6-2019

Exploring the Dynamics of Scientific Research

Shilpa Lakhanpal
Western Michigan University, shilpa.western@gmail.com

EXPLORING THE DYNAMICS OF SCIENTIFIC RESEARCH

by

Shilpa Lakhanpal

A dissertation submitted to the Graduate College
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
Computer Science
Western Michigan University
June 2019

Doctoral Committee:

Dr. Ajay Gupta, Ph.D., Chair
Dr. Rajeev Agrawal, Ph.D.
Dr. Xiaozhong Liu, Ph.D.
Mr. Edward Eckel, MLIS

ACKNOWLEDGEMENTS

I would like to thank my parents for their constant, unwavering support and love throughout every aspect of my life. I would like to thank my advisor Dr. Ajay Gupta for his superb guidance, exceptional mentoring and admirable patience across the length and breadth of my research. Dr. Rajeev Agrawal consistently provided best advice and immensely useful insights through and through. I would like to thank Mr. Edward Eckel and Dr. Xiaozhong Liu for giving timely and useful advice. I would also like to thank Dr. Donald Nelson and Dr. Steve Carr for their support and endorsement during my work as a graduate assistant in the Department of Computer Science. I would like to extend my heartfelt gratitude to the staff of the Department of Computer Science, especially, Ms. Sheryl Todd for helping me through every stage of my degree program.

I would like to thank the Association for Computing Machinery (ACM) and Institute of Electrical and Electronics Engineers (IEEE) for providing us full access to ACM and IEEE digital libraries respectively.

Shilpa Lakhanpal

EXPLORING THE DYNAMICS OF SCIENTIFIC RESEARCH

Shilpa Lakhanpal, Ph.D.

Western Michigan University, 2019

Scientific research papers present the research endeavors of numerous scientists around the world, and are documented across multitudes of technical conference proceedings, and other such publications. Given the plethora of such research data, if we could automate the extraction of key interesting areas of research, and provide access to this new information, it would make literature searches incredibly easier for researchers. This in turn could be very useful for them in furthering their research agenda. With this goal in mind, we have endeavored to provide such solutions through our research. Specifically, the focus of our research is to design, analyze and implement intelligent machine learning algorithms to extract useful information from research publications, which will be immensely useful to researchers, across a wide spectrum of scientific fields.

In the research arena, various topics are studied, researched and developed across various subject areas, in different scientific fields. Looking for trending topics and according a structure to them, can be especially challenging, given the subjective topic representation by the authors of research papers. These challenges are especially exacerbated by the fact that majority of data in research papers is text, and complete, efficient mining of text data still has many open problems. Our research alleviates some of these challenges and endeavors to make the process of browsing, searching and summarizing the state-of-the-art research innovations across various scientific publications easier, especially to a new entrant into a scientific field.

In order to automate the extraction of useful information, we characterize the data in terms of the type of information or knowledge that we seek from research publications. Specifically in the field of Computer Science publications, we characterize words or phrases from the text to represent topics, specific problem-areas and techniques presented in research papers. We achieve this by investigating features of a word or phrase that make it a potential candidate for specifically representing a topic, by mining information from strategic locations of research papers. We present a methodology to learn the topics representing the current state-of-the-art research in a given time period, within a subject area in a scientific field. We have achieved consistently good results as evidenced by precision and recall results from our model.

In the scientific field of computing, there is an indexing scheme called Association for Computing Machinery Computing Classification System (ACM CCS), which has groups of topics that are used to index research articles in digital libraries. In order to facilitate literature search, we use the topics we have learned and present a technique to generate newer clusters or groups that provide insights into how these learned topics can be incorporated into the existing groups of ACM CCS. We also evaluate how the existing groups may need to be rearranged to reflect the current scenario of research. We have performed exhaustive experiments using the digital libraries of research article publications in the field of Computer Science to illustrate and validate our techniques.

TABLE OF CONTENTS

Table of Contents—Continued

Table of Contents—Continued

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ACM	**A**ssociation for **C**omputing **M**achinery
ACM CCS	**A**ssociation for **C**omputing **M**achinery **C**omputing **C**lassification **S**ystem
CFP	**C**all **F**or **P**apers
IEEE	**I**nstitute of **E**lectrical and **E**lectronics **E**ngineers
LDA	**L**atent **D**irichlet **A**llocation
NLP	**N**atural **L**anguage **P**rocessing
SVM	**S**upport **V**ector **M**achine
Tf-idf	**T**erm frequency - inverse document frequency

CHAPTER 1

INTRODUCTION

The process of analyzing unstructured text data with a goal of deriving meaningful information is termed as text analytics or text mining in common parlance. Text mining is a burgeoning field which involves automating the extraction of knowledge from natural language. Analyzing text data can be facilitated if representative summaries of underlying data were available. To examine such data, we apply techniques from Data Mining, Machine Learning and Natural Language Processing. It is imperative to note that the task of studying the data depends on its context of use. As the extracted knowledge will be used to further an objective, it is best to first identify the key aspect of the data. Thus the kind of guidance sought from financial reports might vary greatly from details extracted from the comments of certain products' users. Thus diversity of the underlying text largely dictates the kind of insights we may seek, which make the exploration even more interesting and challenging.

We narrow our focus into a specific type of information that we may seek from text data found in the research sphere. Scientific research papers published across multitudes of technical conferences, journals, patent-filings, funding-proposals, etc. document the research endeavors of numerous scientists around the world. Naturally, a question arises, whether one can put some structure to this plethora of knowledge and help automate the extraction of key interesting aspects of research. We design, analyze and implement intelligent algorithms and automated tools to help answer various queries commonly occurring during a literature search. Our work will benefit new as well as seasoned researchers in seeking information from a research database. We advance the state of the art by providing intelligence to search.

We begin this chapter by presenting an introduction to data mining. We further present

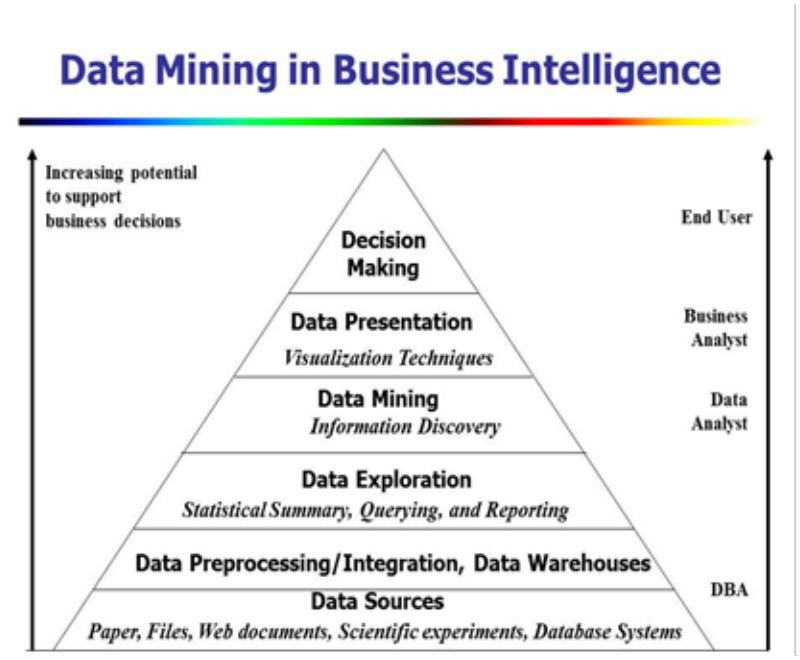
the challenges and motivation behind analyzing text data. We then introduce how our research solves some problems in analyzing text data specific to research sphere. Finally we describe the organization of our dissertation.

1.1. Data Mining

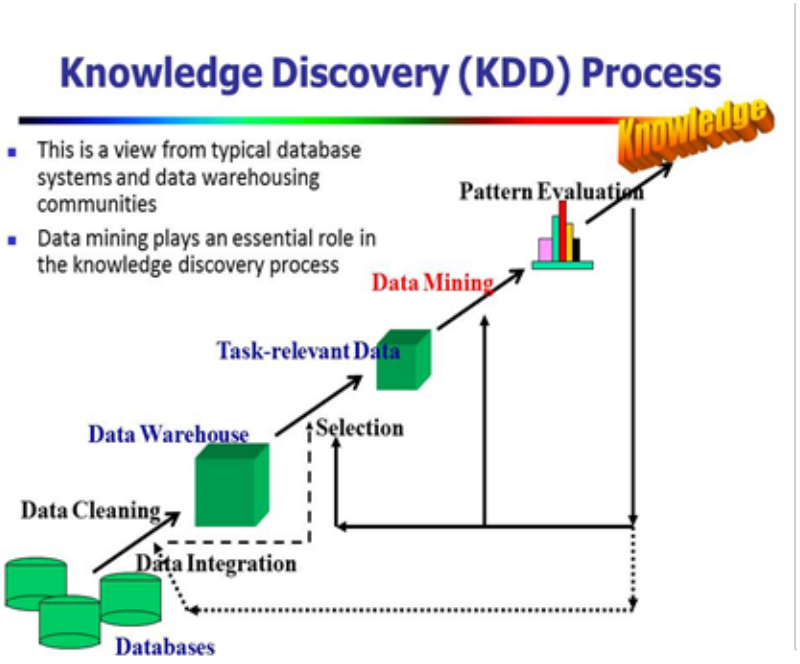
Data mining is the process of analyzing data and extracting useful information from it. This process of knowledge discovery aims at identifying correlations and hidden patterns in massive amounts of data. The information thus derived has tremendous potential to drive decision making strategies. In the business domain, this information can be used to predict future trends and behaviors, allowing companies to make proactive decisions to increase profits, decrease costs or both. Similarly, the advertising industry can use buying patterns to target potential customer segments with better precision. Thus, in practically every field, with the powerful technology of data mining, incoherent data translates itself into unified, coherent, intelligence. Fig. 1.1a and Fig. 1.1b demonstrate the profiles of the people and the steps involved at each stage in the data mining process [1].

The term "Big Data" is used to describe the massive volume of both structured and unstructured data. Big Data can be typically quantified by seven dimensions or the "7 V's namely Variety, Volume, Velocity, Variability, Visualization, Veracity, and Value" [2]. Fig. 1.2 depicts these seven variables [2].

The primary dimension is the data Variety, where data can be broadly grouped into structured and unstructured data. The structured data refers to information that can be easily fit into fields, rows and columns and hence lends itself to seamless inclusion into relational, hierarchical or network databases. This facilitates the tasks of data type definition, data storage, query and analysis. Unstructured data on the other hand, does not have such inherent organization. This type of data mainly consists of text and multimedia content.



(a)



(b)

Figure 1.1: Depicting the data mining process

Figure 1.2: The 7 V's of big data

News articles, web pages, user reviews, emails, business documents, scientific articles, journals, photos, videos, are some of the examples of unstructured data. In order to analyze this type of data, the challenge mainly lies in classifying it into fixed categories, fields, groups or some kind of structured representation. To achieve this, for example in the textual domain, it has been proposed to derive the "content" of data by extracting its "meaning". This is a very hard problem as same text can have different meaning in different contexts. Hence "understanding the text" can provide a very powerful solution toward processing unstructured textual data [3]. Fig. 1.3 compares structured with unstructured data [4].

The Volume of data being generated, analyzed and stored by various businesses is growing significantly. It has been estimated that 80% of all data is unstructured [5]. The all-encompassing term of "digital universe" includes data from over a 100 billion emails exchanged every day, tweets, web articles, research publications, digital movies, security footage, mobile phone messages and many other such sources. The digital universe is expected to grow from 4.4 Zettabytes in 2013 to 44 Zettabytes in 2020, as forecasted in a study by IDC [6]. And about 90% of this data will be unstructured. The sheer volume of big

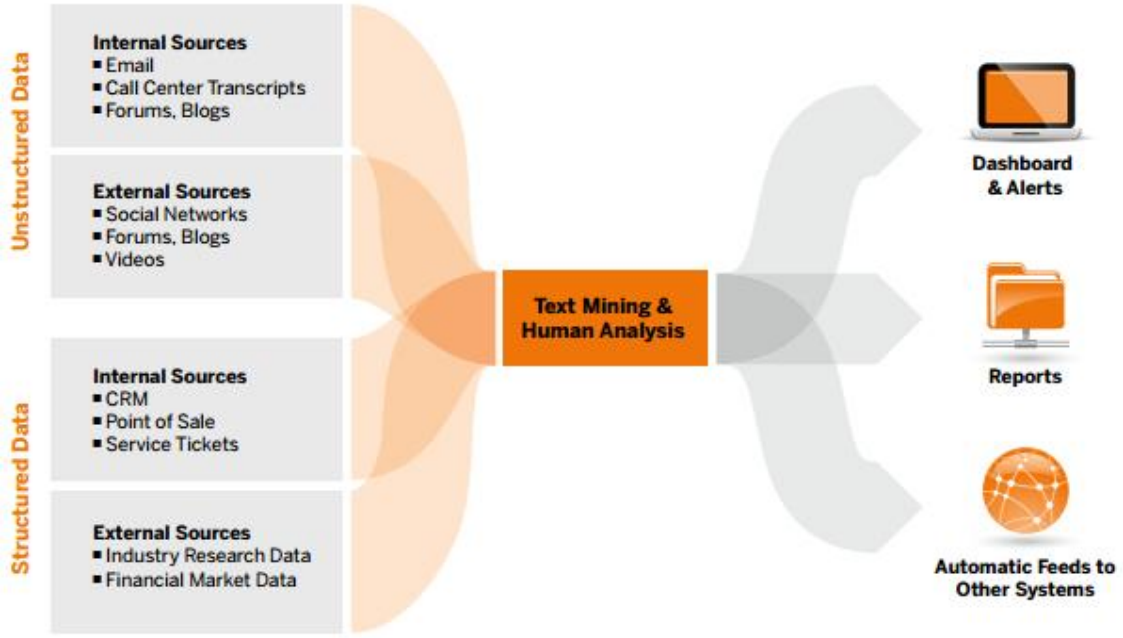


Figure 1.3: Structured vs unstructured data

data presents a major difficulty as traditional techniques are unable to process it. Fig. 1.4 represents the aggressive growth rate of data [6].

Data Velocity characterizes the rate of change within available data, such as when the temporal relationship among two or more data sets changes. It is hence sensitive to frequent bursts of activities, rather than just the ever changing landscape of data [2].

Data Variability refers to data whose meaning is constantly changing [2]. Such data is text data, processing of which particularly involves language processing. The challenge of extracting knowledge from such data mainly lies in the fact that meaning of text changes, based on the context of its usage.

Efficient Data Visualization is one of the major challenges of big data as there are numerous variables and parameters that need to be represented.

Data is not of much use if it is not accurate [2]. Noisy, messy or incomplete data can affect the Veracity of data and hence hinder the decision making process based on it.

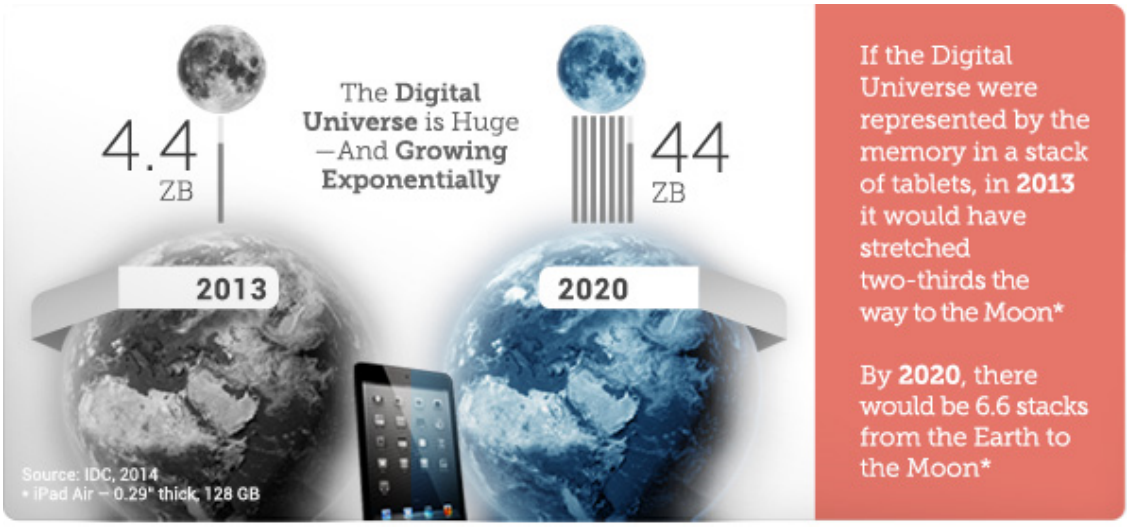


Figure 1.4: Aggressive growth of data

The potential Value of big data is huge, however the cost of poor data analysis is also huge. This begs very efficient analysis [2].

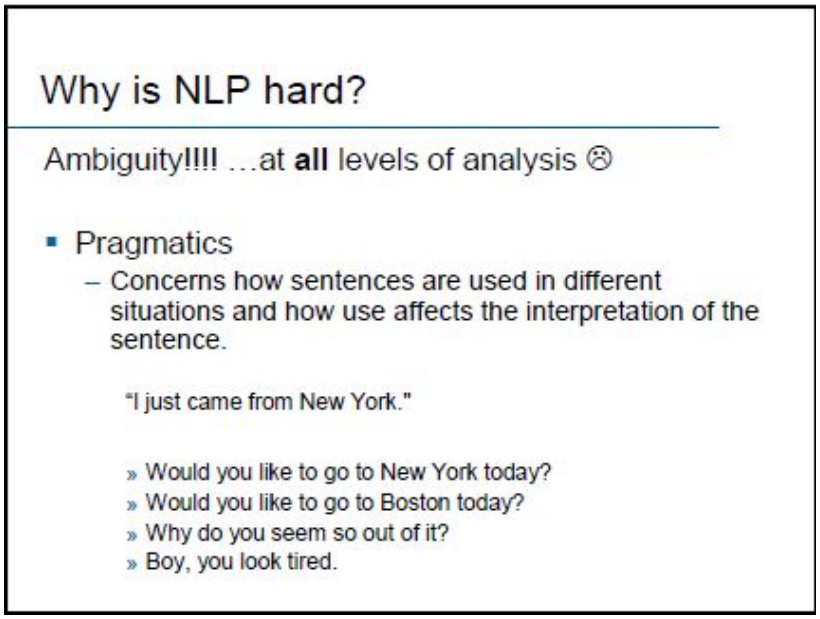
Interestingly, the real challenge in analyzing data is not the volume aspect, as signified by the "big" part of the big data. The big data hype can thus be misleading as it overlooks the core aspect of analytics [5]. This theory has found proponents in the likes of professors and researchers at universities such as Cambridge, Harvard and Northeastern among other digerati [7]. There are three main features that are not just the facets of big data but are characteristic to any type of data [7]. Firstly, the theory or the subject area knowledge is required for tackling data, big or small. Secondly, intrinsic biases in data always exist, which may have the small data sets delineate majority of the features of the larger data supersets. And finally newer unseen patterns that come up with increase in data will also have to be identified by investigating multiple smaller subsets. The essential takeaway is that a problem, however large can be solved by breaking it down into chunks, and solving them first. Hence core solutions toward data analytics will eventually resolve the problem posed by the "massiveness" of big data.

Let us focus our attention to the problem of mining text, which forms a large part of unstructured data. Text mining encompasses techniques for inferring knowledge from text as occurring in myriad contexts. Natural language used by humans to communicate, can convey different meanings in different contexts. Hence automating the gathering of useful information from written narrative, for example, becomes equivalent to understanding the meaning (as humans do), representing this learned information and making educated decisions. Discovering the theme in product reviews, sentiment in user comments, trends from financial blogs, breaking news from newspapers, article summary from magazines, news, and other websites, communication and interaction from social media, research innovations from scientific papers, journals and patents are some examples of the types of information we may seek. Mining such content or theme requires expertise in interrelated subject areas such as machine learning, artificial intelligence, statistics, linguistics and natural language processing.

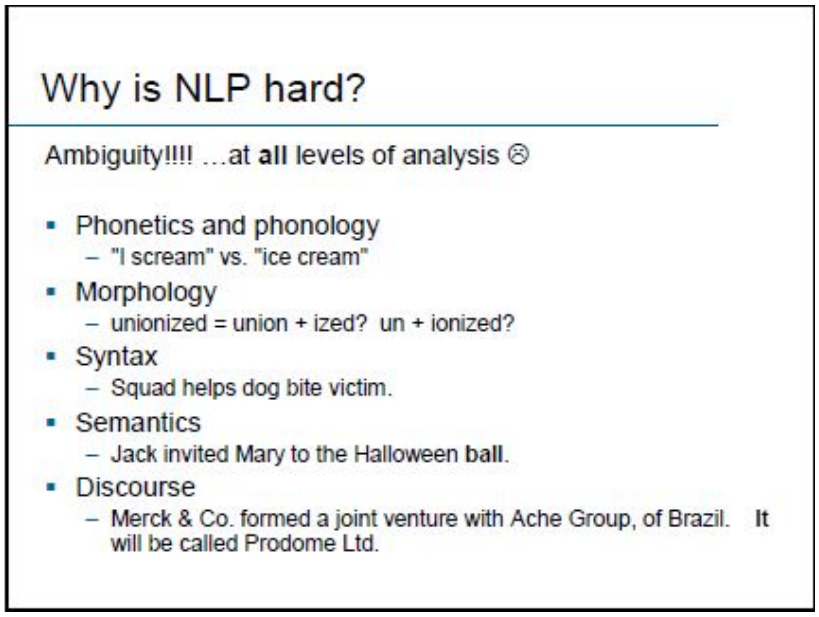
1.2. Natural Language Processing

Natural language processing (NLP) sums up the process of text mining as it aims to automate the understanding of text by computers, in a way analogous to humans. Toward this goal of human to computer translation, analysts encounter ambiguity at all levels. Fig. 1.5a and Fig. 1.5b depict the various dimensions at which such ambiguity occurs [8].

This uncertainty or inexactness occurs because computers do not understand English as a natural language the way humans do. The examples in Fig. 1.6a [9] and Fig. 1.6b suggest that any word, sentence, or a phrase make real sense in the context they are used. Fig. 1.6a [9] depicts how words can have different meanings in various scenarios. Fig. 1.6b further illustrates the possibly conflicting interpretations of the example listed in the syntax category of Fig. 1.5b [8].

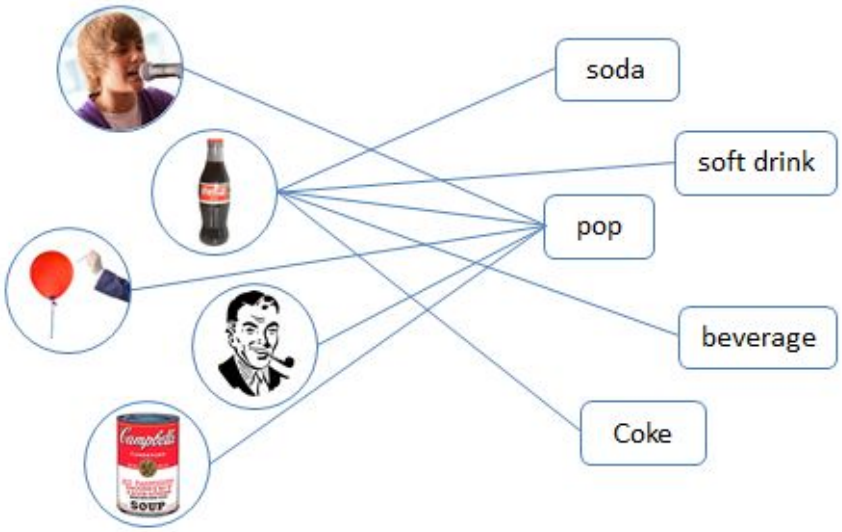


(a)



(b)

Figure 1.5: NLP: complicated process



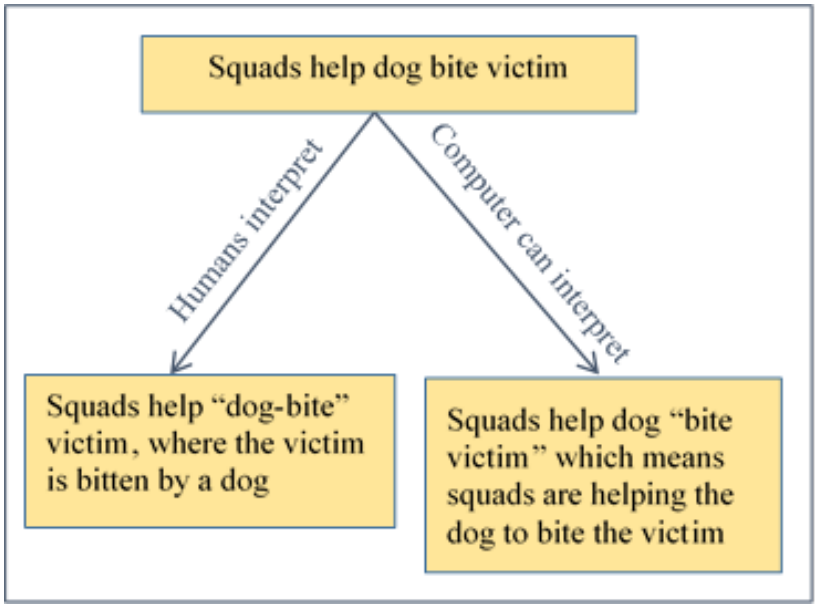
(a) Different contexts: different meanings

Figure 1.6: Challenges in NLP

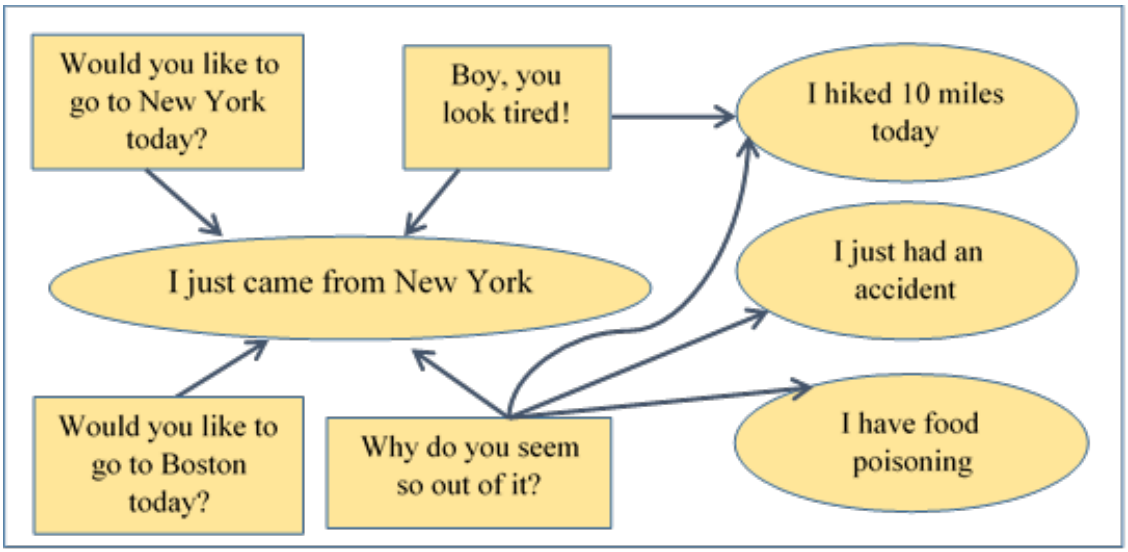
Fig. 1.6c shows examples on how pragmatics or contexts of sentences matter in how they are interpreted. In these examples, depending upon the situation, the same one line can be an answer to multiple different questions. Also various situations can dictate several answers to the same question.

1.3. Where Our Research Comes in

As we have seen above, the challenges in NLP present difficulties in processing text. In our research, we analyze text found in research sphere and focus into extracting specific type of information from such textual data. Scientific research papers published in journals, proceedings of conferences, patents, document the research of numerous scientists from across the world. Finding the state-of-the-art research innovations from this textual data is driven by motivations of *browsing*, *search* and *summarization*. This is especially useful to a new



(b) Different syntactic interpretations



(c) Examples of pragmatic ambiguity in Natural Language Processing

Figure 1.6: Challenges in NLP

entrant into a subject area. A researcher seasoned in one subject area may wish to study another subject area. Hence they may very well be newer entrants into the subject area they are looking to familiarize themselves with. We mostly experiment with research publications in the Computer Science field.

1.4. Our Focus: Enabling Search Related Capabilities in Research Databases

We endeavor to facilitate the process of *browsing*, *search* and *summarization* of the state-of-the-art research innovations in various scientific fields.

1.4.1. Browsing

Students may wish to familiarize themselves with a scientific domain [10]. A research advisor or a professor may want to direct their students to the topics and innovations thereof, of the domain. Such kind of browsing facility will also apprise a new researcher of a path toward a core sub-domain in their choice of a domain area.

1.4.2. Search

A seasoned researcher may want to explore the most recent stage in the development of a subject area [10]. They may want to grasp the most up-to-date features of the subject area, in order to incorporate newest ideas, as they embark into their own research work.

1.4.3. Summarization

Data analysts and research scientists world-wide may want to mine scientific papers with the goal of finding previously undiscovered but potentially very useful correlations, such as the relationship between a drug and an enzyme [11]. An example where such type of pattern

mining has already been incorporated is in the works of pharmaceutical giant, Boehringer Ingelheim [12]. On the lines of summarizing information, a project, called Foresight and Understanding from Scientific Exposition (FUSE), supported by a U.S. intelligence agency is in progress [13]. This project aims to analyze the language of research articles and patents to predict game-changing technologies of the future. Their idea is to extricate the sentiment in the writing in order to forecast the sustenance and potential of a technology [13].

Summarizing text has garnered major attention these days. Whether it is the FUSE project [13] highlighted above, or reducing news items to concise readable synopses [14], summarization in fact mirrors what analysts seek from text, viz. "relevant information". This relevant information is "relevant" according to the context. Therefore, even answering a question such as "Which Google engineering office has the highest average temperature?" entails combining data from webpages listing google offices and historical temperature data [3]. This too serves as an example of summarizing, deriving and presenting relevant information.

Hence we are reasonably and particularly motivated with textual data analysis and propose to focus especially on research papers, journals and articles in the scientific domain. We need to identify what *kinds of information* may be sought, what is the *relevance* of this information, what is the *perspective* through which we look at this knowledge, what *specific problems* need to be addressed, what are the most *effective solutions* and what could be the possible *constraints* and *loopholes*. The rest of the document describes such specific issues and the corresponding approaches that we propose to solving these. Our solutions can obviously be applied to various other problems. Active researchers may want leads from the proposals submitted to funding agencies in order to put forth their own ideas. Records of filed and issued patents can be sampled to reflect the most current studied and investigated technologies in any field. Our solutions can also extend to other domains such as discovering the trends from financial blogs, performing market research at various levels such as a

product's potential for sales, target-market demographics, strategic store locations, etc.

1.5. Organization of our Dissertation

In Chapter 2, we explain how we characterize the data that represents the type of information or knowledge that we seek from scientific publications. Chapter 3 describes our approach to find most frequent research trends using a phrase based technique, which is published in [15], titled "On Discovering Most Frequent Research Trends in a Scientific Discipline using a Text Mining Technique". In Chapter 4, we present our approach to find trending research topics, combining machine learning techniques with results from natural language processing, the complete description of which, along with results and analysis is published in [16], titled "Towards Extracting Domains from Research Publications" and [17], titled "Discover Trending Domains using Fusion of Supervised Machine Learning with Natural Language Processing". Chapter 5 describes our methodology to find similarity between learned trending topics, and using them to improve existing digital indexing schemes, complete with results and analysis, as published in [18], titled "Mining Domain Similarity to Enhance Digital Indexing".

CHAPTER 2

CHARACTERIZING THE DATA

We characterize the data that represents the type of information or knowledge that we seek from technical publications.

2.1. Scientific Field

A scientific field is a systematically organized body of knowledge on a particular discipline. Mathematics, Computer Science, Medical Science are some examples of a scientific field.

2.2. Subject Area

Each scientific field has several broad subject areas which are branches of study within the field. For example, a scientific field such as Computer Science has several subject areas, such as "Networking", "Databases", "Software Engineering", etc.

2.3. Topic, Domain, or Topical Domain

Each subject area contains topics which define that area. These topics are the domains of the subject area within the scientific field. Each subject area can have many domains within it. For example, the subject area "Networking" has domains such as "Wireless Networks", "Ad-Hoc Networks", "Network Architecture", etc. Each such domain can have several sub-domains, thus building up a hierarchy. Irrespective of the hierarchical structure, domains or their sub-domains represent the important topics that are studied, researched and developed in a subject area in a scientific field. As of now, no clear well-accepted hierarchy

of domains and sub-domains exists. Hence we do not differentiate between a domain and a sub-domain. In our discussion, we will use the the terms “topic”, “domain”, or “topical domain” interchangeably as they refer to the same concept.

2.4. Problem-Area

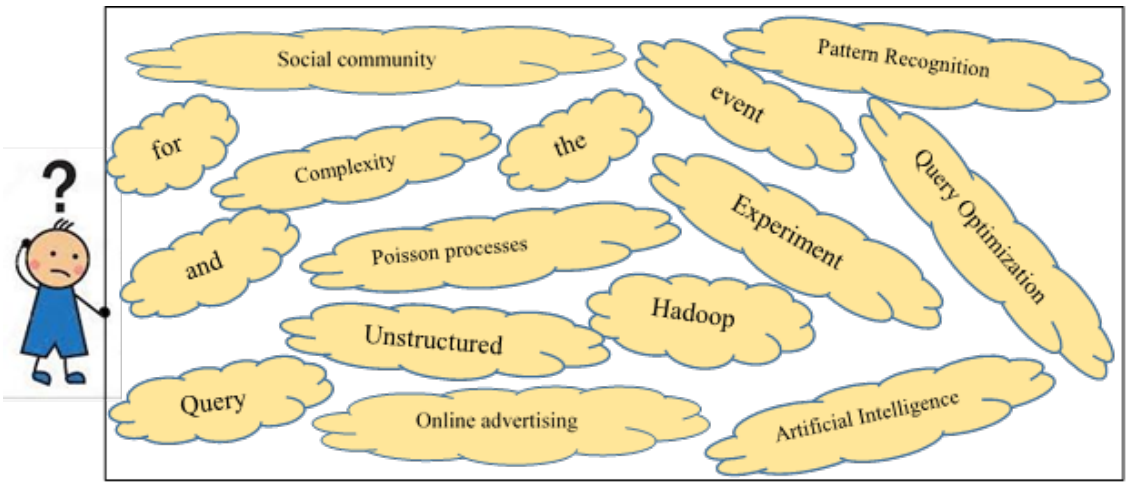
The problem-area addressed in a paper is the focus of research described in that paper. Each research paper or a journal is written to demonstrate the work done by the authors to solve a particular problem, or to achieve a goal. Survey papers are exceptions as they illustrate work done by other researchers. Therefore, the goal or research focus of a paper constitutes its problem-area.

2.5. Technique

For solving a problem, the researchers apply techniques, or may even devise their own techniques.

2.6. Characterizing the Data within Research Papers

The research in each research paper, focuses on, draws ideas and techniques from several domains. These domains are really the broad topics or purpose of the research described in the research paper. The research paper presents techniques and research to address a specific problem-area within broader domains. This problem-area represents the specific focus of research of the research paper. A domain can also be seen as the common topic which runs across several problem-areas discussed across several papers. At the same time problem-areas could be viewed as sub-domains of the parent domain, several levels down in the hierarchy, depending on their specificity. It is interesting to note that a problem-area



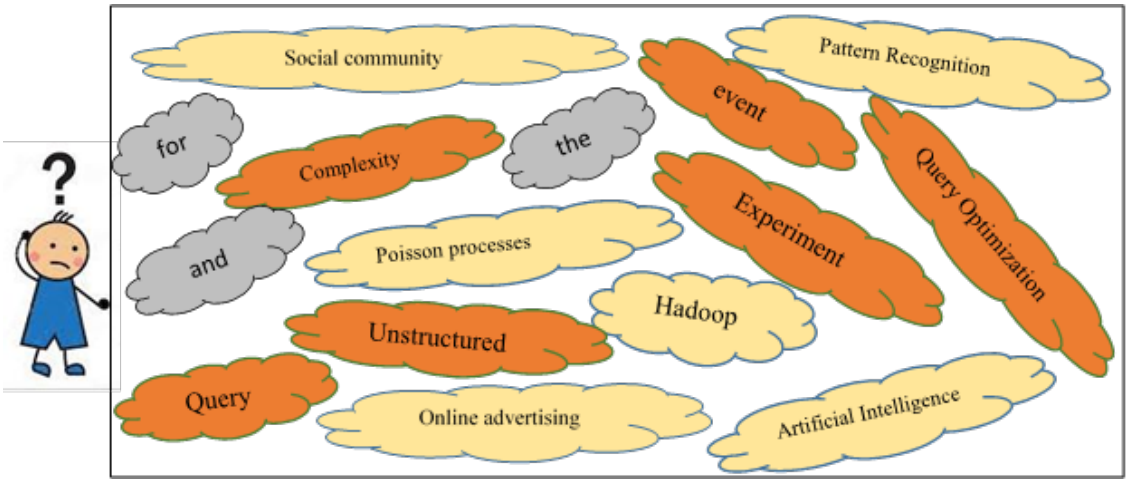
(a) Which of these words or phrases are topical domains?

Figure 2.1: Identifying topical domains

that was initially the focus of a small amount of research may gain a lot of interest over time. As more research becomes focused on it, it starts to generate several smaller problem-areas. Hence the original problem-area now becomes a domain. Hence for the purpose of this document, we do not differentiate between a domain and a problem-area, and refer to both of them as domains.

Any given research paper is basically just a collection of words. When we read the paper, we might be able to decipher what domains it caters to. But this ability to comprehend these topics could be based on our prior knowledge of what constitutes domains. We will certainly be in error if we presumptuously assume that any and every reader will be pre-equipped with the correct understanding of whether a word or a phrase is a domain, or a technique. After all, a new researcher might be totally clueless about the existence of certain topical domains altogether. Fig. 2.1a and Fig. 2.1b illustrate this state of quandary, when sampling papers in the field of Data Mining. Fig. 2.1a presents some common words or phrases obtained from these papers.

Fig. 2.1b illustrates that a new researcher might fairly distinguish that words such as



(b) Still confused about the domains

Figure 2.1: Identifying topical domains

"for", "and" and "the" are prepositions and conjunctions, and are there to add to the semantic meaning of a sentence. Hence, these words are "greyed out" to depict removal from further consideration. Further they may be able to recognize that words such as "complexity", "event", "experiment", "query" and "unstructured" may by themselves not be topics, or techniques. Therefore these words are given a different (dark orange) color in the figure to portray this. Note that we say they may be able to make that distinction, but it could not always be the case as it depends on their knowledge of the subject area. But even then they may very well miss out on "Query Optimization" as a potential topic, as they might misconstrue the word "Optimization" as an additional descriptive word to the already unsure word, "query". But even their educated guesses of the remaining words or phrases may not conclusively provide them with the distinction of domain areas vs techniques. As can be seen from Fig. 2.1b, "Hadoop" and "Poisson processes" may be perceived as domains but they really are implementation frameworks / methodology and technique, respectively.

This problem leads us to further question whether there exists a pre-defined dictionary of any scientific field's topical domains, to begin with. Would such a dictionary correctly label

each research paper according to its topic? The hypothesis that such a repository exists, will make certain solutions possible. We investigate the scope of these solutions in Chapter 4.

But even if such a dictionary existed, the ever evolving nature of science would command a continuous updating of this dictionary. We can expect that continuous additions would take place in all scientific subject areas, due to the incessant advances being made in those areas.

Each domain might have intensive level of research going on in terms of the problem-areas being worked on. Also varied techniques may be used for same problem-areas.

Hence we investigate the features of a word or phrase which make it a potential candidate for a topical domain, or technique. Some of these features are briefly outlined below:

2.6.1. Location

The placement of a word or phrase in the title, abstract, or conclusion, emphasizes that the said item is important. The authors may want to highlight the topical domains, and techniques and hence place them in these major strategic areas of the paper [10] [19] [20].

2.6.2. Frequent Occurrences at These Locations

Are there certain words or phrases which occur at all or majority of the above locations? Would these be more likely to be topics?

2.6.3. Occurrence of Words or Phrases after Certain Prepositions

Are there certain words or phrases that appear after certain prepositions? Can we learn anything about these words or phrases based on the meaning conveyed by the prepositions?

2.6.4. Meaning Conveyed by Words or Phrases

Is there a way we can decipher the meaning of words, when they are standalone, or parts of phrases? Would this meaning be mere dictionary meaning?

2.6.5. Presence of Words or Phrases in Well-accepted Repositories

Are there any well-accepted repositories which contain most researched topical domains? Would these help in processing the topical domains we extract in research papers?

We begin our research by extracting words or phrases representative of topical domains or techniques. We then proceed to identify whether certain characteristics of a word or phrase can help us determine whether they are more likely to represent topical domains. Hence we go on to extract such topical domains. Further we process these topical domains obtained thus, and analyze how they can contribute to reflect the changing state of the art of research.

CHAPTER 3

DISCOVERING FREQUENT RESEARCH TRENDS USING A PHRASE BASED APPROACH

If we consider articles that carry factual contents such as news, scientific research papers or journals, organization or company reports; we observe a common inclination of the writers. This inclination leans towards conveying the gist of the content through the title of the article. Our observation is supported by research scientists and experts from across the scientific to journalistic domain, who repeatedly emphasize that the title reflects the salient points of any article [10] [19] [20] [21]. For a scientific paper, in addition to the title, the abstract captures the essence, approach or goal of the paper [20] [21] [22].

The lead that title and abstract in fact aim to convey the main gist of a scientific research paper, lends itself to a conclusion that title and abstract actually try to summarize the most important ideas of the paper. The authors of a research paper tend to highlight the key items of their research in the title and abstract. Hence the title and abstract of a research paper encompass within their component words or phrases, the core topic, aim, technique, or methodology of that paper. In this chapter, we extract from the titles and abstracts of each research paper, the words or phrases that represent the topic, problem-area, or technique. Though we do not differentiate between a topic or a technique, nevertheless, the resulting words or phrases are representative of research discussed in the research paper. When extracted from a collection of research papers, such words or phrases depict the research trends across the collection. Armed with this information, we use the databases from across various conferences, from primary organizations that promote academic and scholarly interests in the scientific or computing field. Specifically we look at the titles and abstracts of scientific

papers from some of the conference proceedings. We use a rather simple but highly intuitive technique to analyze these titles and abstracts in our preliminary exploration.

3.1. Related Work

Mining text is an active field of research. There are several ways in which information can be retrieved from large amount of textual data. Document summarization is one such methodology. Few of its applications are summarizing news pieces [14], and drawing a summary from multiple documents [23] [24]. Unlike these, in our approach toward trends extraction from a plethora of articles, we do not have to look at the entire text, rather just the title and/or abstract of an article. We use the core idea that the latter two fields already summarize the content of an article. Authors in [23] use a statistical approach to predict sentences that contribute to the document's summary. We use a frequency counting technique, which does not rely on a probabilistic model. Another challenging aspect of text mining is document topic modeling. Statistical techniques are commonly used to develop models in order to discover the theme of a document [25] [26] [27] [28]. Latent Dirichlet Allocation (LDA) described in [25] [26] is a statistical topic model that discovers the hidden theme or topics from a collection of documents. Assuming an imaginary generative process of constructing these documents, LDA then tries to backtrack from these documents to infer a set of topics that are likely to have generated the collection [29]. While being a powerful tool at discovering the thematic structure of text, LDA makes certain assumptions, which make it inappropriate when considering the semantics of a language. One such assumption is that a document is a "bag of words", where the order of words does not matter [25]. We extract phrases from a sentence, which convey some meaning by themselves. The idea of keyword extraction has been used in [30], where most frequent single words in a text are considered to be conveying the inherent idea of a text. However, the authors do not present

a well-formed approach of combining similar or related words together as they do not explain the recreation of multiple word phrases after already having segmented the text into single words. Phrases and sub-phrases have been extracted to decipher only the most frequent keywords in [31], but the authors seem to have erroneously evaluated the performance of their approach as linear rather than quadratic. Moreover, although they describe a valid approach, their algorithm has mistakes, and does not correctly implement this approach. We provide a sound algorithm, which not only calculates the frequency of phrases, but also allows one to easily cluster the related documents together.

3.2. Definitions

As we foray into the description of our technique, we want to emphasize that in our analysis, we are dealing with the constructs or elements of the English language. From the linguistic aspect, we should note that the title is a sentence, which is a grammatical unit of one or more words that expresses an independent statement. We further define some more components of the language to which we shall frequently refer to, in the course of our discussion.

3.2.1. Word

A single and distinct element of language which has a meaning and is used with other words to form a sentence, clause or phrase

3.2.2. Stopword

Word in the language, such as “and”, “the”, which is very common, but is not very useful when selecting text that answers a user’s query

3.2.3. Sentence

A sequence of words that is complete in itself, containing a subject and predicate, conveying a statement, question, exclamation, or command, and consisting of a main clause and, optionally, one or more subordinate clauses

3.2.4. Clause

A unit of grammatical organization next below the sentence in rank and in traditional grammar said to consist of a subject and predicate

3.2.5. Phrase (Ph)

A small group of words standing together as a conceptual unit, typically forming a component of a clause

3.2.6. m-gram

A contiguous sequence of m words in a given sequence of words

3.2.7. Sub-phrase (SPh)

An m-gram substring of a phrase Ph, that keeps the left to right continuous order of words intact

3.3. Phrase Extraction

Grammatically, the title of a paper could be a sentence, clause or phrase. A title is first mined to extract its constituent phrases, which would be enclosed between or delimited by

well-defined stopwords. The abstract is processed in the same way. By counting the frequency of phrases across the collection of research papers, it would be possible to generate the most frequently occurring phrases, and hence the most frequent trend in current research. For example, if the phrase "text mining" occurs most often, that means current research in data mining is focused on mining text.

The authors in [31], extract phrases and sub-phrases to decipher the most frequent keywords from research papers but they seem to have erroneously evaluated the performance of their approach as linear rather than quadratic. Moreover, despite having a valid approach, their algorithm has mistakes, and does not correctly implement the approach. We provide a correct algorithm, and also offer a correct performance analysis.

Let RP_i denote the i^{th} research paper and Ph_j denote the j^{th} phrase in this paper. This phrase will be delimited by stopwords and punctuations. Thus, along the same lines as [31] we represent:

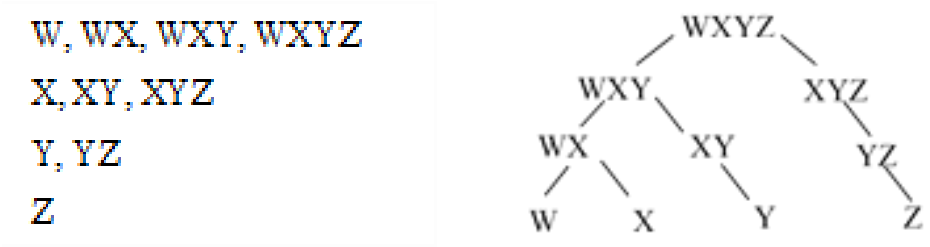
$RP_1 = [Ph_1, Ph_2, Ph_3, ...]; RP_2 = [Ph_2, Ph_4, Ph_5, ...]; ...$

Our technique elicits such phrases from each title and abstract. Obviously a phrase may occur in many RP_i's, e.g. notice that Ph_2 occurs in RP_1 and RP_2 in the above example. Different papers may not use the exactly same whole phrase but a part of the phrase; hence we need to extract sub-phrases from the phrases.

In order to extract sub-phrases from each phrase, we build them from left to right keeping the sequence of words fixed. Since most phrases and subsequently their sub-phrases will be common across different research papers, we reverse the above representation. Let SPh_j denote the j^{th} sub-phrase in phrase Ph_i, and RP_k denote the k^{th} paper containing thej^{th} sub-phrase. The new representation looks like:

$Ph_1 = [SPh_1, SPh_2, SPh_3, ...]; Ph_2 = [SPh_1, SPh_4, SPh_3, ...]; ...$

$SPh_1 \in [RP_1, RP_2, ...]; SPh_2 \in [RP_1, ...]; SPh_3 \in [RP_1, RP_2, ...]; ...$



(a) (b)

Figure 3.1: Sub-phrases of the phrase "WXYZ"

3.4. Sub-phrase Extraction

In order to maintain the semantic meaning of a phrase, we are only interested in sub-phrases of each phrase. The sub-phrases keep the sequence of words from the phrase intact. This allows us to optimize the extraction process as this effectively reduces from considering the 2^n possible substrings of a phrase to n(n+1)/2 = O(n^2), where n is the number of words in a phrase. To illustrate this, if "WXYZ" is a phrase of length 4 in a sentence, then left-right extraction of the sub-phrases results in the sub-phrases as outlined in Fig. 3.1a.

A similar technique of sub-phrase extraction has been proposed earlier in [31], where top-down disassociation of a phrase was done as demonstrated in the Fig. 3.1b. However, the authors claim to extract all the above possible n^2 sub-phrases from an n-word phrase in O(n) time. It seems to be an incorrect claim. They have explicitly addressed each of the sub-phrase in their technique. And in order to explicitly reference each such sub-phrase, it takes O(n^2) time, which is erroneously claimed to be linear time.

3.5. Our Technique

We have programmed our technique in Java and R. An important preprocessing step is the stemming of the words in the titles and abstracts. Stemming reduces each word to

```
1: procedure PHRASE_FREQUENCY(RP)
2:     for all sentences in the title and abstract of each RP_k do
3:         for all phrases Ph_i in RP_k do
4:             for all sub − phrases SPh_j of Ph_i do
5:                 SPh[SPh_j] += RP_k
6:             end for
7:         end for
8:     end for
9:     for all sub − phrases SPh_j in SPh do
10:        list_j = SPh[SPh_j]
11:        SPhCount[SPh_j] = number_of_items in list_j
12:    end for
13:    sort SPhCount[SPh_j] in descending order by value
14: end procedure
```

Figure 3.2: Algorithm PHRASE_FREQUENCY.

its root. Suppose we look at a collection of data mining research papers and we find 25 papers having the sub-phrase "association rule mining" and 2 papers having the sub-phrase "association rules mining". We would want to reflect "association rule mining" trending from 27 papers because the above two phrases essentially convey the same meaning. This makes sense when we want to explore trends in the field of data mining.

The sub-phrases are extracted from each phrase Ph_i, where each subsequent sub-phrase is added as a key to a hashmap, SPh. $SPh[sub]$ denotes the list of research papers containing the subphrase, sub. Finally, the hashmap, $SPhCount[sub]$ contains the number of research papers corresponding to $SPh[sub]$. Sorting $SPhCount[sub]$ in descending order by value yields the desired result, that is the most recurrent sub-phrases across all papers. The algorithm, PHRASE_FREQUENCY in Fig. 3.2 encompasses the main steps of our technique.

Our technique is rather simple, yet highly intuitive, effective and inherently powerful in reflecting the most recent work or the most researched techniques in a domain. For example, a researcher looking at thousands of papers in data mining might want to know what people have been working on the most, in past 5 years. Our technique can give an answer to this question. Suppose the result to the above query is "text mining". Our technique gives this

Table 3.1: Most frequent sub-phrases of the papers presented at the ACM/IEEE Supercomputing Conference from 1988-2013.

m-gram	**Subphrase-length**	**Subphrase**
unigram	1	parallel, performance, computing, systems, simulation, applications
bigram	2	high performance, parallel computing, parallel programming, massively parallel
trigram	3	high performance computing, parallel file systems, molecular dynamic simulation, massively parallel computing
four-gram	4	high performance computing systems, interactive parallel programming tool

result along with the list of respective papers, where "text mining" is addressed.

Given L sentences, each containing at most P phrases with maximum n words in each phrase, and at most SPh_{max} distinct sub-phrases across all the L sentences, the time complexity of the PHRASE_FREQUENCY algorithm is O($(L * P * n^2) + SPh_{max} * log(SPh_{max})$). A crucial observation is that given its semantics, each sentence can only have a certain number of phrases, and there is a small upper-bound on this number. The same rationale is true for the sub-phrases of P. Hence $P * n^2$ is much smaller than L. Similarly SPh_{max} is much smaller than L. Thus, the time complexity proves to be almost linear with respect to L.

3.6. Experimental Results

We have conducted many experiments using the database of research papers from various conferences and journals. For example, using only 1781 papers from the ACM/IEEE Supercomputing Conference (SC) from 1988 to 2013, Table 3.1 shows the most frequent m-grams or sub-phrases of length 1 to 4.

As can be readily seen, unigrams do not make much sense by themselves, and the reader

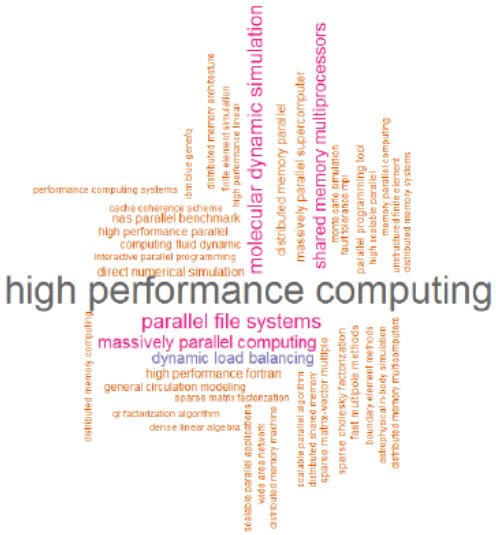
(a) Bigrams word cloud from SC88 – SC13

Figure 3.3: Research trends in Supercomputing subject area, 1988 - 2013

can be left guessing the context in which each single word is used. It can be inferred that the bigrams begin to make sense. For example, the frequent occurrence of the bigram, "high performance" indicated that more research is being concentrated on "high performance computing". This intuition is validated by the frequent trigram, namely, "high performance computing". As is also evident from above, we have found that m-grams with m >= 2 give more meaningful results, and the bigram or trigram appear to give the most coherent trending results. However, further investigation is needed to design adaptive techniques that can "quickly" identify appropriate values of m given a context, rather than weeding through all the values.

The word clouds in Fig. 3.3a and Fig. 3.3b represent the bigrams and trigrams with respect to their frequencies. We can definitely see the trending research areas in the supercomputing subject area.

To further corroborate the initial findings, as another example, we used our technique on the collection of 3068 research papers obtained from the IEEE International Conference



(b) Trigrams word cloud from SC88 – SC13

Figure 3.3: Research trends in Supercomputing subject area, 1988 - 2013

Table 3.2: Most frequent sub-phrases of the papers from the IEEE ICDM (2001 to 2013) and ACM SIGKDD (1999-2013)

m-gram	**Subphrase-length**	**Subphrase**
unigram	1	mining, data, clustering, learning, model, patterns,...
bigram	2	data mining, time series, association rules, social networks, data streams, feature selection, support vector, text classification,...
trigram	3	support vector machines, association rules mining, high dimensional data,...

on Data Mining (ICDM) from 2001-2013 and the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD) from 1999-2013. Most frequent occurring m-grams in these papers are illustrated in Table 3.2.

Since the papers are retrieved from a data mining conference, “data” and “mining” as keywords are expected and superfluous to our goal of finding the trending research areas within “data mining”. The unigram, “model”, although interesting is meaningless without context. As before, the bigrams such as “association rules”, “social networks”, “support

Figure 3.4: Word cloud from ICDM (2001-2013) and KDD (1999-2013)

vector", etc. provide insight to the techniques used in data mining. This is further confirmed by their recurrence as a part of the most frequently occurring trigrams.

The word cloud in Fig. 3.4 presents the bigrams resulting from this experiment.

3.7. Conclusions and Future Work

We have presented a technique to discover current research trends in a subject area. The technique yields encouraging results. We have published the motivation, technique and results as described in this chapter in [15]. This methodology can be improved further by incorporating permutations of related phrases. For example, "research paper recommendation" and "recommending research papers" should be treated as the same phrase. Frequently used synonyms can also be considered when deciding the similarity of two phrases. For example, "method" and "technique" render the same meaning. Another focus area is to extract the sentence structure, such as subject, or object from the title. The position of prepositions can be also useful in evaluating this. As an example, the title, "Leveraging Sentiment Analysis

for Topic Detection" [32] can tell us "what" is used for "Topic Detection" or "Sentiment Analysis" is used for "what". One such problem-area, is what we go on to address in the next chapter.

CHAPTER 4

DISCOVER TRENDING DOMAINS USING FUSION OF SUPERVISED MACHINE LEARNING WITH NATURAL LANGUAGE PROCESSING

Semantics is a branch of linguistics that deals with the meaning of words and phrases in a particular context. A word is a single and distinct element of language which has a meaning and is used with other words to form a sentence. A phrase is a small group of words, typically having a meaning as a conceptual unit. A sentence is a group of words, complete in itself, conveying a meaning in the form of a statement, question or exclamation. A preposition is a word which acts as a connector between words and/or phrases in a sentence, thus adding to the semantic meaning of sense, reference or other such logic to the sentence. For automating the understanding of language, one of the steps is to elicit the semantic meaning of each sentence. And towards achieving this purpose, we need to understand how and in what context, the prepositions are used.

In our previous chapter, we have talked about the area we liked to focus on next, which is to extract useful ideas by taking into account the sentence structure. We have also planned to treat a phrase and its permutations as same, when looking at the phrases within a set of frequent ones. With these ideas in mind, and focusing on the utility of prepositions within each sentence, we introduce the idea of preposition disambiguation.

Our technique, as described in this chapter, extracts theme from each research paper in the form of its domain. We derive interesting phrases based on their placement in the vicinity of certain prepositions by using results of preposition disambiguation. Even though a research paper has structure in terms of its division into sections such as abstract, introductions, etc., still the text in these sections is just a bag of words for a computer. Hence we train a

computer algorithm to classify the interesting phrases as to whether they are domains or not. Therefore phrases are accorded a meaning and this meaning is derived exactly as the respective authors themselves wished to convey. Besides the quality of fusing knowledge from NLP and supervised learning, our technique effectively derives meaning of text without explicitly using the constructs of NLP.

4.1. Related Work

Analyzing the focus of research by extracting information from research database is becoming an active field. Techniques from NLP domain have been employed toward this goal. A bootstrapping learning technique has been proposed in [33] to extract items such as domain areas, focus of research and techniques from research papers. Using dependency trees and starting with some handwritten semantic patterns in three categories of domains, focus, and techniques, their methodology learns new patterns. Although the work provides key insights, their results are not that encouraging as they themselves claim that their system failed to correctly address patterns which it found to be outside their three pre-defined categories [33]. Analysis of their results indicates that their technique for domain extraction has high recall but suffers from low precision [33]. This indicates that although they are able to retrieve domains, they also incorrectly mark non-domains as domain areas. Our approach does not explicitly use NLP per se but fuses NLP and supervised learning to obtain good results of high precision and high recall for labeling domains.

Supervised learning for text classification has been widely used in applications of NLP. Hidden Markov Models (HMMs) are statistical tools for modeling generative sequences that can be characterized by an underlying process generating an observable sequence [34]. In NLP, they are used to mark the part-of-speech category of various words in text. The HMM model is a stochastic analog of finite state automaton, with probabilistic transitions

between states. HMMs have been used for sentence classification [35], where the preferred sequential ordering of sentences in the abstracts of "Randomized Clinical Trial" papers, facilitated its use. The sentences in the abstract are supposed to be ordered in sequence of "background", "objective", "method", "result" and "conclusion" [35] and model-states are aligned to these sentence types. Our approach does not depend on a generative process as the "domain", "problem-area" and "technique" can occur in any random order in a title. Hence our approach targets more generic solutions.

In our previous work [15], we extracted the prevalent trends of research using a phrase-based approach. We created a simple but intuitive technique to analyze the titles of a collection of research papers. A title was first mined to extract its constituent phrases, which were enclosed between or delimited by well-defined stopwords. By counting the frequency of phrases across the collection of research papers, it was possible to generate the most frequently occurring phrases, and hence the most frequent trend in prevalent research. The titles tend to be unique, and hence the ordered sequential left to right structure of phrases may be restrictive as we did not account for the permutations. In this paper, we take our work much further by incorporating a fusion of NLP with intelligent machine learning techniques to extract meaningful domain areas from research papers.

4.2. Definitions

We have listed some of the relevant constructs of English language in Chapter 3 Section 3.2 as they are used in describing our techniques. Here we define some more important concepts as they shall be used for discussion.

4.2.1. Preposition

A word governing, and usually preceding, a noun or pronoun and expressing a relation to another word or element in the clause

4.2.2. Preposition with Intention Sense

The preposition that indicates that the phrase following it specifies the purpose (i.e., a result that is desired, intention or reason for existence) of an event or action

4.2.3. Phrase of Interest (Interesting Phrase)

A phrase that follows a preposition with intention sense and ends before the next preposition in the clause or ends with the end of the clause

4.2.4. Derivative

Keyword or keyword phrase which has one or more words in common with an interesting phrase

4.2.5. Domain Word

A word that denotes or has a potential for naming a well-accepted domain area, or is a part of a phrase denoting a well-accepted domain area

4.3. Preposition Sense Disambiguation

A preposition as defined above expresses a relation between two elements of a clause. One relation can be conveyed by different prepositions depending on the context in which they are

used. Conversely one preposition can convey different meanings. The position of prepositions in text and their contextual use can provide extremely useful insight into the meaning of text. Much work has been dedicated to extricate the "sense" or the "relation" conveyed by the presence of various prepositions within different group of words [36] [37]. We would like to explain the meaning of intention. For example the intention sense is communicated by the preposition "for" in the phrase "system for extracting data". According to the work in [36], the "complement" of the preposition conveys the "intention" or "purpose". In the English language the complement generally refers to a noun phrase, pronoun, a verb, or adverb phrase [38]. Another term used to denote the "complement" is called the "object" of the preposition as used in [37], who have identified an inventory which presents 32 different meanings, built on the "relations" established by the usage of prepositions in various settings. It may be noted that the 7 different senses [36] seem to encompass the 32 relations elicited by authors in [37]. Hence we chose to work with the senses of the prepositions. Authors of a technical paper may want to communicate the crux of their paper through their titles [20] most likely by using technical terminology while paying less attention to nuances of English language such as adverbs or pronouns [38]. Hence, for simplicity we pick the complement that will be delimited at the other end by the next preposition or end of the clause and define it as an "interesting phrase".

We have compiled a complete list of prepositions after reviewing several English handbooks. Careful study of the preposition senses narrowed down in [36] has allowed us to create our set of prepositions with intention sense, $\boldsymbol{PI}$ as depicted here:

$\boldsymbol{PI} = [\text{“}for\text{”}, \text{“}to\text{”}, \text{“}towards\text{”}, \text{“}toward\text{”}]$

We denote each preposition in this set as $\boldsymbol{p_i}$. We denote all other prepositions as $\boldsymbol{p_o}$.

The complement, $\boldsymbol{C}$ is a phrase that is extracted based on the permutations of $\boldsymbol{p_i}$ and $\boldsymbol{p_o}$ in a clause. $\boldsymbol{E}$ denotes the end of the clause. The following depicts the relevant permutations

and the corresponding complement:

$\boldsymbol{p_i C p_o}$

$\boldsymbol{p_i C p_i}$

$\boldsymbol{p_i C E}$

This complement, C, is the interesting phrase. It should be assumed that there is a space between each two consecutive words, even though these spaces are not explicitly presented in the above representation.

4.4. Fusion of Title and Keywords

We start with the title of a research paper as the authors would probably want to highlight the goal of their research in their title [19] [20] [21]. In order to relay their goal in as succinct form as possible yet making it comprehensive enough, they might include the underlying theme or main topic or the domain of their research. Since interesting phrases by their very definitions reflect the "purpose" or the "goal" in their respective sentences, we extract the interesting phrases from the titles. These interesting phrases in most cases shall hint upon the domains of the papers. Writing is largely subjective, and each author's perspective of the goal of their research dictates its representation. But in order to garner a wider audience, they might hint upon the larger domain.

In the keyword section of a research paper, the authors list the key phrases or key words of their documents [39]. Since titles tend to be unique, their constituents may not by themselves be good representatives of general domain areas. The keywords on the other hand are more commonly and widely used, well accepted set of general terms that various authors use to label their work. Hence they serve as generic terms which authors might use to depict their domains, problem-areas and techniques. We combine the knowledge gained from the interesting phrases from the title with the keywords and key phrases of the respective paper.

Thus essentially we are using the important sections of a paper to get at the major theme of that paper. To retrieve the generic aspect of the interesting phrase, we retain those keywords and/or key phrases that have any words in common with the interesting phrase.

4.4.1. Extracting Derivatives

Grammatically, the title of a paper could be a sentence, clause or phrase. We scan each title, $\boldsymbol{T_{ti}}$ to find the prepositions with intention sense.

Next, we list various example permutations of $\boldsymbol{p_i}$ and $\boldsymbol{p_o}$ within an example title, $\boldsymbol{T_{ti}}$. Note that in a research paper title, one or more instances of $\boldsymbol{p_i}$ and $\boldsymbol{p_o}$ can occur in several, all or more permutations than the ones listed here:

$\boldsymbol{T_{ti}} = w_1..w_{j-2}\boldsymbol{p_i}w_j..w_{k-2}\boldsymbol{p_i}w_k..w_{l-2}\boldsymbol{p_o}w_l..w_{m-2}\boldsymbol{p_i}w_m..w_n$

Next, we extract those interesting phrases that follow any instance of a $\boldsymbol{p_i}$ preposition and are delimited at the other end by any instance of a $\boldsymbol{p_i}$ or $\boldsymbol{p_o}$ or the end of the title. For title, $\boldsymbol{T_{ti}}$, the phrases of interest, $\boldsymbol{PHOI_{ti}}$ are listed here:

$$\boldsymbol{PHOI_{ti}} = \begin{matrix} w_jw_{j+1}..w_{k-3}w_{k-2} \\ w_kw_{k+1}..w_{l-3}w_{l-2} \\ w_mw_{m+1}..w_{n-1}w_n \end{matrix}$$

The next step involves finding an intersection between phrases in set, $\boldsymbol{PHOI_{ti}}$ with the keyword section, $\boldsymbol{KW_{ti}}$ of that particular paper. In this step, we retain those keyword or keyword phrases which have one or more words in common with the interesting phrases. This resultant set, $\boldsymbol{D_{ti}}$ or the derivative becomes the main element of our analysis. The following is an example set, $\boldsymbol{KW_{ti}}$ of paper with title, $\boldsymbol{T_{ti}}$:

$$\boldsymbol{KW_{ti}} = \begin{matrix} w_jw_{k+1} \\ w_{p-3}w_{p-2} \\ w_qw_{q+1}w_{q+2} \end{matrix}$$

$$w_m w_{m+1} w_{k-1}$$

Note that words in the key phrases appearing in the keyword set could be in any order. We would like to stress that our approach considers a word by itself as a stand-alone entity and hence the order of words in the key phrases with respect to the interesting phrases does not matter. It is the word's appearance at strategic locations within the interesting phrase and the keyword section which clues us in to its importance in its part as the derivative. The interesting phrase already has a meaning based on its derivation and its words find accentuated generic meaning when they also occur within the keyword section. Hence our technique *infers* the meaning of a word without actually using a dictionary, thesaurus or even NLP.

The resultant derivative set, $\boldsymbol{D_{ti}}$ of that paper looks like this:

$$\boldsymbol{D_{ti}} = \begin{matrix} w_j w_{k+1} \\ w_m w_{m+1} w_{k-1} \end{matrix}$$

4.5. Supervised Classification

Classification is the task of assigning one of a small number of discrete valued labels to the input data. We classify each derivative as a "Domain" or "Not Domain". Hence our classifier takes the approach of supervised learning as the training data (derivative) will be accompanied by labels indicating the class of the derivative.

We build a repository of sub-domain areas in a major domain area of a scientific field through extensive research and analysis of important and trending topics across various scientific conferences and journals. These sub-domains are considered domains as they are nodes in the hierarchical structure alluded to in Chapter 2 Section 2.3. This repository consists of a list of single words or unigrams (1-grams). These unigrams either as stand-alone or as part of a phrase built from other members of this list represent well accepted

domain areas. We may wish to point out that though such unigrams by themselves may sometimes not be domains, but them being a part of the topics from which they are derived, make them a domain word. We stress on the fact that this list contains well accepted domains as the latter have been obtained from credible sources viz. scientific conferences which are organized by experts in said scientific field.

We analyze each derivative, and if it has any word from this repository, we label the derivative as a "Domain". In case the derivative finds no match in the repository, that labels it as a "Not Domain". Thus, we analyze the list of derivatives and assign corresponding class labels to them. We reiterate that without knowing the actual meaning of a word, we are *inferring* its significance. Such as a word in the derivative is likely a domain word if is found in the repository of domains list.

The next step in creating the classifier is deciding what features of the derivatives are relevant.

4.5.1. Session Identifiers

A scientific conference has various sessions each of which assembles the papers dealing with similar topics in one group. Each such session is identified by a name which represents the topic of each group in a comprehensive yet succinct way. Hence logically this session identifier represents the domain of its group of papers. We process each derivative to see if it has any word in common with the session identifier. Any common word between the derivative and the session identifier sets the feature of the derivative as "Found in Session: True". No common word sets the feature as "Found in Session: False". An important point to be noted is that we do not restrict each derivative of a paper to the latter's respective session identifier. Rather we compare it across the entire set of session identifiers across the years of the conference under analysis and consider at least one match as a positive find and

no match at all as negative. The reason we use the entire set is that grouping of the papers into each session and naming the session identifier is subjective and based on the conference committee's opinions and preferences.

4.5.2. Abstract Count

An abstract of a paper is written so as to contain the main elements of the paper in a synoptic form [22]. This makes it a likely section to contain the underlying theme and hence the domain area of the paper. Therefore the likelihood of any word of the derivative to be a domain word could be supported by its appearance in its respective paper's abstract. Since the domains are generic and different papers could share a domain area, hence we match words from each derivative across all papers in the data set. Therefore we count the abstracts containing at least one word of the derivative. This frequency becomes a relevant feature, because different abstracts containing the words of the derivative validate the importance of a derivative. If a derivative contains more domain words, it adds to its validity of becoming a domain as a whole. For example, a derivative "pattern recognition" has a count of 50, if "pattern" occurs in 30, "pattern recognition" occurs in 5 and "recognition" occurs in 15 abstracts.

We discretize the count of the abstracts as integer values from 1 to 5, after dividing the count values into groups of 5.

4.5.3. Naïve Bayes Classifier

The Bayes rule in probability theory is represented in equation 4.1 [40].

$$\boldsymbol{P(Y|X)} = \frac{\boldsymbol{P(X|Y)P(Y)}}{\boldsymbol{P(X)}} \tag{4.1}$$

Our feature extractor functions create a feature set containing relevant feature values for all given derivatives. Since the appearance of a word of the derivative in a session identifier and in an abstract is independent, we use a Naïve Bayes classifier as it works well with independent features. We denote a feature vector as $\boldsymbol{X}$ and the class label as $\boldsymbol{Y}$. Our feature vector is represented in equation 4.2 [40].

$$\begin{aligned} &\boldsymbol{X} = [X_1 X_2] \\ &\text{where } X_1 = \textit{Found in Session Identifier} \\ &\text{where } X_2 = \textit{Abstract Count} \end{aligned} \tag{4.2}$$

The class label $\boldsymbol{Y}$ takes binary values as represented in set:

$$\boldsymbol{Y} = \begin{matrix} \textit{Domain} \\ \textit{Not Domain} \end{matrix}$$

We would like to model $\boldsymbol{P(X|Y)}$, where $\boldsymbol{X}$ is a feature vector, and $\boldsymbol{Y}$ is its associated label. Our task is demonstrated in Fig. 4.1. It may be pointed out that feature X_1 is a binary attribute, while feature X_2 is a 5-valued attribute.

In order to accurately estimate $\boldsymbol{P(X|Y)}$, we need to consider the number of parameters we must estimate, given our $\boldsymbol{X}$ and $\boldsymbol{Y}$. Hence we need to estimate a set of parameters, $\boldsymbol{\theta_{ij}}$ given in equation 4.3.

$$\boldsymbol{\theta_{ij} = P(X = x_i | Y = y_j)} \tag{4.3}$$

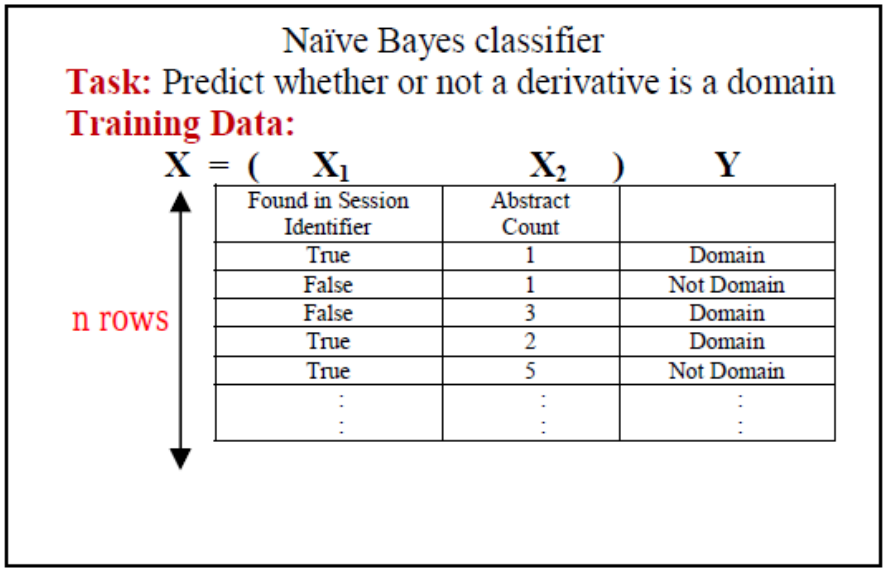


Figure 4.1: Naïve Bayes classifier

Since Naïve Bayes works with the simplified assumption of conditional independence among the attributes, $\boldsymbol{P(X|Y)}$ is calculated using equation 4.4.

$$\boldsymbol{P(X|Y) = P(X_1|Y)P(X_2|Y)} \tag{4.4}$$

The conditional independence assumption reduces the number of parameters to be estimated. Although this reduction may not be dramatic enough for our case, given the small number of features and their possible values, we may wish to point out that it will be considerable when we apply the Naïve Bayes classifier to extract more knowledge from research papers. An example of this knowledge is the set of techniques applied in research papers. The reason for this is that the relevant features for techniques may be more in number, and additionally may have multiple values.

4.6. Our Technique Exemplified

We describe our approach using an example. Fig. 4.2a and Fig. 4.2b depict diagrams portraying the steps to arrive at the derivative. We use data of a paper from the ACM Special Interest Group on Data Communication (SIGCOMM) 2013 conference.

Fig. 4.3 depicts the processing for each derivative to find relevant features using all session identifiers and abstracts from all papers of SIGCOMM conference series from years 2010-2014.

Section 4.7 discusses the results of our experiments in detail.

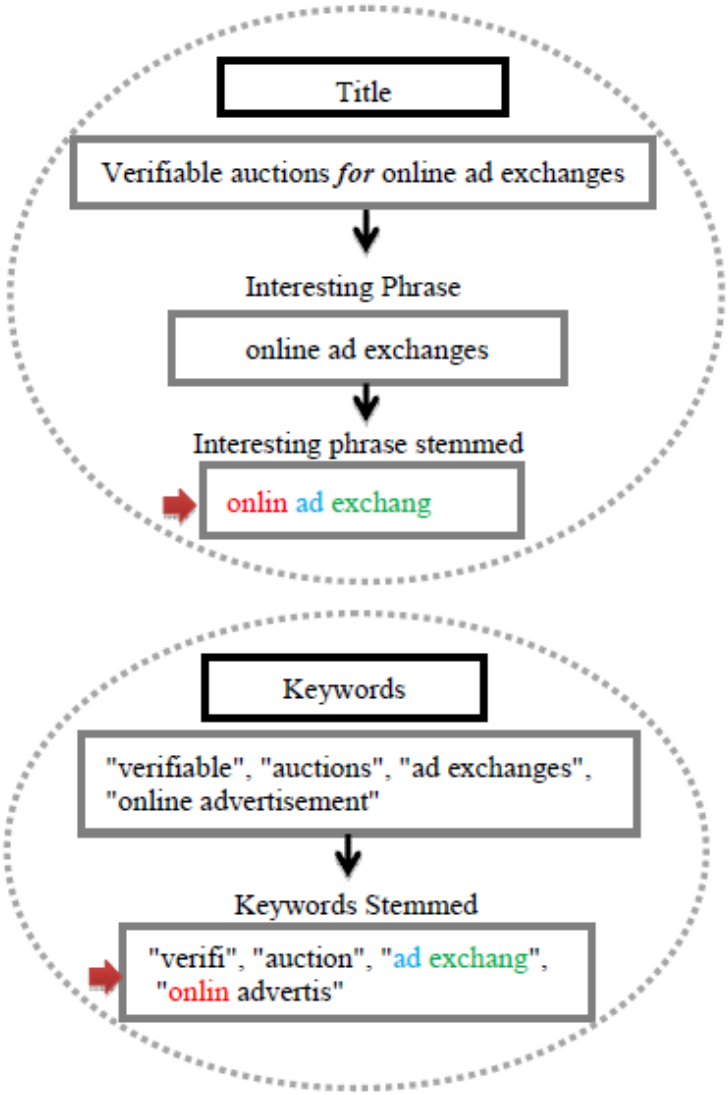
4.7. Preliminary Experimental Evaluation

We have programmed our technique using Python and some of its packages including NLTK. Although our approach is extendable to any scientific field, we test our technique on the research conferences in the field of Computer Science.

In order to create a repository of domain areas, our strategy is to collect the topics from the Calls for Papers (CFP) of top conferences of a large domain within Computer Science. CFP for any conference contain topics under which papers are sought. Hence they are one of the definitive sources of domains, well-accepted by experts in the scientific field. These topics are in the form of sentences, clauses or phrases. We remove all the punctuations, stopwords and newline characters from these topics. This corpus is then stemmed, and each word hence becomes a domain word in our list of domains.

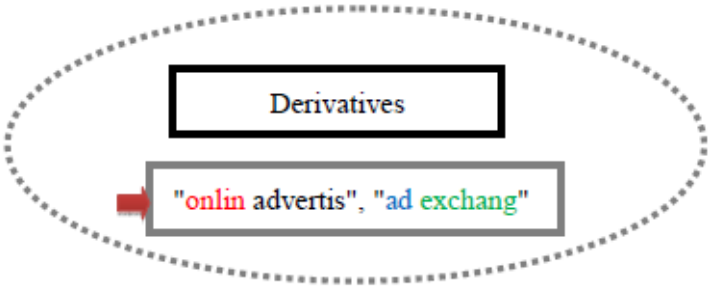
4.7.1. Datasets Used

In a set of experiments on conferences on Data Mining, we collected the topics from the Calls for Papers sections from the IEEE International Conference on Data Mining series



(a) Extracting interesting phrase from title and presenting the keywords

Figure 4.2: Diagram for our technique



(b) Extracting derivatives

Figure 4.2: Diagram for our technique

(ICDM), the IEEE International Conference on Data Engineering (ICDE), and the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD) from 2010-2014. These topics give us the domains repository. For data analysis, we collected papers from ACM SIGKDD from years 2010-2014. This data consists of 939 papers from all sessions including the regular research track sessions, in addition to the keynote, panel, demonstration, poster, industrial and government track. We have extracted titles, and keyword lists from each of these papers. Of the 939 paper titles, 367 have prepositions with intention sense. Of the 367, we get 228 non-empty derivative sets. These non-empty derivative sets result when there is a match between the interesting phrase and the keyword list. From the 228 non-empty derivative sets, we get 272 derivatives, because one derivative set can have more than one derivative.

The final dataset of 272 (ACM SIGKDD) derived as explained above is small at a first look, but the key thing to note here is that this is the derivative list. These are the derivatives which were extracted using our technique, from their "respective" papers, and have subsequently become the key element of analysis. We emphasize that our point of contention was never the size of the dataset, rather the intelligence we derive from it, based on fusion of different sources of data. We process the derivatives using the list of all session identifiers for

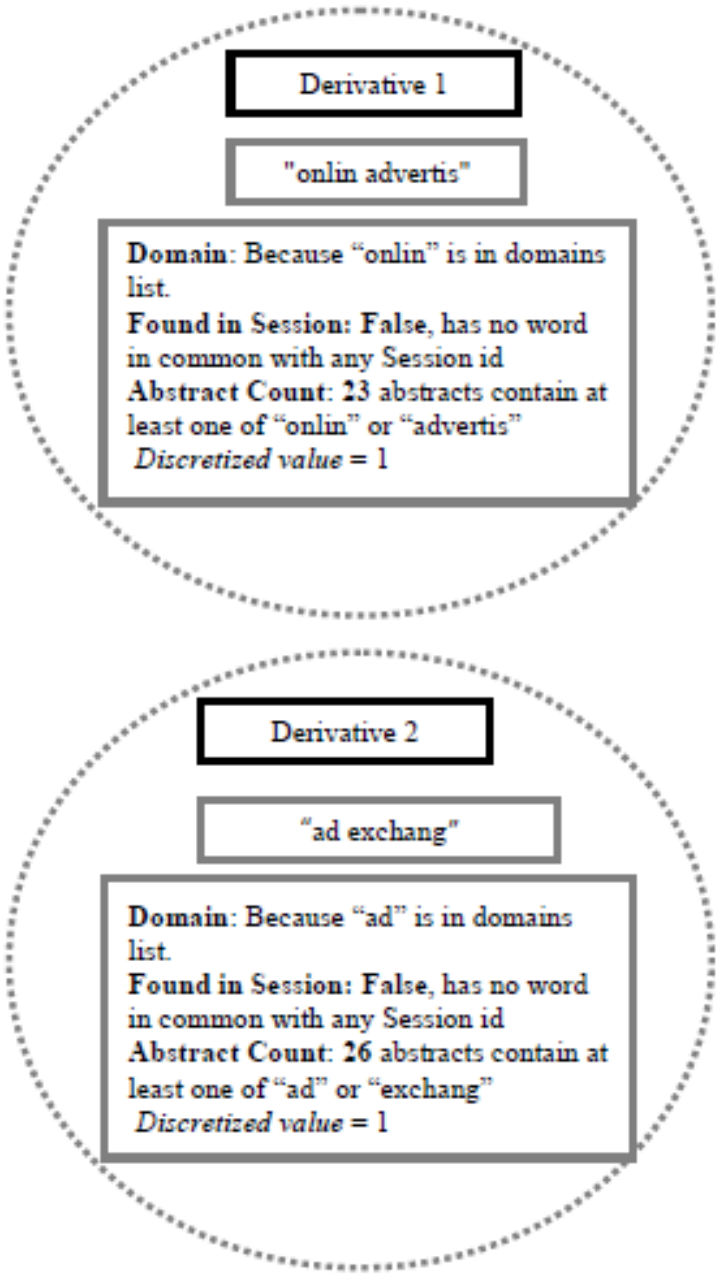


Figure 4.3: Processing each derivative

Table 4.1: Count of successive datasets

Conference	**Titles**	**Titles with *PI***	**Derivatives**
SIGKDD	939	367	272
SIGCOMM	414	136	99
ICDCS	369	139	113

reasons noted in Section 4.5.1. Session identifiers have been rarely used in identifying true domains of papers; despite the fact that they prove to be good sources of useful information. Hence we have innovated on using them as a feature. We use the abstracts from all the papers of all the years of the conference under analysis, viz. KDD. The reason simply is that authors exercise their choice in choosing titles and may not use prepositions with intention sense. But this no way implies that their domain is not the same as that of the authors that do use prepositions with intention sense. Hence we cannot restrict the "analysis" of our derivatives to only the abstracts of the papers from which they are derived.

Table 4.1 summarizes the count of the successive datasets as we progress in our analysis in various sets of experiments.

After having extracted the feature sets for the derivative data as explained above, we divide them into a training set and a test set in the ratio of 70%-30% respectively. The training set is used to train a Naïve Bayes classifier.

To validate the efficacy of our technique we conducted a set of experiments on conferences on Computer Networks and Wireless Communication, where we created a domain list using topics from the Calls for Papers sections from the IEEE International Conference on Computer Communications (INFOCOM), the ACM International Conference on Mobile Computing and Networking (MobiCom), and the ACM Special Interest Group on Data Communication (SIGCOMM) from 2010-2014. We collected papers from ACM (SIGCOMM) from 2010-2014.

In a set of experiments on conferences on Distributed and Parallel Computing, we gathered a domain list using Call for Papers sections from IEEE International Conference on Distributed Computer Systems (ICDCS), the IEEE International Parallel and Distributed Processing Symposium (IPDPS) and the ACM Symposium on Principles of Distributed Computing (PODC) from 2010-2014. For data analysis, we collected papers from IEEE ICDCS from 2010-2014.

4.7.2. Results

The two most frequent and basic measures for information retrieval effectiveness are precision and recall. In binary classification, precision is the fraction of retrieved instances that are relevant, while recall is the fraction of relevant instances that are retrieved. The precision and recall values are calculated using true positives, false positives, and false negatives which result from running the classifier on the test set. The formula is given in equation 4.5. True positives (TP), refer to the cases within the test set when domains are correctly identified, while false positives (FP) mean when certain "not domains" are labeled as domains. True negatives (TN) on the other hand correctly identify "not domains", while false negatives (FN) incorrectly label domains as "not domains".

$$Precision = \frac{TP}{TP + FP}$$
$$Recall = \frac{TP}{TP + FN} \tag{4.5}$$

The values for TP, FP, TN, and FN for one iteration of each dataset are listed in Table 4.2.

Table 4.2: TP, FP, TN, FN values for 1 iteration

Conference	**TP**	**FP**	**TN**	**FN**	**Precision**	**Recall**
SIGKDD	52	2	21	6	0.963	0.8965
SIGCOMM	15	2	8	4	0.8823	0.7895
ICDCS	15	6	10	2	0.7143	0.8823

Table 4.3: Average precision and recall

Conference	**Precision**	**Recall**
SIGKDD	95.54%	87.97%
SIGCOMM	90.42%	76.60%
ICDCS	77.15%	81.88%

Our technique has high precision and high recall as is demonstrated by average precision and recall values from the 100 iterations for each set of experiments. These values in percentages are tabulated in Table 4.3.

There is generally a tradeoff between precision and recall, where a higher value of one can be achieved at the cost of the other. Our technique scores as it generates fairly high values for both precision and recall.

The accuracy of the classifier is defined in equation 4.6.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.6}$$

The average accuracy of the classifier from the 100 iterations for each set of experiments is tabulated in Table 4.4.

Table 4.4: Average accuracy

Conference	Accuracy
SIGKDD	87.05%
SIGCOMM	77.24%
ICDCS	74.33%

4.8. Exhaustive Experimental Analysis

We have conducted extensive experiments on datasets from subject areas within Computing field: Networking, Digital Content and Software. The exhaustive list of the publications, which we have used for analysis, is presented in Table 4.5. In case of conferences, symposiums and workshops, in addition to papers from the regular research track sessions, we have included papers from other sessions as well, such as keynote, panel, demonstration, poster, industrial and government track sessions. Note that each publication can cater to one or several subject areas. The corresponding years in the parenthesis represent that the data is available for that range of years and hence only that data has been used in our experiments. Note that some conferences were held by ACM in conjunction with other organizations during some of the years, while some had special interest group names attached to them, however we record the names as appearing in the latest year of publication. Subject areas,"Networking", "Digital Content" and "Software" are abbreviated as "N", "D" and "S", respectively in this table.

4.8.1. Datasets Used

For each subject area under analysis we first conducted experiments on all the publications under it on a per year basis.

Table 4.5: Conference data used for analysis; N:Networking, D:Digital Content, S:Software

Conference Name	**Subject Area**
ACM/IEEE Symposium on Architectures for Networking and Communications Systems ANCS (05-14)	N
ACM SIGSAC Conference on Computer and Communications Security CCS (05-14)	N, D, S
Annual IEEE/ACM International Symposium on Code Generation and Optimization CGO (05-14)	S
ACM MobiCom Workshop on Challenged Networks CHANTS (06-14)	N
ACM International Conference on Information and Knowledge Management CIKM (05-14)	N, D, S
ACM/IEEE/IFP International Conference on Hardware/Software Codesign and System Synthesis CODES+ISSS (05-13)	S
ACM International Conference on emerging Networking EXperiments and Technologies CoNEXT (05-14)	N
International Workshop on Data Management on New Hardware DaMoN (05-14)	D
ACM Symposium on Dynamic languages DLS (05, 07-14)	S
ACM Symposium on Document Engineering DocEng (05-14)	N, D, S
ACM International Workshop on Data Warehousing and OLAP DOLAP (05-14)	N, D, S
International Conference on Embedded Software EMSOFT (05-14)	S
European Conference on Computer Systems EuroSys (06-14)	N, S
International Conference on Generative Programming: Concepts and Experiences GPCE (05-14)	S
ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games I3D (05-14)	S
ACM SIGPLAN International Conference on Functional Programming ICFP (05-14)	S

Conference Name	Subject Area
International Conference on Software Engineering ICSE (05-14)	S
ACM International Symposium on Memory Management ISMM (06-14)	S
International Conference on Interaction Design and Children IDC (05-14)	N, D
Conference on Internet Measurement Conference IMC (05-14)	N, D
ACM/IEEE-CS Joint Conference on Digital Libraries JCDL (05-13)	N, D, S
ACM SIGKDD International Conference on Knowledge Discovery and Data Mining KDD (05-14)	D
ACM SIGPLAN/SIGBED Conference on Languages, Compilers, and Tools for Embedded Systems LCTES (05-14)	S
Annual International Conference on Mobile Computing and Networking MobiCom (05-14)	N
ACM International Workshop on Data Engineering for Wireless and Mobile Access MobiDE (05-13)	D
ACM International Symposium on Mobile Ad-Hoc Networking and Computing MobiHoc (05-14)	N
Annual International Conference on Mobile Systems, Applications, and Services MobiSys (05-14)	N
ACM International Symposium on Mobility Management and Wireless Access MobiWac (06-14)	S
ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems MSWiM (05-13)	S
ACM Workshop on Network and Operating Systems Support for Digital Audio and Video NOSSDAV (05-06, 08-14)	N, D, S
ACM SIGPLAN Workshop on Partial Evaluation and Program Manipulation PEPM (06-14)	S
ACM International Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks PE-WASUN (05-14)	S

Conference Name	Subject Area
ACM Workshop on Programming Languages and Analysis for Security PLAS (06-14)	S
ACM SIGPLAN Conference on Programming Language Design and Implementation PLDI (05-14)	S
ACM Symposium on Principles of Distributed Computing PODC (05-14)	N, S
ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems PODS (05-14)	D
International Symposium on Principles and Practice of Declarative Programming PPDP (05-13)	S
ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming PPoPP (05-14)	S
ACM Symposium on QoS and Security for Wireless and Mobile Networks Q2SWinet (05-12, 14)	S
Annual ACM Symposium on Applied Computing SAC (05-14)	N, S
ACM SIGGRAPH / Eurographics Symposium on Computer Animation SCA (05-13)	S
ACM Conference on Embedded Networked Sensor Systems SenSys (05-14)	N, D, S
ACM conference on SIGCOMM SIGCOMM (05-14)	N
International ACM SIGIR Conference on Research and Development in Information Retrieval SIGIR (05-14)	D
ACM International Conference on Measurement and Modeling of Computer Systems SIGMETRICS (05-14)	N, D
ACM SIGMOD International Conference on Management of Data SIGMOD (05-14)	D
ACM International Joint Conference on Pervasive and Ubiquitous Computing UbiComp (05-14)	N
Annual ACM Symposium on User Interface Software and Technology UIST (05-14)	S
ACM international workshop on Vehicular inter-networking, systems, and applications VANET (05-13)	N

Conference Name	Subject Area
ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments VEE (05-14)	N, S
ACM Symposium on Virtual Reality Software and Technology VRST (05-10, 12-14)	S
International ACM Conference on 3D Web Technologies Web3D (05-14)	S
ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation, and Characterization WiNTECH (06-14)	N
Workshop on Privacy in the Electronic Society WPES (05-14)	N, D, S
Winter Simulation Conference: Simulation: Making Decisions in a Complex World WSC (05-13)	S
International Conference on Underwater Networks & Systems WUWNET (06-14)	N, D, S

We have used 26, 18, and 37 conferences respectively from the subject areas: "Networking", "Digital Content" and "Software". For reference, these conferences are listed in Table 4.5, where for example, 26 conferences are marked as "N", signifying subject area as "Networking". Tables 4.6, 4.7 and 4.8 present the yearly count of titles, titles containing prepositions with intention sense, and derivatives derived from the conferences in these subject areas.

It can be seen that the derivative count is small. But irrespective of the size of this data, we are interested in the intelligence we derive from it.

Table 4.6: Count of datasets for 26 conferences in “Networking”

Year	Titles	Titles with PI	Derivatives
2005	1190	506	442
2006	1293	539	483
2007	1303	557	497
2008	1649	713	699
2009	1616	701	679
2010	1708	723	702
2011	1689	671	659
2012	2078	875	885
2013	1867	760	774
2014	1725	724	677

Table 4.7: Count of datasets for 18 conferences in “Digital Content”

Year	Titles	Titles with PI	Derivatives
2005	966	386	295
2006	1007	382	343
2007	1121	453	402
2008	1276	522	475
2009	1338	548	493
2010	1361	559	512
2011	1502	569	535
2012	1630	640	646
2013	1542	604	606
2014	1439	567	518

Table 4.8: Count of datasets for 37 conferences in "Software"

Year	Titles	Titles with PI	Derivatives
2005	2025	875	602
2006	2053	928	691
2007	1864	839	590
2008	2365	1098	887
2009	2120	973	788
2010	2228	1003	810
2011	2420	1006	789
2012	2672	1160	842
2013	2466	1077	807
2014	1597	672	646

4.8.2. Classifiers

The generative classifier, Naïve Bayes works with the simplified assumption of conditional independence among the features, and hence converges to its asymptotic accuracy faster [41]. This is especially true where data sets are smaller. Our features are independent since a derivative's constituents can appear in the abstract irrespective of their appearance in session identifiers. Moreover, we are looking at scientific publications in a specific subject area on a yearly basis. Hence the data under consideration is small and the derivatives are even a further subset of this data. Decision Trees are advantageous in our case as they work well when we do not have to worry whether our data is linearly separable [42]. Support Vector Machines (SVMs) work well as classifiers [43]. Since ours is a data driven approach, we test these three different classifiers, in order to thoroughly analyze the data. We have already described, tested, and seen preliminary results for Naïve Bayes Classifier in Subsection 4.5.3 and Section 4.7.

4.8.3. Results

We have described and defined some measures of information retrieval effectiveness, viz. precision, recall and accuracy in Subsection 4.7.2. F1 score is another such measure and is the harmonic mean of precision and recall and also measures the classifier's accuracy, and it's formula is given in equation 4.7.

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{4.7}$$

Our technique has high precision and high recall as is demonstrated by average precision and recall values from the 100 iterations for each set of experiments. These values in percentages are tabulated in Table 4.9 for the "Networking" subject area. High values of F1 score viz. above 90% across all classifiers, are very impressive and signify that our technique works very well and the classifiers are returning accurate results. Our technique is validated by consistent good performance across the "Digital Content" and "Software" subject areas as illustrated by results in Tables 4.10 and 4.11.

The derivatives which are classified as domains via the supervised learning techniques as described above are in effect domains representative of the state of the art of research in recent years. These could be existing topics as well as newer emerging topics in research. For each subject area for each year from 2005-2014, we extract domains from all publications during that year. Next, again for the subject area under analysis, we look at the domains for two five-year periods, 2005-2009 and 2010-2014. Now, for each five year period, if one domain appears all five years it would have a maximum frequency of 5. We collect the domains which appear in majority of the five years, that is three or more years. The reason is that since

Table 4.9: "Networking": Average precision, recall and F1 score for 100 iterations

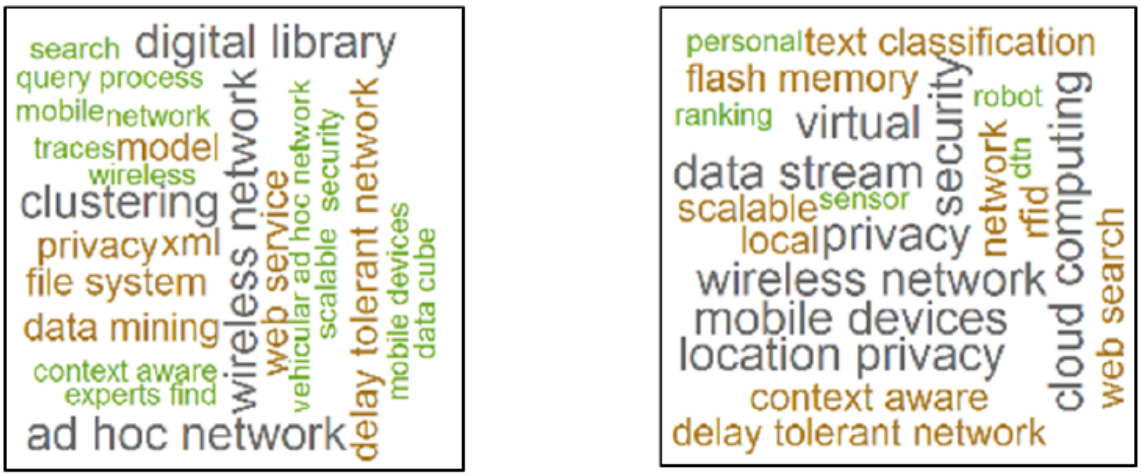
		Naïve Bayes			SVM			Decision Trees		
Year	**Derivatives**	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**
2005	442	95.39%	92.38%	93.86%	95.36%	91.68%	93.48%	95.37%	91.82%	93.56%
2006	483	93.25%	88.42%	90.77%	93.22%	88.03%	90.55%	93.24%	88.08%	90.59%
2007	497	96.08%	88.51%	92.14%	96.88%	88.11%	92.29%	97.02%	88.09%	92.34%
2008	699	95.81%	87.72%	91.59%	95.79%	87.45%	91.43%	95.86%	87.56%	91.52%
2009	679	95.72%	86.38%	90.81%	96.21%	85.41%	90.49%	95.78%	85.75%	90.49%
2010	702	94.91%	86.77%	90.66%	94.88%	85.87%	90.15%	94.99%	86.77%	90.69%
2011	659	94.82%	88.06%	91.32%	93.34%	89.11%	91.18%	93.37%	88.97%	91.12%
2012	885	96.37%	89.70%	92.92%	96.64%	89.03%	92.68%	96.37%	89.61%	92.87%
2013	774	92.42%	88.57%	90.45%	93.02%	88.25%	90.57%	92.97%	88.18%	90.51%
2014	677	94.89%	89.93%	92.34%	94.97%	88.15%	91.43%	94.89%	89.86%	92.31%

Table 4.10: "Digital Content": Average precision, recall and F1 score for 100 iterations

		Naïve Bayes			SVM			Decision Trees		
Year	**Derivatives**	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**
2005	295	96.81%	93.76%	95.26%	96.81%	94.76%	95.77%	96.81%	94.76%	95.77%
2006	343	91.12%	91.23%	91.17%	91.09%	90.86%	90.97%	91.09%	90.84%	90.96%
2007	402	94.87%	87.05%	90.79%	93.10%	87.94%	90.45%	92.93%	87.95%	90.37%
2008	475	96.20%	89.76%	92.87%	96.19%	89.50%	92.72%	96.19%	89.42%	92.68%
2009	493	95.56%	86.98%	91.07%	95.53%	86.13%	90.59%	95.38%	87.07%	91.04%
2010	512	94.45%	84.70%	89.31%	85.86%	100.00%	92.39%	85.86%	100.00%	92.39%
2011	535	96.03%	89.29%	92.54%	92.21%	92.82%	92.51%	91.57%	93.63%	92.59%
2012	646	95.84%	87.63%	91.55%	93.97%	89.04%	91.44%	93.18%	89.59%	91.35%
2013	606	93.92%	82.40%	87.78%	83.54%	96.10%	89.38%	83.36%	96.45%	89.43%
2014	518	94.40%	80.40%	86.84%	83.38%	99.39%	90.68%	83.38%	99.39%	90.68%

Table 4.11: "Software": Average precision, recall and F1 score for 100 iterations

		Naïve Bayes			SVM			Decision Trees		
Year	**Derivatives**	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**
2005	602	96.75%	93.91%	95.31%	96.75%	93.91%	95.31%	96.75%	93.91%	95.31%
2006	691	94.71%	92.41%	93.55%	94.71%	92.41%	93.55%	94.71%	92.41%	93.55%
2007	590	95.01%	91.77%	93.36%	95.00%	91.62%	93.28%	95.00%	91.62%	93.28%
2008	887	95.10%	90.78%	92.89%	95.10%	90.78%	92.89%	95.10%	90.78%	92.89%
2009	788	95.78%	89.19%	92.37%	95.77%	89.07%	92.30%	95.77%	89.06%	92.29%
2010	810	94.05%	91.85%	92.94%	94.05%	91.82%	92.92%	94.04%	91.63%	92.82%
2011	789	95.85%	93.17%	94.49%	95.89%	93.02%	94.43%	95.89%	93.01%	94.43%
2012	842	96.07%	93.02%	94.52%	96.07%	92.85%	94.43%	96.07%	92.85%	94.43%
2013	807	95.61%	89.04%	92.21%	95.61%	89.04%	92.21%	95.61%	89.04%	92.21%
2014	646	97.92%	87.28%	92.29%	94.45%	89.90%	92.12%	94.45%	89.89%	92.11%



(a) “Networking”

Figure 4.4: Trending domains: 2005-2009 (left) and 2010-2014 (right)

we are interested in the trend of the state of the art of research, we need to know which domains are being researched more in the five year period. We present the trending domains for the three subject areas for 2005-2009 and 2010-2014, in Figures Fig. 4.4a, Fig. 4.4b and Fig. 4.4c.

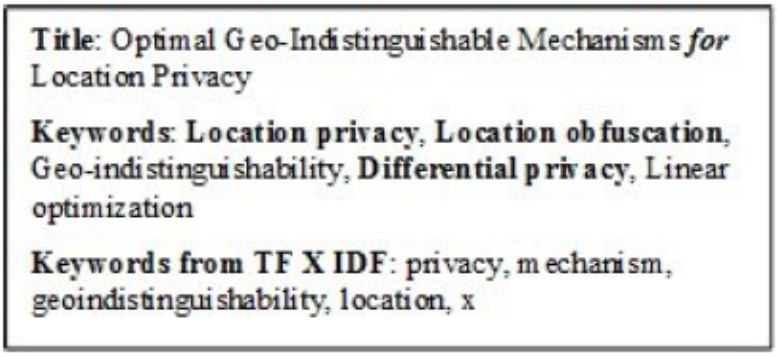
A technique, namely term frequency - inverse document frequency (tf-idf) has very often been used in extraction of keywords of each document in a collection [44]. This technique combines the frequency of a phrase within a document with its inverse document collection frequency to generate a composite weight of that phrase for each document. For the subject area, “Digital Content”, for the years 2010-2014, we use a corpus of full papers along with session identifiers. We extract the keywords of each paper using tf-idf. These keywords are upto 3-grams.

Now we will justify our approach of using keywords for extracting derivatives and not the ones generated by tf-idf. We would like to discuss specific examples, wherein tf-idf loses out on specific information. We take the example of a paper from the ACM SIGSAC Conference on Computer and Communications Security (CCS) 2014. Fig. 4.5a depicts the original

(b) "Digital Content"

(c) "Software"

Figure 4.4: Trending domains: 2005-2009 (left) and 2010-2014 (right)



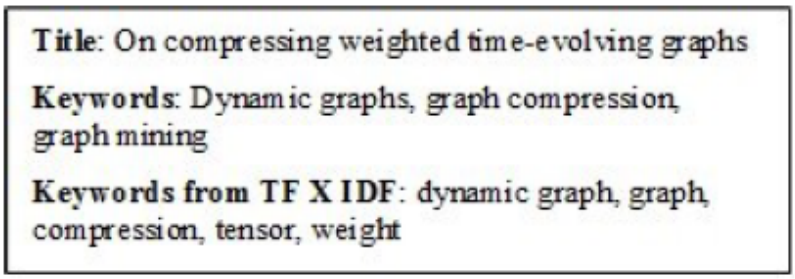
(a) ACM CCS 2014 paper

Figure 4.5: Comparing keywords against ones obtained by tf-idf

keywords in the paper and the ones obtained by tf-idf. Fig. 4.5a also has the keywords that our techniques takes into account (derivatives) for domain extraction, marked in bold font. It can be easily seen that the tf-idf keywords miss out these latter keywords. As another example, we take a paper from ACM International Conference on Information and Knowledge Management (CIKM) 2012. Fig. 4.5b depicts the original keywords in the paper and the ones obtained by tf-idf. It can be seen that the word "mining" is missed out in tf-idf keywords. It could be owing to its larger frequency across the collection of "Digital Content" subject area which offsets its importance and makes it appear as a common redundant word. However for the focus of our domain extraction technique, we absolutely cannot do away with an important word such as "mining" as the latter has particular significance in the area of "Digital Content".

4.9. Conclusions and Future Work

We have obtained very encouraging results from our technique. We have applied *fusion of NLP with supervised classification* and developed a methodology for extracting domains from scientific papers. We have used a *fusion of data* from different strategic sections of each



(b) ACM CIKM 2012 paper

Figure 4.5: Comparing keywords against ones obtained by tf-idf

paper. Thus our approach contributes to exciting possibilities for developing the genre of hybrid methodologies. We have introduced this technique in [16]. We have published the complete motivation, technique and results as described above in [17].

We have performed extensive experiments on datasets from subject areas within Computing field: Networking, Digital Content and Software, and achieved good results validating our approach.

The domains learned from this technique also consist of newer domains reflecting the current state of the art of research, besides the already well-known domains. We are interested in finding out how these domains reflect the changing landscape of research. In the next chapter, we discuss our technique towards this goal.

CHAPTER 5

MINING DOMAIN SIMILARITY TO ENHANCE DIGITAL INDEXING

The research in each research paper, focuses on, draws ideas from several domains. These domains are really the broad topics or purpose of the research paper. It would be immensely helpful if such a research paper were tagged by these domains. This would aid the researcher in scanning these domains to figure out their interest in the paper.

To the best of our knowledge, there is no definitive database of domains in a given subject area in any given scientific field. In Chapter 4, we have used Call for Papers of various conferences as database of domains. In this chapter, we look at another widely accepted resource of domains, and aim at converging the knowledge from the sources. In the scientific field of computing, there is a system called Association for Computing Machinery Computing Classification System (ACM CCS) designed by ACM [45]. Since 1960's it is a standard scheme that has a set of domains, which are used to tag research articles. These domains characterize the topics of the state of the art of the computing field. Essentially each article is tagged by its relevant domains. But these latter domains come from a fixed set, namely, ACM CCS. The 2012 ACM CCS is the latest version. The state of the art of the research in computing field changes practically every month, with newer problem-areas being worked on. The problem-areas that garner sufficient interest generate sub problem-areas and hence become domains in their own right. We propose a technique to add newer domain topics to the existing similar topics in ACM CCS.

5.1. Related Work

Essentially the ACM CCS has a hierarchical structure with several top level domains, each having several levels of sub-domains under them. Current ontology evolution techniques are prone to inconsistencies and complexities [46]. Finding similarity at the element-level has been found to be more productive than that at the hierarchical level [47]. Ignoring the hierarchical structure of the ontology, domains are the topics at the element-level of the ACM CCS. Hence we focus on the base element of the ontology, namely, domain.

While our technique works with ACM CCS, it is extendible to other taxonomies as well. For design, research and analysis of our technique, we have worked with scientific research articles written in English language, in the computing field.

ACM's digital library is a database of publications. Each publication can be of a different type such as, conference, workshop, symposium, journal, etc. ACM tags each of its publications by the subject areas which that publication's research articles cater to. For example, subject areas such as: "Networking", "Software", and "Digital Content".

In our earlier work [17], we presented a technique to extract state-of-the-art domains from research papers. We work with 18 publications in the ACM digital library from the years 2010-2014 tagged by the subject area "Digital Content". We extract domains from 7474 research articles in these publications. A subset of the extracted domains is presented in Fig. 5.1. For focused analysis, we work with this subset for the scope of this chapter.

5.2. Our Technique

We propose a technique to find similarity between these domains. We further use visualization techniques to depict this similarity. Following this, similar domains are clustered together. These clusters provide insights into how new topics may be introduced to a digital

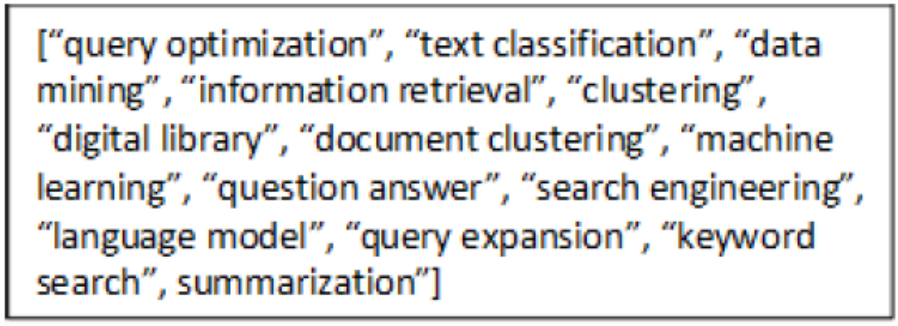


Figure 5.1: A subset of domains from subject area "Digital Content", 2010-14

library classification / tagging / indexing scheme such as ACM CCS. The rest of the chapter is organized as follows. Section 5.3 presents a technique for finding similarity between domains. Section 5.4 describes how to visualize and cluster similar domains. Section 5.5 discusses the results and conclusions.

5.3. Finding Similarity Between Domains

In this section we describe a technique to find pair-wise similarity between domains.

5.3.1. Using WordNet WuP Similarity

Princeton University's WordNet [48] is a large lexical database of English language, which groups words together based on their lexical categories and senses. Each word may have several different senses, and hence several different meanings, based on the context in which the word is used in a sentence or a clause. In order to find out the similarity between two words, there exist several similarity metrics that compare the senses between the given two words. An important thing to note is that high semantic relatedness between two words can result in a higher score even though the words may not be directly similar in meaning. As stated in [49], word relatedness represents a larger set of potential relationships between

Table 5.1: WuP similarity scores example

Word 1	**Word 2**	**WuP Similarity Score**
language	text	0.93
language	document	0.75
text	document	0.77

words, with word meaning similarity being a sub-case of this relatedness. Wu-Palmer (WuP) similarity [50] is a metric that returns a score which denotes how related two word senses are. We use WuP similarity as it has advantages over other similarity metrics in terms of performance [51]. Table 5.1 depicts some examples. WuP similarity scores range from 0 to 1 in the increasing order of similarity.

We record WuP similarity scores between each pair of domains from the domain dataset in Fig. 5.1, in order to assess how each domain semantically relates to the other. For single word domains, we assign to their WuP-domain-similarity score, the WuP similarity score between them. For domains which are phrases, we assign to their WuP-domain-similarity score, the maximum WuP similarity score among all the two word combinations of the constituent words of the phrases. The WuP similarity scores of two word combinations of two example phrases, "natural language processing" and "text mining" are listed in Table 5.2. The maximum score of 0.93, is a positive indicator of domain relatedness as techniques of natural language processing are frequently applied in mining text [52].

TheWuP-domain-similarity scores for the domain dataset given in Fig. 5.1, are presented in Table 5.3. We make some interesting observations. The domains "information retrieval" and "data mining" have a WuP-domain-similarity score of 1. This matches the common parlance, as in the research community, these two phrases are used interchangeably. The domains "document clustering" and "language model" have a high WuP-domain-similarity score of 0.88. This similarity is again corroborated by the fact that these two domains are

Table 5.2: WuP similarity scores for word pairs in: "natural language processing" and "text mining"

Word 1	Word 2	WuP Similarity Score
natural	text	0.53
natural	mining	0.50
language	text	0.93
language	mining	0.40
processing	text	0.27
processing	mining	0.25

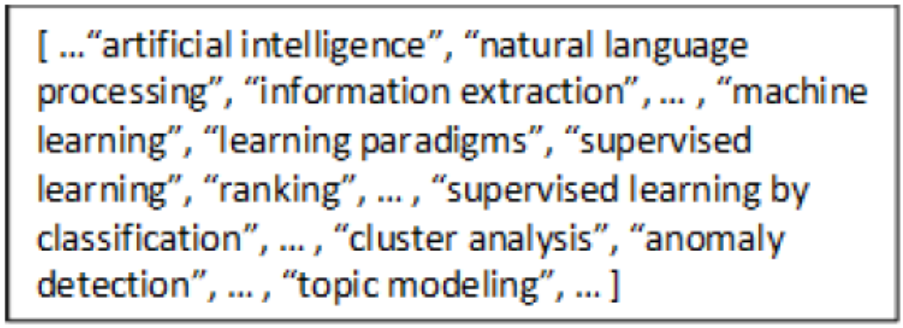


Figure 5.2: A subset of the 11th group of ACM CCS

very closely related in language model based document clustering [53] [54].

5.3.2. Using ACM Computing Classification System (ACM CCS)

We described the concept of ACM CCS in the beginning of this chapter.

As mentioned earlier, the ACM CCS has a hierarchical structure with several top level domains, each having several levels of sub-domains. We ignore the hierarchical structure among the domains and sub-domains and simply combine all sub-domains along with its parent domain into its group. Hence we have 13 groups of domains. A subset of the 11th group of 2012 ACM CCS is depicted in Fig. 5.2.

We analyze each pair of domains with respect to the 13 ACM CCS groups. If each

Table 5.3: WuP-domain-similarity scores for domains given in Fig. 5.1

	clustering	**query optimization**	**language model**	**question answer**	**information retrieval**	**document clustering**	**summarization**	**digital library**	**data mining**	**text classification**	**keyword search**	**machine learning**	**query expansion**	**search engineering**
clustering	1.00	0.40	0.40	0.40	0.73	1.00	0.27	0.73	0.73	0.55	0.33	0.50	0.40	0.33
query optimization	0.40	1.00	0.59	1.00	0.67	0.40	0.67	0.29	0.56	0.59	0.62	0.55	1.00	0.62
language model	0.40	0.59	1.00	0.73	0.71	0.88	0.59	0.62	0.62	0.93	0.86	0.75	0.62	0.86
question answer	0.40	1.00	0.73	1.00	0.75	0.60	0.71	0.43	0.59	0.62	0.67	0.55	1.00	0.67
information retrieval	0.73	0.67	0.71	0.75	1.00	0.73	0.67	0.80	1.00	0.80	0.80	0.80	0.71	0.80
document clustering	1.00	0.40	0.88	0.60	0.73	1.00	0.31	0.73	0.73	0.77	0.40	0.62	0.40	0.59
summarization	0.27	0.67	0.59	0.71	0.67	0.31	1.00	0.29	0.56	0.59	0.62	0.38	0.67	0.62
digital library	0.73	0.29	0.62	0.43	0.80	0.73	0.29	1.00	0.80	0.60	0.33	0.62	0.36	0.89
data mining	0.73	0.56	0.62	0.59	1.00	0.73	0.56	0.80	1.00	0.62	0.62	0.62	0.59	0.62
text classification	0.55	0.59	0.93	0.62	0.80	0.77	0.59	0.60	0.62	1.00	0.80	0.86	0.62	0.80
keyword search	0.33	0.62	0.86	0.67	0.80	0.40	0.62	0.33	0.62	0.80	1.00	0.71	0.67	1.00
machine learning	0.50	0.55	0.75	0.55	0.80	0.62	0.38	0.62	0.62	0.86	0.71	1.00	0.55	0.71
query expansion	0.40	1.00	0.62	1.00	0.71	0.40	0.67	0.36	0.59	0.62	0.67	0.55	1.00	0.67
search engineering	0.33	0.62	0.86	0.67	0.80	0.59	0.62	0.89	0.62	0.80	1.00	0.71	0.67	1.00

domain in the pair has at least one word that belongs to the same group, then we label their ACM-domain-similarity score as 1 on a scale of 0 to 1, with 1 indicating highest similarity.

5.3.3. Combining Domain-similarity from WuP and ACM CCS

WuP similarity, while scoring as a high similarity relatedness metric, is restricted to word senses in the English language. It does not recognize similar words in the computing field vocabulary. In this respect, the ACM-domain-similarity score proves to be useful.

With new techniques spinning off practically every day, newer buzz words are being invented. ACM CCS, while a comprehensive vocabulary of the computing field, may still not keep pace with the rapid and incessant advancement in the field. Regardless of the advancement, the terminology of domains may vary across various research groups. From Section 5.3.1, we see the phrases, "natural language processing" and "text mining" are related as indicated by their high WuP-domain-similarity score. The phrase "text mining" does not appear in any of the 13 groups of ACM CCS domains. And the phrase "natural language processing" appears in only one group and obviously does not share any group with "text mining". Because of this, these two would well be assigned an ACM-domain-similarity score of 0, which would be counterproductive. In cases such as these, the WuP-domain-similarity score can prove to be valuable.

In order to combine the domain-similarity scores obtained from WuP-domain-similarity and ACM-domain-similarity, we need to assign appropriate weights to each. For example, assigning a 50% weight to each, we can add equal contribution of each and record the resulting value as the corresponding final domain-similarity. Table 5.4 presents the matrix for our domains after combining WuP-domain-similarity and ACM-domain-similarity scores. We get some interesting insights. The domains "document clustering" and "language model" have a higher domain-similarity score of 0.94 now after adding the contribution of ACM CCS. And

Table 5.4: Combining WuP-domain-similarity with ACM-domain-similarity

	clustering	**query optimization**	**language model**	**question answer**	**information retrieval**	**document clustering**	**summarization**	**digital library**	**data mining**	**text classification**	**keyword search**	**machine learning**	**query expansion**	**search engineering**
clustering	1.00	0.70	0.70	0.70	0.86	1.00	0.63	0.86	0.86	0.77	0.67	0.75	0.70	0.67
query optimization	0.70	1.00	0.79	1.00	0.83	0.70	0.83	0.64	0.78	0.79	0.81	0.77	1.00	0.81
language model	0.70	0.79	1.00	0.86	0.86	0.94	0.79	0.81	0.81	0.97	0.93	0.88	0.81	0.93
question answer	0.70	1.00	0.86	1.00	0.88	0.80	0.85	0.71	0.79	0.81	0.83	0.77	1.00	0.83
information retrieval	0.86	0.83	0.86	0.88	1.00	0.86	0.83	0.90	1.00	0.90	0.90	0.90	0.86	0.90
document clustering	1.00	0.70	0.94	0.80	0.86	1.00	0.65	0.86	0.86	0.88	0.70	0.81	0.70	0.79
summarization	0.63	0.83	0.79	0.85	0.83	0.65	1.00	0.64	0.78	0.79	0.81	0.69	0.83	0.81
digital library	0.86	0.64	0.81	0.71	0.90	0.86	0.64	1.00	0.90	0.80	0.67	0.81	0.68	0.94
data mining	0.86	0.78	0.81	0.79	1.00	0.86	0.78	0.90	1.00	0.81	0.81	0.81	0.79	0.81
text classification	0.77	0.79	0.97	0.81	0.90	0.88	0.79	0.80	0.81	1.00	0.90	0.93	0.81	0.90
keyword search	0.67	0.81	0.93	0.83	0.90	0.70	0.81	0.67	0.81	0.90	1.00	0.86	0.83	1.00
machine learning	0.75	0.77	0.88	0.77	0.90	0.81	0.69	0.81	0.81	0.93	0.86	1.00	0.77	0.86
query expansion	0.70	1.00	0.81	1.00	0.86	0.70	0.83	0.68	0.79	0.81	0.83	0.77	1.00	0.83
search engineering	0.67	0.81	0.93	0.83	0.90	0.79	0.81	0.94	0.81	0.90	1.00	0.86	0.83	1.00

this is despite the fact that the WuP-domain-similarity of 0.88, discussed in Section 5.3.1, has only half weightage in the overall domain similarity score. This means that the ACM-domain-similarity score of these two domains is contributing effectively to the rest half of the measure. Research already corroborates high relatedness of these domains [53] [54], and the high domain-similarity as evident from our technique, proves that our technique does well in finding related domains.

5.4. Clustering Domains

After finding similarities between domains, we need to combine related domains into groups or clusters.

5.4.1. Using Multidimensional Scaling

Multidimensional Scaling (MDS) provides a visual representation of the pattern of proximities among a set of objects [55], which in our case are the domains. MDS technically finds an optimal configuration of points, corresponding to domains in a 2-dimensional space, which represents how the domains relate to each other. MDS takes as its input a distance matrix. Distance between two domains is the opposite of similarity. To compute the distance between two domains, we subtract their domain-similarity value from 1. We record the pairwise distances of the domains into a distance matrix. We input the distance matrix, which is a symmetric matrix, to the MDS algorithm. The output of the MDS algorithm is a matrix where each row is a domain and the corresponding column entries are the x,y coordinates signifying the location of the domain in a 2-dimensional plane. MDS is essentially a dimensionality reduction technique, where the columns can be seen as the features of the domains.

5.4.2. Using K-Means

We need to cluster the related domains together. We use a popular clustering algorithm, K-Means [56]. The output matrix of MDS serves as the input to K-Means. We need to specify the number of clusters for K-Means algorithm. Experimenting with 3 and 4 clusters gives us Fig. 5.3 and Fig. 5.4 respectively. It may be noted that location of the domains is dictated by the application of MDS and clusters are defined by K-Means.

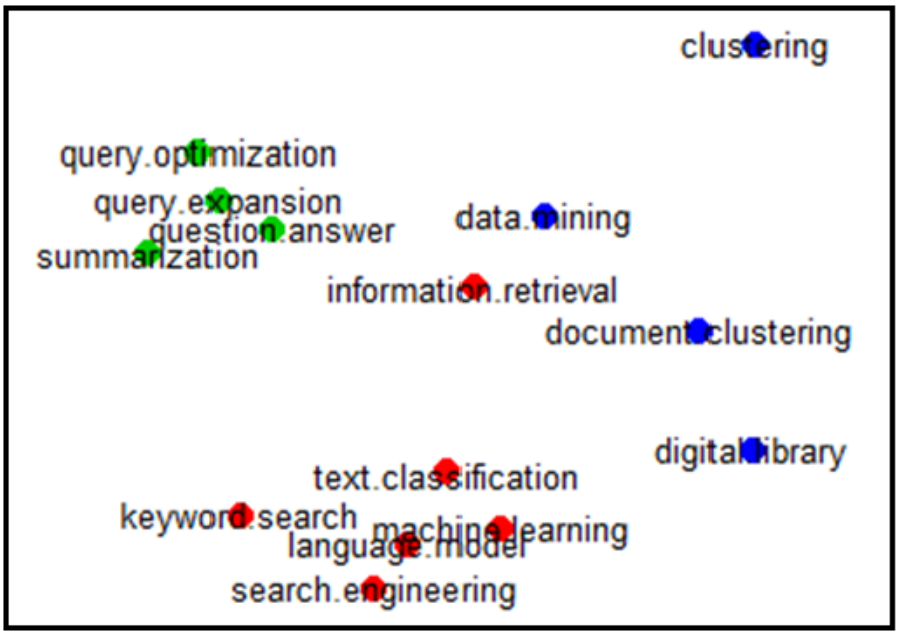


Figure 5.3: K-Means with 3 clusters

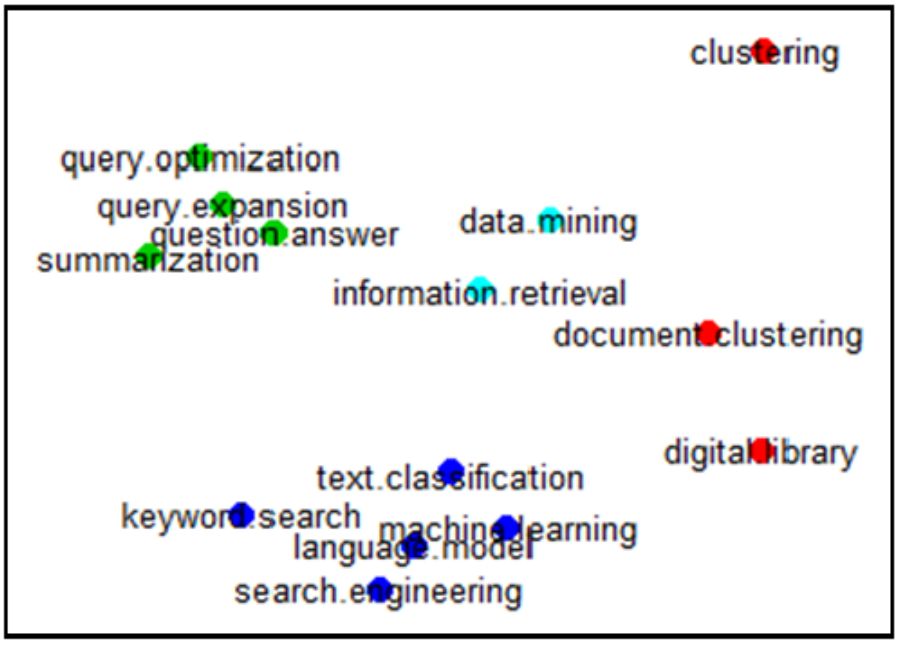


Figure 5.4: K-Means with 4 clusters

5.5. Results and Conclusions

Fig. 5.3 presents interesting insights. The domain "text classification" is clustered with the domain "machine learning". "Text classification" or any related domain, such as "text categorization" do not appear in ACM CCS. It may be noted that "text classification" and "text categorization" have a domain-similarity score of 1. Text classification or categorization has important implications and applications in machine learning [57] [58] [59]. An entry of the domain "text classification" in ACM CCS, perhaps in the 11^{th} group, which is the same group as "machine learning" may help direct search towards articles applying machine learning techniques for text classification or categorization. The work [57], "Machine Learning in Automated Text Categorization" has 8370 citations as of July 2017, indicating the unquestionable interplay between these two domains.

The domain "information retrieval" appears in 8^{th} group of ACM CCS, which also has the domains "query optimization" and "data mining". "Machine learning" appears in the 11^{th} group. As a result of our technique, "information retrieval" is clustered with "machine learning" in Fig. 5.3, and with "data mining" in Fig. 5.4. While Fig. 5.4 clustering maintains it's original ACM CCS grouping with "data mining", Fig. 5.3, introduces it into a different group. "Query optimization" on the other hand doesn't share any cluster with these domains. Does that indicate that the domain "information retrieval" would be better suited in a different group than its original in ACM CCS?

Our technique opens up questions and concerns such as the above. We have presented a method to incorporate newer domains into existing classification / indexing schemes. And we have also indicated a possible re-grouping of domains to increase relevance of search.

We have presented a new domain, "text classification" that could be added to ACM CCS. We have pointed out heavily cited research articles as evidence which bolsters the

results of our technique. Using the count of citations in this case is an example of passive crowdsourcing. The viability of our technique is evident by the fact that our data mining approach has been supported by a passive crowdsourcing approach.

We have published the complete motivation, technique and results as described in this chapter in [18].

An interesting follow up investigation could also be as to what domains are to be phased out as they are no longer in circulation. Our future work involves analyzing the effect on clusters by choosing an optimal value for number of clusters in K-Means. We also plan to evaluate clusters by applying several clustering algorithms, beyond K-Means. We also wish to evaluate several similarity metrics beyond WuP similarity.

BIBLIOGRAPHY

[1] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2011.

[2] E. Mcnulty. (2014). Understanding Big Data: The Seven V's, [Online]. Available: http://dataconomy.com/2014/05/seven-vs-big-data.

[3] J. Xavier. (2013). Google scientist Jeff Dean on how neural networks are improving everything Google does, [Online]. Available: https://www.bizjournals.com/seattle/blog/techflash/2013/08/google-scientist-jeff-dean-on-how.html?page=all.

[4] M. Walker. (2012). Structured vs. Unstructured Data: The Rise of Data Anarchy, [Online]. Available: http://www.datasciencecentral.com/profiles/blogs/structured-vs-unstructured-data-the-rise-of-data-anarchy.

[5] M. Barrenechea. (2013). Big Data: Big Hype?, [Online]. Available: http://www.forbes.com/sites/ciocentral/2013/02/04/big-data-big-hype.

[6] IDC. (2014). The Digital Universe of Opportunities: Rich Data and the Increasing Value of the Internet of Things, [Online]. Available: https://www.emc.com/leadership/digital-universe/2014iview/executive-summary.htm.

[7] K.N.C. (2014). The Economist explains: The backlash against big data, [Online]. Available: http://www.economist.com/blogs/economist-explains/2014/04/economist-explains-10.

[8] C. Cardie. (). History (of natural langage processing) [pdf document], [Online]. Available: http://www.cs.cornell.edu/courses/cs674/2003sp/history-4up.pdf.

[9] B. MacCartney. (2011). Computational Linguistics (aka Natural Language Processing, [Online]. Available: http://nlp.stanford.edu/~wcmac/papers/20110526-symsys-100-nlp.pdf.

[10] C. Wang, M. Danilevsky, N. Desai, Y. Zhang, P. Nguyen, T. Taula, and J. Han, "A Phrase Mining Framework for Recursive Construction of a Topical Hierarchy", in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2013, pp. 437–445.

[11] G. Belsky. (2012). Why Text Mining May Be The Next Big Thing, [Online]. Available: http://business.time.com/2012/03/20/why-text-mining-may-be-the-next-big-thing.

[12] R. V. Noorden, "Trouble at the text mine", *Nature*, vol. 483, pp. 134–135, 7388 2012. DOI: 10.1038/483134a.

[13] S. Reardon, "Text-mining offers clues to success", *Nature*, vol. 509, p. 410, 7501 2014. DOI: 10.1038/509410a.

[14] BBC. (2012). Summly: Teenager launches top-selling news app, [Online]. Available: http://www.bbc.com/news/technology-20181537.

[15] S. Lakhanpal, A. Gupta, and R. Agrawal, "On discovering most frequent research trends in a scientific discipline using a text mining technique", in *Proceedings of the 2014 ACM Southeast Regional Conference, ACM SE 2014*, ACM, 2014. DOI: 10.1145/2638404.2638528.

[16] S. Lakhanpal, A. Gupta, and R. Agrawal, "Towards Extracting Domains from Research Publications", in *Modern Artificial Intelligence and Cognitive Science Conference (MAICS) 2015, CEUR Workshop Proceedings*, vol. 1353, CEUR-WS, 2015, pp. 117–120.

[17] S. Lakhanpal, A. Gupta, and R. Agrawal, "Discover trending domains using fusion of supervised machine learning with natural language processing", in *Proceedings of 2015 18th International Conference on Information Fusion (Fusion)*, IEEE, 2015, pp. 893–900.

[18] S. Lakhanpal, A. Gupta, and R. Agrawal, "Mining Domain Similarity to Enhance Digital Indexing", in *Proceedings of 9th International Conference on Management of Digital EcoSystems, MEDES 2017*, ACM, 2017, pp. 88–92. DOI: 10.1145/3167020.3167033.

[19] M. R. Mann. (). Headlines, [Online]. Available: http://www.columbia.edu/itc/journalism/isaacs/client_edit/Headlines.html.

[20] A. Hertzmann. (2010). Writing Research Papers, [Online]. Available: http://www.dgp.toronto.edu/~hertzman/courses/gradSkills/2010/writing.pdf.

[21] BMC. (). Writing titles and abstracts, [Online]. Available: https://www.biomedcentral.com/getpublished/writing-resources/writing-titles-and-abstracts.

[22] P. Koopman. (1997). How to Write an Abstract, [Online]. Available: http://users.ece.cmu.edu/~koopman/essays/abstract.html.

[23] M Saravanan, S Raman, and B Ravindran, "A probabilistic approach to multi-document summarization for generating a tiled summary", in *Proceedings of Sixth International Conference on Computational Intelligence and Multimedia Applications (ICCIMA'05)*, IEEE, 2005.

[24] J. Goldstein, V. Mittal, J. Carbonell, and M. Kantrowitz, "Multi-document summarization by sentence extraction", in *Proceedings of the 2000 NAACL-ANLPWorkshop on Automatic summarization*, vol. 4, ACM, 2000, pp. 40–48. DOI: 10.3115/1117575.1117580.

[25] D. M. Blei, "Probabilistic Topic Models", *Communications of the ACM*, vol. 55, pp. 77–84, 4 2012. DOI: 10.1145/2133806.2133826.

[26] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet Allocation", *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[27] Z. Yin, L. Cao, J. Han, C. Zhai, and T. Huang, "Lpta: A Probabilistic Model for Latent Periodic Topic Analysis", in *Proceedings of the 2011 IEEE 11th International Conference on Data Mining*, IEEE, 2011. DOI: 10.1109/ICDM.2011.96.

[28] A. Hindle, M. W. Godfrey, and R. C. Holt, "What's hot and what's not: Windowed developer topic analysis", in *Proceedings of the 2009 IEEE International Conference on Software Maintenance*, IEEE, 2009. DOI: 10.1109/ICSM.2009.5306310.

[29] E. Chen. (2011). Introduction to Latent Dirichlet Allocation, [Online]. Available: http://blog.echen.me/2011/08/22/introduction-to-latent-dirichlet-allocation/.

[30] S.-d. Zhu, X.-h. Wu, and J.-p. Fan, "Analysis of Bulletin Board System Hot Topics Based on Multiple Keywords Combination", in *Proceedings of the 2011 International Conference on Management and Service Science*, IEEE, 2011. DOI: 10.1109/ICMSS.2011.5999066.

[31] K. Shubhankar, A. Singh, and V. Pudi, "A frequent keyword-set based algorithm for topic modeling and clustering of research papers", in *Proceedings of the 2011 3rd Conference on Data Mining and Optimization (DMO)*, IEEE, 2011, pp. 96–102. DOI: 10.1109/DMO.2011.5976511.

[32] K. Cai, S. Spangler, Y. Chen, and L. Zhang, "Leveraging Sentiment Analysis for Topic Detection", in *Proceedings of 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, IEEE, 2008, pp. 265–271. DOI: 10.1109/WIIAT.2008.188.

[33] S. Gupta and C. Manning, "Analyzing the Dynamics of Research by Extracting Key Aspects of Scientific Papers", in *Proceedings of 5th International Joint Conference on Natural Language Processing*, Asian Federation of Natural Language Processing, 2011, pp. 1–9.

[34] N. Schneider. (2017). Algorithm for HMMs, [Online]. Available: http://people.cs.georgetown.edu/nschneid/cosc272/f17/14_viterbi_slides.pdf.

[35] R. Xu, K. Supekar, Y. Huang, A. Das, and A. Garber, "Combining Text Classification and Hidden Markov Modeling Techniques for Structuring Randomized Clinical Trial Abstracts", *American Medical Informatics Association (AMIA) Annual Symposium*, vol. 2006, pp. 824–828, 2006.

[36] C. Boonthum, S. Toida, and I. Levinstein, "Preposition Senses: Generalized Disambiguation Model", in *International Conference on Computational Linguistics and Intelligent Text Processing 2006, Proceedings Lecture Notes in Computer Science (LNCS)*, Springer, 2006, pp. 196–207.

[37] V. Srikumar and D. Roth, "Modeling Semantic Relations Expressed by Prepositions", *Transactions of the Association for Computational Linguistics (ACL)*, vol. 1, pp. 231–242, 2013.

[38] C. Dictionary. (). Prepositional phrases, [Online]. Available: https://dictionary.cambridge.org/us/grammar/british-grammar/prepositional-phrases.

[39] A. T. Sherman. (1996). Some Advice on Writing a Technical Report, [Online]. Available: https://www.csee.umbc.edu/~sherman/Courses/documents/TR_how_to.html.

[40] (2013). A crash course in probability and Naïve Bayes classification, [Online]. Available: http://www.cs.colostate.edu/~cs545/fall13/dokuwiki/lib/exe/fetch.php?media=wiki:13_naive_bayes.pdf.

[41] A. Y. Ng and M. I. Jordan, "On discriminative vs. generative classifiers: A comparison of logistic regression and naïve bayes", in *Proceedings of the Advances in Neural Information Processing Systems 14 (NIPS 2001)*, 2001.

[42] C. Guestrin. (2007). Decision trees, [Online]. Available: http://www.cs.cmu.edu/~guestrin/Class/10701-S07/Slides/decisiontrees.pdf.

[43] A. Fernandez. (2014). Support Vector Machines, [Online]. Available: https://www.stat.berkeley.edu/~arturof/Teaching/EE127/Notes/support_vector_machines.pdf.

[44] S. E. Robertson and K. S. Jones, "Relevance weighting of search terms", *Journal of the American Society for Information Science*, vol. 27, pp. 129–146, 3 1976.

[45] ACM. (2012). The 2012 ACM Computing Classification System, [Online]. Available: http://www.acm.org/about/class/2012.

[46] P. Haase and J. Völker, "Ontology Learning and Reasoning - Dealing with Uncertainty and Inconsistency", in *Uncertainty Reasoning for the Semantic Web I. URSW 2006, URSW 2007, URSW 2005, Lecture Notes in Computer Science Proceedings*, vol. 5327, Springer, 2008, pp. 366–384.

[47] M. Yatskevich and F. Giunchiglia, "Element level semantic matching", in *Meaning Coordination and Negotiation Workshop, ISWC*, 2004.

[48] P. University. (2010). Wordnet: A Lexical Database for English, [Online]. Available: http://wordnet.princeton.edu.

[49] D. Jurafsky and J. H. Martin, *Speech and Language Processing*, 2nd ed. Prentice Hall, 2008.

[50] Z. Wu and M. Palmer, "Verb semantics and lexical selection", in *Proceedings of the 32nd Annual meeting on Association for Computational Linguistics (ACL)*, 1994, pp. 133–138.

[51] D. Lin, "An Information-Theoretic Definition of Similarity", in *Proceedings of the Fifteenth International Conference on Machine Learning (ICML)*, 1998, pp. 296–304.

[52] A. Kao and S. R. Poteet, Eds., *Natural Language Processing and Text Mining*, 1st ed. Springer-Verlag London, 2007.

[53] G. Erkan, "Language model-based document clustering using random walks", in *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL)*, 2006, pp. 479–486.

[54] X. Liu and W. B. Croft, "Cluster-based retrieval using language models", in *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR)*, 2004, pp. 186–193.

[55] W. S. Torgerson, "Multidimensional scaling: I. Theory and method", *Psychometrika*, vol. 17, pp. 401–419, 4 1952. DOI: 10.1007/BF02288916.

[56] J MacQueen, "Some methods for classification and analysis of multivariate observations", in *Proceedings of the 5-th Berkeley Symposium on Mathematical Statistics and Probability*, 1967, pp. 281–297.

[57] F. Sebastiani, "Machine learning in automated text categorization", *ACM Computing Surveys (CSUR)*, vol. 34, pp. 1–47, 1 2002. DOI: 10.1145/505282.505283.

[58] G. Berardi, A. Esuli, and F. Sebastiani, "Utility-Theoretic Ranking for Semiautomated Text Classification", *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 10, 1 2015. DOI: 10.1145/2742548.

[59] Z. Zheng, X. Wu, and R. Srihari, "Feature selection for text categorization on imbalanced data", in *ACM SIGKDD Explorations Newsletter - Special issue on learning from imbalanced datasets*, vol. 6, 2004, pp. 80–89.

A. Link to the Code and Readme Files

For my code and explanatory readme files, please refer to my repository on GitHub, at the following link: https://github.com/coder-sl/Dynamics-of-Research