFix(NewRepositoryOut,DocumentMap);
outputField("<Place of publication:>", "Place of publication",NewRepositoryOut,DocumentMap);

```

```

outputField("<Material identity number:>", "Material identity number", NewRepositoryOut,
DocumentMap);
outputField("<Publisher:>", "Publisher", NewRepositoryOut, DocumentMap);
outputField("<Conference date:>", "Conference date", NewRepositoryOut, DocumentMap);
outputField("<Conference location:>", "Conference location", NewRepositoryOut, DocumentMap);
outputField("<Number of references:>", "Number of references", NewRepositoryOut, DocumentMap);
outputField("<Language:>", "Language", NewRepositoryOut, DocumentMap);
outputField("<Treatment:>", "Treatment", NewRepositoryOut, DocumentMap);
FixAbstract(NewRepositoryOut, DocumentMap);
// outputField("<Abstract:>", "Abstract", NewRepositoryOut, DocumentMap);
outputField("<Controlled indexing:>", "Controlled indexing", NewRepositoryOut, DocumentMap);
outputField("<Uncontrolled indexing:>", "Uncontrolled indexing", NewRepositoryOut, DocumentMap);
outputField("<Classification codes:>", "Classification codes", NewRepositoryOut, DocumentMap);
NewRepositoryOut<<"\t</ITEM:"<<DocumentMap["ITEMNUMBER"].c_str()<<">\n\n";

}

void FixAbstract(ofstream& out, map<string,string> DM){
string temp="";
int x=0;
if((temp=trim(DM["<Abstract:>"]))!=""){
if((x=temp.find("!\n"))!=-1){
// if(x=986){
temp.erase(x,2);
// }
}
}
out<<"\t<Abstract:>\n\t\t"<<temp.c_str()<<"\n\t</Abstract:>\n\n";
}

void CountryFix(ofstream& out, map<string,string> DM){
string temp="";
if(trim(DM["<Country of publication:>"])!=""){
temp=trim(DM["<Country of publication:>"]);
if(temp=="Can")
temp="Canada";
if(temp=="Engl")
temp="England";
if(temp=="Neth")
temp="Netherlands";
if(temp=="Scotl")
temp="Scotland";
if(temp=="Jpn")
temp="Japan";
}
out<<"\t<Country of publication:>\n\t\t"<<temp.c_str()<<"\n\t</Country of publication:>\n\n";
}

void outputField(string Name1,string Name2,ofstream & output,map<string,string> & DocumentMap){
output<<"\t<"<<Name2.c_str()<<":>\n\t\t";
if(DocumentMap.find(Name1)!=DocumentMap.end()){
output<<DocumentMap[Name1].c_str();
}
output<<"\n\t</"<<Name2.c_str()<<":>\n\n";
}

```

```

string AcceptResponse(){
    char CurrentCharacter;
    string Response;
    cout<<">";
    while((CurrentCharacter=cin.get())!='\n'){
        Response+=CurrentCharacter;
    }
    return Response;
}

bool QueryUser( string& RepositoryName ){
    bool VALIDName=false;
    cout<<"\nPlease Enter The FileName of the\ndesired repository.\n";
    RepositoryName=AcceptResponse();
    ifstream in;
    in.open(RepositoryName.c_str());
    if(in.fail()){
        in.close();
        cout<<"\nInvalid Filename.\n";
    }
    else{
        VALIDName=true;
        in.close();
    }
    return VALIDName;
}

string GetNextSubTag(istream& FieldStream ){
    char CurrentCharacter;
    string SubField;
    SubField="";
    FieldStream.get(CurrentCharacter);
    while(!FieldStream.eof()&&CurrentCharacter!='>'){
        FieldStream.get(CurrentCharacter);
    }
    if(!FieldStream.eof()){
        FieldStream.get(CurrentCharacter);
    }
    while(!FieldStream.eof()&&CurrentCharacter!='<'){
        SubField+=CurrentCharacter;
        FieldStream.get(CurrentCharacter);
    }
    while(!FieldStream.eof()&&CurrentCharacter!='>'){
        FieldStream.get(CurrentCharacter);
    }

    return trim(SubField);}

```