Leveraging open source web resources to improve retrieval of low text content items

A THESIS
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Ayush Singhal

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

JAIDEEP SRIVASTAVA

August, 2014

Acknowledgements

There are many people to whom I feel grateful. First of all, I would like to thank my advisor Prof. Jaideep Srivastava for being such a nice and open-minded advisor. He always encouraged me to pursue my research interests. I am grateful to him for taking me as his PhD student even after one year of graduate studies. I also express my special gratitude to Prof. Vipin Kumar for giving me the opportunity to come to Univesity of Minnesota for my PhD. His advice about PhD and his understanding nature helped me so much in times of difficulties. I am grateful to him for that. I sincerely thanks Prof. Nikolaos Papanikolopoulos and Prof. Adam Rothman for spending their valuable time to serve on my thesis committee. I also thanks Prof Maria Gini and Prof. Abhishek Chandra for being in my preliminary exam committee.

The way I have developed during my PhD work and the strength to be able to face several challenges is a result of the wonderful teachings of Dr. Krishnan. I feel indebted to him for his constant efforts to encourage me to do this work with the right motivation– for the cause of education and welfare of the society. Because of him, today I feel that all the dynamics in my life have a clear purpose and I can be always joyful to pursue that purpose.

I am extremely grateful to Dr. Ankush Mittal, who introduced me to research during my undergraduate studies at IIT -Roorkee. He has given me several loving suggestions both before and during my PhD which I would like to use in time to come.

I am most delighted to express my loving feelings towards my dear friends Sanyam and Ashish. They have been like my brothers throughout my PhD work. I also thank Shashank, Ankit and Dr. Saket for being such nice friends.

I would like to share my appreciation for my family members, especially my mother, who cooperateed very favorably with me. She always encouraged me to fulfill the aim with which I have come to US. I am grateful for their love and sacrifices. I extend a loving thanks to my brothers Arpit and Kushagra for just being good brothers.

Dedication

To my teacher Dr. Krishnan,
who always keeps me up...
with his teachings, precept and encouragements.

Abstract

With the exponential increase in the amount of digital information in the world, search engines and recommendation systems have become the most convenient ways to find relevant information. As an example, the number of web pages on the world wide web was estimated to be over a trillion mark by the year 2008. However, search today is no longer limited to documents on the world wide web. The new "information needs" such as multimedia items (images, videos) opens up challenging avenues for scientific research. Thus the search techniques used to find items which are content rich (e.g documents) no longer holds for items with low-text content. In the literature, several solutions are proposed for developing search framework for multimedia item search which includes using the visual or audio content of such items for retrieval purposes. However, there is little research on this problem in the domain of scientific research artifacts. This thesis investigates the problem of retrieval of low-text content items for search and recommendation purposes and propose novel techniques to improve retrieval of such items. In particular, we focus on scientific research datasets owing to their importance and exponential growth in the last decade.

One of the main challenges in searching research datasets is the lack of text content or the meta-information about the datasets. While the datasets themselves have raw content, the problem of low text content makes the conventional text based search techniques inadequate for their retrieval. In comparison to the multimedia items, where visual and audio features have been utilized to enhance search or recommendation based retrieval, scientific research datasets lack a uniform schema for representing their raw content. Although solutions such as curation and annotation by experts/data scientists exists but these are infeasible for practical operation on a large scale. As a solution, this thesis provides a computational and an efficient framework for retrieving such low-text content items.

We primarily present two retrieval models, namely, (1) a user profile based search, and (2) keyword based search. For the user profile based search model, we show that the text content of the item can be derived from the user's profile and the relevance ranking can also be derived based on the information in the users profile. We find that the proposed approach which leverages the information from open source web resources for item extraction outperforms the local content based extraction approaches. For the keyword based search model, we have developed a

content rich database for research datasets. We use novel content generation techniques to overcome the low-text content challenge for datasets. The content information is extracted from open source and crowd sourced web resources like academic search engines and Wikipedia. In addition to the stand-alone quantitative assessment of the content generated, we evaluate the efficiency of the entire keyword based search framework via a user study. Based on user responses, the thesis reports positive evidence that the proposed search framework is better than the popular general purpose search engine for searching research datasets with a context based queries.

The ideas developed in this thesis are implemented in a real search tool- DataGopher.org: an open source search engine for scientific research datasets. Moreover, the approaches developed for research datasets can have application to other low-content items such as short text document, news feeds, Twitter and Facebook postings. In summary, the computational approaches proposed in this work advance the state-of-the-art in retrieval of low-text content items. Whereas the extensive evaluations that are performed on items like scientific research datasets and low text content documents demonstrate the validity of the findings.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

An information retrieval system will tend not to be used whenever it is more painful and troublesome for a customer to have information than for him not to have it.
Calvin N. Mooers

1.1 Information retrieval

The term, Information retrieval (IR), has been used since 1950s to refer to automated approaches to retrieve information (documents) from a small or large collection of information. Since IR's early beginning, the systems have evolved to accommodate various needs and styles of different users. Reviewing the history of evolution of IR systems is therefore very interesting. However, before delving into the history of their evolution, the following definitions and concepts will be helpful in understanding the various advances in the IR research that have happened over the decades.

1.1.1 Basic definitions and concepts

In figure 1.1, we show a schematic of a basic IR system. Given a query (by the user), the main function of the IR system is to find the most relevant item from the database over which the search is done.

User query It is one form of input from the user. It is an expression of his information need. In general, a user query is understood to be a set of unstructured or structured form of text input

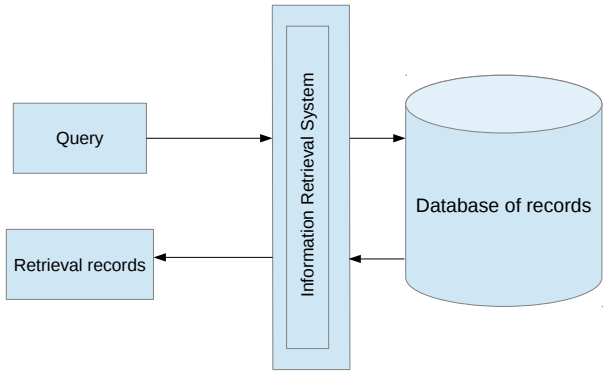


Figure 1.1: A schematic of a basic IR system.

provided by the user in a natural language or the syntactical language of the IR system. Most often this is understood as a active input from the user to the system. However, in our work, we use the term user query to refer to both the active inputs (in form of keywords) and to passive inputs (derived from user's profile by the system).

Indexable database Indexing is a way to make the search in the database efficient. The documents (items) stored in the database are retrieved based on these indexes. Instead of scanning the full content of the item which is both computation and cost intensive, indexes help to reduce the search time as well as the storage. One of the simplest way to understand is through a simple diagram shown in figure 1.2. There is a common index file consisting of unique list of the terms in all the documents (stop-words are filtered out from the indexes in most IR systems). Any search input is matched with the indexes to retrieve all the documents containing the index term. There are further complex forms of indexes but we do not discuss their details here.

Some other well known definitions and concepts in the IR literature are described below.

Vector models These corresponds to algebraic models used to represent text documents or a query. A document is represented in N-dimension where each dimension corresponds to a

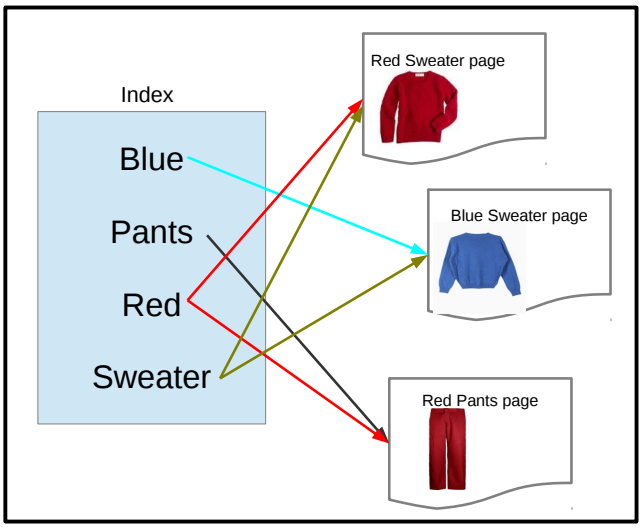


Figure 1.2: A simple example to explain indexing of documents.

term (text token) in the corpora of documents. Once the document is converted into a vector representation, several algebraic operations can be performed like computing document similarity using cosine coefficient. In the document vector, the terms appearing in the document are given weights depending on their frequency in the document. There are several weighing schemes used in the literature. tf*idf is one of the most popular weighting scheme used in IR technologies.

term-frequency (tf): the weight of the terms in the document vector is equal to the frequency count of the term in the document.

inverse document frequency (idf): the weight of the terms in the document vector is inversely proportional to the co-occurrence of the terms across documents in the corpora. This scheme is based on the concept that more commonly appearing terms should be given lesser weight in retrieval.

*tf*idf:* This scheme is a combination of the term frequency and the inverse term frequency in a manner that the final score is computed as a product of the tf and the logarithm of the idf score of each term. There were several variants of it that were proposed in the literature.

Document tagging/classification In several IR systems, one of the commons techniques used is assigning keywords or key-phrases to the documents. Since the full content of the document

will increase the index size, tagging the documents with relevant and important keywords reduces the search time by filtering the documents based on small indexes constructed using only the keywords (document tags). This is sometimes referred to as document classification as well. In the literature, both supervised and unsupervised approaches are used to classify documents.

1.1.2 A brief history

Today, the advent of the Internet has expanded the use of IR systems to practically all the fields in our life. The Internet technology has connected all the bits and pieces of information across the globe into a single "global database". However, finding relevant information from the huge web corpus is practically an impossible task at the cost of human efforts. At this end, the modern concept of search engines and recommendation systems act as rescuers. While such systems have capabilities to index the billions of resources available in the web corpus, yet the task of finding the most relevant result for a user's information need is not less than finding a 'needle in a haystack' or even worse.

While the word 'Google' makes us feel-"just search Google" for your information need, such intelligent systems have been evolving from very simple yet novel IR techniques. Originally (ignoring the pre-history of mechanical and electro mechanical device), the IR systems were used to search data (documents) locally within the computer[4]. The IR systems were just beginning to emerge with the concept of indexing documents and retrieving them. Soon the technology became popularized for library databases. Based on Cleverdon's experiments[5] of comparing Taube's Uniterm system[6] with conventional document classification techniques, the word based indexing in IR became popularized for academic search.

The IR systems soon moved out of the boolean retrieval framework to a rank retrieval framework by introducing the concept of term frequency[7]. Other major improvements happened in IR systems in the '60s when the documents and queries were viewed in the N dimensional space[8] and similarity was compared using cosine coefficient[9]. The feature of relevance feedback (supporting iterative search to improve the quality of search results) was added to the IR technology by the popular work of Rocchio[10]. Following this, the concept of term frequency[7] was extended to incorporate inverse-term frequency to formalize a new weight tf*idf [11].

Building on the concept of tf*idf, Robertson et al[12] introduced BM25, a ranking function for the retrieved results. In the vector space models, the advancement was made by using Latent

semantic indexing (LSI) for dimensionality reduction of document vectors. So even before the world wide web was developed, there was significant advancement in the IR technologies in the areas of indexing, ranking and searching.

After the invention of web in late 1990, the growth of web documents was exponential and several practical challenges emerged which required innovation of the conventional IR techniques. Some of the major landmarks for IR techniques related to web search are PageRank[13] and HITS by Klienberg[14]. These were then used for improving the ranking function. So the search systems like Google and Bing, as we see now, leverage several features of the web. For example, the use of query logs is done to understand user's intent, such as automated spell correction[15], automated query expansion[16] and more accurate stemming[17]. Moreover, advances in search systems to make them more personalized for users is based on understanding their usage patterns and profile. In this light, the modern recommendation systems have covered most of the spectrum of a user's information needs, ranging from books to clothes to songs. Such systems work in a passive way i.e. the items are recommended to the user automatically without expressing explicit information need.

1.2 Research challenges and thesis problem

As described in the previous section, the early growth in the IR technologies was primarily driven by the lack of efficient techniques to satisfy general retrieval needs. However, the advent of the web opened up the research directions for extending IR systems to satisfy various kind of information needs of the users such as personalized search and recommendations. As a result, the IR technologies are not only limited to text documents. Specialized search engines like Google images for images, Youtube for videos and Amazon for daily purpose items, to name a few, have already been successful research products in the real world. The IR approaches available for full text documents, were not applicable for non-text based items like images, videos, songs, daily use products and other non-text items. Several research ideas from the field of image processing, data mining and social networks were utilized to overcome challenges in retrieval of such items. However, the availability of the newer web resources like scientific research datasets, short text documents like twitter feeds, news clippings, product reviews etc have led to a further expansion in the information needs of the users. Unlike the full-text documents and the multi-media items like images and videos, these new web resources pose newer

research challenges which are not addressable by the state of art in IR. Enlisted below are the research challenges we identified for these new web resources.

1. Web resources like scientific research datasets and short text documents have very low text content. Low text content puts these resources in the long dark tail of the web corpus.

2. Scientific research datasets lack a uniform schema for representing their raw content. Thus the multimedia retrieval approaches which exploit item's structural information such as visual features are no longer applicable.

3. Identifying relevant keywords for short text documents becomes challenging due to low linguistic and structural information.

4. Information need for scientific research datasets is generally expressed as a context based query[1] . Hence, the text information describing the content of the datasets is not the best way to index these resources.

In this thesis, we categorize the newer web resources such as scientific research datasets and short text documents as low text content items. Given the above mentioned challenges with the retrieval of such items, the aim of this dissertation research is to develop algorithms and search frameworks for enhancing retrieval of such low text content items in a manner that satisfy the information need of the user of this resources.

1.3 Overview of the thesis

Figure 1.3 gives the big picture overview of this thesis. As shown in the figure, by leveraging the information from open source web resources to generate relevant content for low text content items, we get efficient and superior retrieval (search) performances.

1.3.1 Organization of the thesis

The thesis is organized as follows:

- Chapter 1: Introduction

[1] Based on a user survey

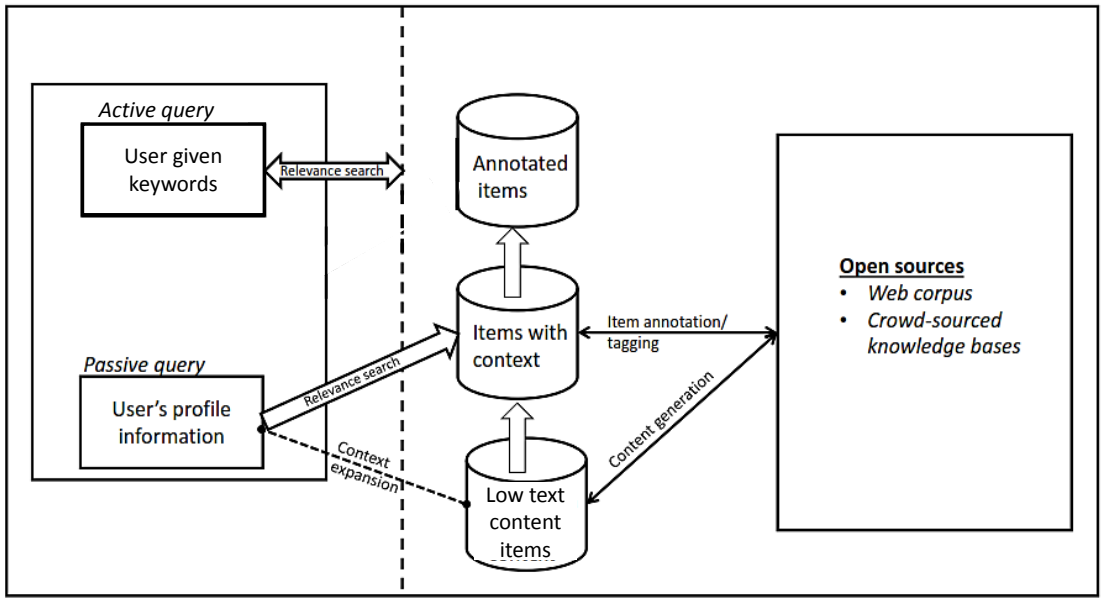


Figure 1.3: **Thesis diagram**

It gives a general introduction to the thesis, the basics of Information retrieval, a brief history of IR, challenges in current IR systems for new information needs and the thesis research problem.

- Chapter 2: Leveraging user profile to enhance search of low-content items

  This chapter describes a dataset search approach based on user's profile information. This chapter provides understanding about how the low text content items can be retrieved by expanding context from the user's profile.

- Chapter 3: Automating keyword assignment to short text documents

  In this chapter, we demonstrate the use of information derived from open source web resources to assign relevant keywords to short text documents.

- Chapter 4: Automating annotation for research datasets

  In this chapter, we propose a framework for generating structured annotation for scientific research datasets.

- Chapter 5: Annotation expansion for low text content items

It is an extension of the previous chapter. In this chapter, we discuss an approach to automatically expand the keywords assigned to low text content items.

- Chapter 6: A keyword based search for research datasets

  In this chapter, we discuss a keyword based search model for searching research datasets. This chapter provides insight about how low content objects can be made searchable using the approaches discussed in previous chapters.

- Chapter 7: Conclusions:

  We finally summarize our efforts in developing algorithms for enhancing retrieval of low text content items based on leveraging open source web resources. We give some directions for potential future extension of this thesis.

Chapter 2

Leveraging user profile to enhance search of low-content items

Our day to day needs ranging from shopping items, books, news articles, songs, movies, research documents and other basic things have flooded several data-ware houses and databases both in volume and variety. To this end, intelligent recommendation systems and powerful search engines offer users a very helpful hand. The popularity and usefulness of such systems owes to their capability to manifest convenient information from a practically infinite storehouse. As an example, the number of web pages on the world wide web is estimated to be over a trillion mark [18] by year 2008. Thus recommendation systems such as Amazon, Netflix and similar others take initiative to know user's interest and inform users about the items of their interest. Although these systems differ from each other according to the application they are

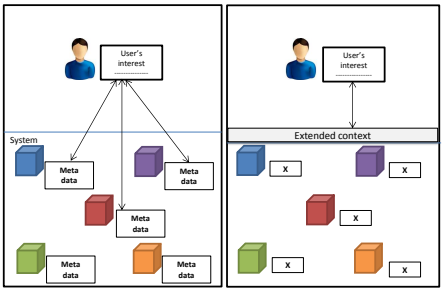


Figure 2.1: A comparison of interest to item matching (a) when the meta-data information about the items is available vs. (b)when it has to be derived by context extension.

used for, the core mechanism of finding items of user's interest is that of user's interest to item matching.

While tremendous progress has been made in developing techniques for perfecting interest to item matching [19], most of these approaches assume that the meta-data information (e.g. description of item properties, rating for the item) about the item of interest is available to the system. In this work we study the interest to item matching problem for a case when we have no meta-data information available about the item of interest. This scenario is shown in figure 2.1. As shown in the figure, the scenario on left shows the common interest-item matching framework, whereas, the scenario on the right shows the missing meta-data problem. We study this problem in the context of finding research datasets of interest for a user.

Research datasets, like any other daily-need item, have become an important part of research for data scientists. At present, any theory or algorithm holds higher value if it can be validated on some real world datasets. Such expectations in research are realistic since technology has made data collection simpler than ever before. However, given the infinite variety of datasets available, a common problem that several data mining researchers, especially working in inter-disciplinary areas, face is to identify the most relevant dataset for their research problem. Moreover, unlike other items which generally have a common database source, items like datasets do not yet have a single common repository. For this purpose, some attempts such as UCI machine learning repository [20], Stanford graph data [21] and few other open source repositories have helped researchers. the use of such sources for an automated search or a recommendation system.

In this chapter, we propose an algorithmic approach to handle the situation of missing meta-data information about the item. Given the user's interest, we learn the context of the item by extending the context around user's interest using an external database. Datasets are identified from the context using web intelligence from search engines and online thesaurus. Finally, we model the ranking of the datasets to maximize the accuracy of the dataset to be recommended. We have tested the performance of the proposed framework on user's queries generated from real world dataset. We have compared the performance of the proposed approach for dataset finding with the state-of-art supervised classification approach [22] on the DBLP corpus. The proposed approach gives us a recall improvement of about 36% over the baseline approach. Using the overall framework we are able to answer 90% of the user queries correctly in the top-4 recommendations.

2.1 Background and Problem Statement

The problem of interest to item matching can be mathematically described in the following manner. Given a user's interest (I) and a list of items (D), the problem is to rank the items in D relevant to user's interest (I).

However, here, we are looking at a slightly complex scenario where the list of items is not explicitly available. So, we are given the user's interest I but the list of items is not available. The only thing known about the item is its general category g. Here, we are interested in finding datasets which are of interest for a given user.

2.2 Related Work

There are three broad areas of research, namely, techniques/algorithms developed for item to interest matching, objective of maximization and the application of this problem in different domains.

From the algorithmic perspective the two main streams of data mining techniques extensively studied in the literature are collaborative-filtering based and content based techniques. In order to recommend relevant items to the user, the collaborative filtering based techniques use information from the social network of the user. In such approaches, the main task is to recommend items from the items of interest of similar users[23]. Within collaborative filtering based methods there are three main categories of collaborative filtering techniques: memory based, model based and hybrid CF algorithms[24].

The content based techniques used in interest to item matching problem use content from the user's profile or the item profile to create a suitable match. These techniques can further be categorized in two broad categories namely, keyword based and semantic analysis based. The keyword based are more closely related to information retrieval literature and a lot of research work exists for this [25]. Recently, to overcome the limitation of keyword based matching, semantic analysis based techniques were developed to match content of user and the item. There are two major research directions in the semantic analysis based approaches: ontology based semantic analysis [26]and tag based recommendation[27]. As a further addition to the semantic based matching techniques, a few works have used extrinsic knowledge sources such as Wikipedia to improve content matching[19, 28].

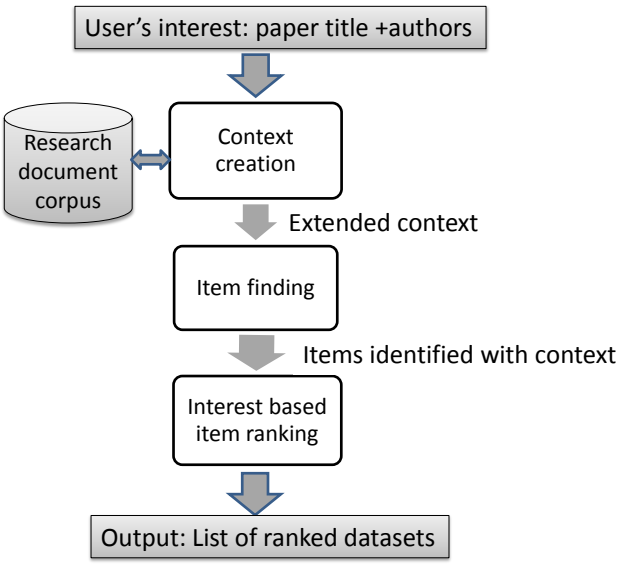


Figure 2.2: A framework for the proposed approach.

In terms of selecting the objective function for interest-item matching, the commonly investigated measures are accuracy and novelty. Recently, the interest of the community has extended to consider serendipity (surprise) as a new objective of recommendations [29].

The next spectrum of recommendation system literature deals with the application of recommendation system. The recommendation systems penetrate in various aspects of human life. Some of the most popular recommendation application can be found in the field of web keyword recommendation [30], new article recommendation [31], music/song recommendation [32], movie recommendation [33], book recommendation [34], item recommendation [35, 36], research article recommendation [37, 38, 39],citation recommendation[40, 41, 42], tag recommendation [43, 44].

2.3 Proposed Approach

In this section, we describe the proposed approach for identifying dataset. As explained in the previous section, the main challenge here is that the metadata information about the items of interest is not available. In order to fill this gap of lack of meta data for items of interest we

propose a three step approach (as enumerated below and in figure 3.5). The proposed approach basically derives meta data information about the items of interest by extending the user's given interest (I). The following section describe the proposed steps.

2.3.1 Context creation from user's interest

As mentioned earlier, the main task here is to extend the context of user's interest so that it can be used as meta-data for the items for recommendation. Here we assume that the user's interest I is denoted by a research document. So the basic ingredients of user's interest I are- topic of the research document, the abstract summary of the research document and the author names of the research document. Given, I=$\{$document topic, abstract summary, authors of document$\}$, the context is created by using an external corpus C consisting of research documents. We have extended the context by finding documents which are related to user's interest. Document relatedness is measured in the following manner:

1. *Content based similarity:* The content based similarity is the standard approach of comparing similarity(or relatedness) between any two documents. In order to compare the documents in the corpus C with the document topic and abstract summary in user's interest I, we have used the TF-IDF model to create tf-idf vector representation of each document. Standard natural language preprocessing such as stop-word removal, special character removal are done as the first step. The similarity comparison between the tf-idf vectors of two documents is done using *cosine similarity* metric. The documents in the research corpus are then ranked in order of their cosine similarity with user's interest. A document with lower cosine similarity score gets a higher rank.

2. *Author based similarity*: The other information available about the user's interest is the names of the authors on the document of his interest. This information is also used to extend the context of user's interest. In the content based similarity we used the content information of the user's interest and using a corpus of research documents, the documents in the corpus were ranked. In the author based similarity, each document in the corpus C are ranked based on the minimum normalized google distance [45] between authors' information in user's interest and the authors of the documents in the corpus C. The documents in the corpus are ranked using the following metric:

$$sim(d_I, d_j) = min(NGD(A_k^I, A_l^j))$$

where, dI is the document in user's interest

d_j is a document in the corpus

A_k^I denotes the k^{th} author in the user's interest I

A_l^j denotes the l^{th} author of the j^{th} document in the corpus

NGD stands for Normalized Google distance

Normalized Google distance between two words is defined in the following manner:

$$NGD(x,y) = \frac{max\{logf(x),logf(y)\} - logf(x,y)}{logM - min\{logf(x),logf(y)\}}$$

where M is the total number of web pages searched by the search engine; $f(x)$ and $f(y)$ are the number of hits for search terms $x$and y, respectively; and $f(x,y)$ is the number of web pages on which both x and y occur.

If the two search terms x and y never occur together on the same web page, but do occur separately, the normalized Google distance between them is infinite. If both terms always occur together, their NGD is zero, or equivalent to the coefficient between x squared and y squared [45].

We have extended the use of NGD to find semantic relatedness between authors of research documents. The main motivation to use semantic metric to find relatedness between two entities is two folds. Firstly, search engines like Microsoft academic search and other academic search engines provide access to the largest and the most updated corpus of research documents while the curated sources such as DBLP [46] may not provide the complete information of author's network. Secondly, the NGD provides a computationally cheaper and a simple way to find relatedness between two authors. We have used the Microsoft academic search engine to obtain the query output for author names.

As explained above, context extension for user's interest is done by ranking research documents in a corpus C. We described two different ranking approaches- one using only the content information from user's interest I and the second uses the author information from user's interest I. In the next step, a single metric to rank the documents in the corpus based on the full information of user's interest is obtained. In order to accomplish this, the two ranks for each document in the corpus are aggregated using a standard rank aggregation approach given by Borda in 1970 [47].

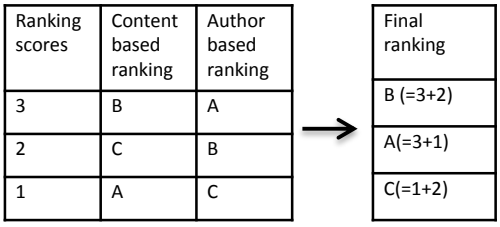


Figure 2.3: A simple illustration to show rank aggregation using Borda count aggregation technique.

The Borda rank aggregation technique is explained in figure 2.3. The figure illustrates the rank aggregation mechanism using a simple example. Here the final rank for document 'A' is obtained by adding the two scores given to 'A' by the content based similarity and the author based similarity. As shown in the figure, the final score for 'A' is 4, for 'B' it is '5' and for 'C' it is 3. Hence the final order of the documents after aggregation is B $>$ A $>$ C. The rank aggregation approach is helpful to account both for the content based ranking and the author similarity based ranking.

At the end of this step, an extended context for user's interest is created using a corpus of research documents C. The documents which are more relevant for user's interest are ranked higher in the corpus.

2.3.2 Item finding

As mentioned earlier, the context or the meta-data information around the item of interest (here items are datasets) is not explicitly available. Thus the context creation is done by extending the user's interest using an external research corpus. In this step we discuss an algorithmic approach for item identification from the extended context of user's interest. The approach is based on identification of the item of a particular category (here datasets).

Before generating a candidate set of items there are two natural language processing techniques used. The approach is detailed in the figure 2.4. As shown in the figure, at the first step relevant section of the document is extracted from the document. It is assumed that dataset (item of interest) description appears in certain well defined section of the document. Standard parsing techniques are used to extract sections which described the experimental setup. Details of this parsing is explained in the dataset preparation Section 2.4.1. Then standard natural language processing techniques such as stop word removal and special character removal are

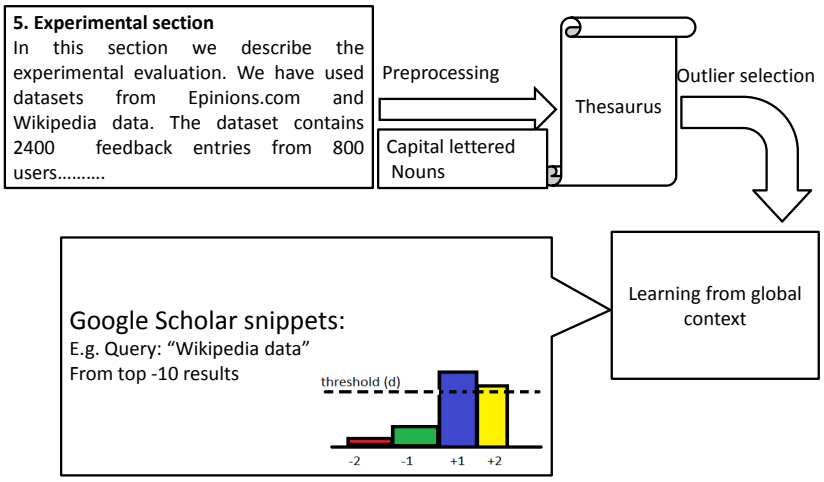


Figure 2.4: A pictorial representation of the dataset extraction algorithm.

applied. Next, using the property of item of interest- whether it is a noun or verb, firstly, candidate words with the same property are extracted. So in the present case, only words which are noun are extracted as candidate dataset names. Secondly, using the property of name nouns that they start with capital letter, only nouns which start with capital letters are considered as candidates for dataset names.

At the next step, the candidate item names are evaluated against the knowledge base of a thesaurus. Thesaurus is a source of world English knowledge and contains words in their different form with their Standard English usage. Thus this knowledge base helps identify from the candidate names, the ones which are not used in common English language practice. This step is called as the *outlier selection* step, where the outliers of the knowledge base are considered as the new set of candidates. The idea behind using this approach for pruning Standard English words is based on subjective observation of the names of datasets used in research documents.

At the final step, dataset names are identified from the candidate set using the "global context" generated using web intelligence from the search engines. The context mining is done from the text in the *snippets* which are output of any query given to a search engine. Here we have used the Google scholar search engine to collect snippet information for a query. The

query (example shown in figure) is a combination of the candidate term and the term "data". Given a query of the form [d_i data], the search engine returns top-k results for the query. In addition to the document titles, search engines such as Google scholar provide information about context from the document where the terms appear and these are called the snippets. We use the content information from both the title and the snippets in the top-10 results to identify the frequency of the candidate term appearing adjacently to the term 'data'. We look at three forms of adjacency, namely, 1^{st} left neighbor, 2^{nd} left neighbor, 1^{st} right neighbor, 2^{nd} right neighbor and get the frequency of each positioning from the top-10 results. This gives a frequency distribution of various positioning at which the item category (data) appear from the candidate word (dataset name). However, out of all the positioning at which one expect the item name and its category to appear, the position of 1^{st} right neighbor is the most important one. Thus the final dataset names are identified based on the 1^{st} right neighbor frequency of the candidate name with the term data. The threshold is determined by optimizing the f_1-measure value using the precision and recall variation over different values of the threshold (d). This will be discussed later in the experiments in section 2.4.1.

2.3.3 Item ranking

An important step after identification of items from the extended user's interest is to rank the items based on user's interest. For each document in the corpus (ranked based on step 1), items were identified using step 2. However, the same item can be identified in multiple documents making ranking of items a non-trivial task. In order to rank items, we have modeled the frequency and the popularity of the items using the following function:

$$R(D_i) = \Sigma exp^{(-x_j)}$$

where, $R(D_i)$ is the rank for dataset d_i and x is the rank of the document d_j in which D_i is used.

An exponentially decaying fucntion for ranking has the following advantages. Firstly, it increases the score of datasets (or item) if they are used in documents which are highly ranked in the extended context for the user. Secondly, the scores of a dataset increases if it is used frequently.

2.4 Experimental Analysis

In this section we discuss in details the experimental design and the dataset used for evaluating the proposed approach. There are two set of experiments performed to evaluate the proposed approach. The first set of experiments are conducted to evaluate the performance of the item finding algorithm. The second set of experiments are conducted to evaluate the performance of the overall item ranking approch. This section is sub divided into two separate sections.

2.4.1 Experimental analysis for item finding algorithm

In this section we discuss the experiment design and the dataset used for evaluating the item finding algorithm.

Dataset description

For the purpose of evaluating the proposed approach we have used 400 research documents from the DBLP bibliography corpus. We have used papers which are published in important data mining venues like KDD, ICDM, CIKM, WWW. The full research documents were obtained in their pdf version from the web and then converted to text format for subsequent parsing and extraction. The relevant sections like experimental sections or dataset description section were extracted by text parsing. We used the root terms like 'Experiment','Analysis','Evaluation','Data' to identify and extract sections of interest from the text files of the research documents. The extract of relevant sections from these 400 documents is the input for the item finding algorithm.

Ground truth

The ground truth for the dataset names were extracted from the 400 research document by manual labelling. Each document was marked with the dataset used in that document name and finally all the dataset names were collected together.

Baseline

The baseline used in this work is the state -of art approach for dataset extraction proposed by Lu et al [22]. They use a supervised classification based approach to identify terms which denote dataset used in the research document. They have used the structural information of the

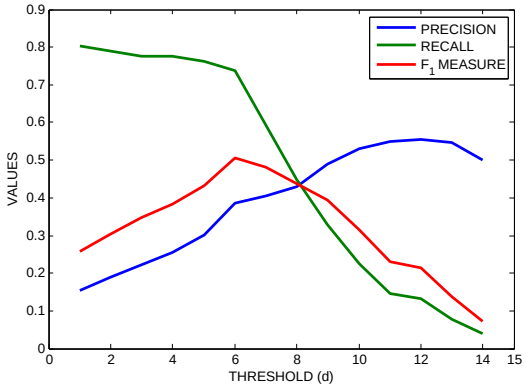


Figure 2.5: Plot showing the variation of precision, recall and F_1 measure with the threshold (d) for dataset names in google scholar snippet.

sentence around each word to create its local context. We have used a neighborhood of 5 words for each terms that is considered for classification. The results for this approach were obtained by 10-fold cross validation technique using a random forest decision tree.

Evaluation metrics

We have used the standard performance evaluation metrics such as precision, recall and F_1 measure to evaluate and compare the performance of the proposed approach with the baseline approach. In the figure 2.5 we show the variation of precision, recall and F_1 measure with the threshold value (d) [explained in Section 2.3.2]. The optimum value of the threshold is determined by the optimizing the F_1 measure value. As shown in the figure 2.5, the F_1 measure is highest at a threshold value of 6. Thus using a threshold value of 6 for the frequency of the 1^{st} right neighbor we can determine 74% of the total correct datasets.

We compare the performance of our approach with the baseline. As shown in the table 2.1, the proposed approach gives a significant improvement in terms of recall. The recall jumps to 74% in comparison to 38% using the baseline approach which uses only the local context for identifying dataset names. In terms of precision, the baseline approach seems to perform well. However, high precision is inherently due to the class imbalance problem in a classification setting. The number of instances identified in the minority class are already very less which tend to favor the precision. However it should be noted that recall is more important than precision in the following scenario because it is more important to identify items which have to

Table 2.1: Comparison of precision, recall and f_1 measure values for the proposed item finding approach and the Lu et al approach.

	Precision	Recall	F_1 measure
Proposed approach	0.39	0.74	0.51
Baseline	0.52	0.38	0.44

be recommended.

Owing to the higher recall, we also get a 7% improvement in the F_1 measure. These results show that the proposed approach using global context from search engines and using world knowledge base such as thesaurus is more appropriate for finding dataset names, used in computer science research, than the baseline approach. Next, we discuss the experimental setup for validating the performance of the overall framework.

2.4.2 Experimental analysis for overall framework

In this section we discuss the experimental design and the dataset used for evaluating the overall framework for finding interesting datasets for a user.

Dataset description

As mentioned earlier that the context creation for user's interest is done using an external corpus of research papers. For this purpose, we have used a corpus consisting of 9000 research document from top-tier data mining forums. We have only considered documents which were published between 2001 and 2010. The metadata information about these research articles was available from the DBLP bibliography corpus [46]. On the user's side, we have used 20 test queries denoting interest of 20 users. This data consists of information about the document title, its summary abstract and the authors of the document. The evaluation was done for these 20 test queries. The 20 test queries consists of research documents which were published in the year 2010. We have used research documents from year 2010 for our test query inorder to capture the prediction capability of the proposed approach for identifying dataset name which were actually used in research later.

Ground truth

The basic idea of the experiment is to see if the using the proposed approach for interest to item matching can find datasets of relevance for the user's interest. For the purpose of testing, we have considered 20 test queries and the ground truth is the actual dataset which was used in the document that is entered as query. Since there can be more than one datasets which are used in a single research document, we consider all of them in our ground truth.

Baseline

To the best of our knowledge, not much work has been done in the field of identifying interesting datasets for a user. So we evaluate the strength of adding author based similarity to improve the context of user's interest in comparison to the standard content similarity based ranking. The purpose of the experiments is to see if aggregating ranks obtained from author similarity improves the context creation for user's interest and thus find relevant datasets for the user.

Evaluation metrics

In order to compare the relevancy of the dataset recommended by the proposed approach and the baseline approach, we have used two evaluation criteria.

Recall@k (R@k): The recall@k is defined as the ratio between the original datasets that appear in the top k recommendations for user's query. The recall is averaged for all the user queries. This metrics captures the exact match for the ground truth dataset and the datasets recommended in the top-k.

Co-usage probability (CUP): It may so happen that the dataset in some top-k recommendation are relevant but not exact. The recall measure cannot capture the relevancy of the dataset in case there is no exact match. In the literature, user studies have been used to capture relevance of some recommendation. However, we use wisdom of population from search engines as the ground truth to define a systematic metric for the purpose of evaluation. A similar metric was used for evaluating bibliography recommendation [41]. This metric essentially captures the probability of co-usage of the original datasets and the recommended datasets. For each pair of datasets $< d_o, d_r >$ where d_o is the original dataset used and the d_r is the recommended one, we calculate the probability (CUP) that these two datasets have been co-used in the past as :

$$CUP = \frac{\text{hits}(d_o,d_r)}{\text{hits}(d_o)} \text{ in academic search engine}$$

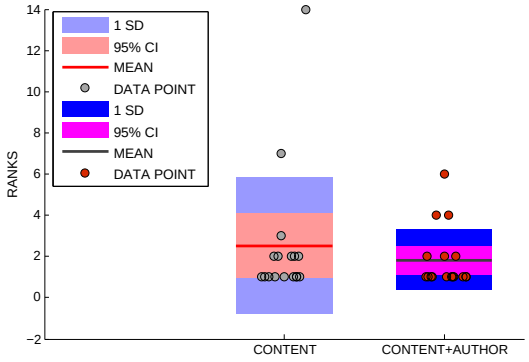


Figure 2.6: The box plot shows the distribution of ranks for datasets for the test queries.

The counts of dataset is obtained using the exact phrase matching capability of search engines. We have used the Google scholar search engine to find the exact count when a dataset d_o appears in research documents and how many times d_r appear together with d_o. For example, a query such as "Epinions data" gives the count of documents in which "Epinions" and "data" appeared adjacent. The same can be done to check if two datasets were together referred as data in some documents.

2.5 Experimental results and discussion

We evaluated the proposed approach (which uses both the content and author information for context creation) with the baseline (which uses only the content information for context creation). In the figure 2.6, we have compared the distribution of ranks at which the exact dataset was recommended for the 20 test queries. In other words, the ranks corresponds to the position of the original dataset (the ground truth of the 20 test queries) in the top-k recommendation of the proposed and the baseline approaches. The box-plot shows that the mean rank using the proposed approach is lower (1.8) in comparison to the baseline approach(2.5). This means that on an average the correct datasets were identified within top 1.8 of the recommendations. From the figure 2.6, we can also notice from the spread that the variation of ranks in case of baseline approach is higher than in the case of the proposed approach. The dots in the plot corresponds to individual results for the 20 test queries.

In the figure 2.7, we show the variation of recall with k. The plot shows that the recall@2

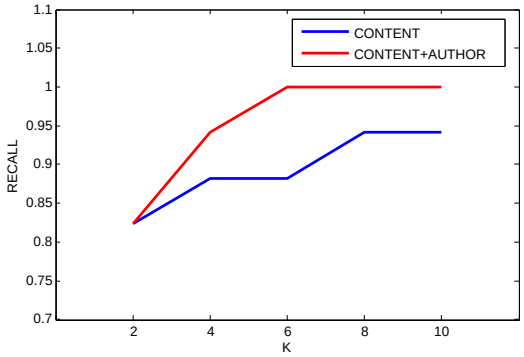


Figure 2.7: Plot showing variation of recall in the top-k dataset recommendation. The values of k are varied from 2 to 10

is nearly 83% for both baseline and proposed approach. However, the proposed approach (Content+Author) outperforms the baseline approach (Content) at recall@4 by about 10%. The recall@k further improves as k increases. From this plot, we observe that by adding author similarity for context creation the recall in top-4 is significantly improved.

Next, we evaluate the performance of the baseline with the proposed approach using the CUP criteria. The CUP score is averaged for all the 20 test queries. In the figure 2.8, we show the variation of the CUP score in the top-k recommendation. The CUP score is computed after eliminating the datasets which were exact match in the ground truth. Thus the CUP score for top-1 means the first dataset in the recommendation that was not the exact match. Higher CUP score means higher probability that the dataset in top-k was used with the original dataset in the ground truth. The plot shows that appending author based ranking with the content based ranking improves the probability of finding dataset related to the ground truth dataset. The decreasing trend in the CUP score signify that the probability to find related datasets is higher in higher ranked recommendations than in lower ranked recommendations.

In summary, adding author information in ranking help to improve the context for user's interest. The proposed approach shows improvement both in terms of recall@k and the CUP score.

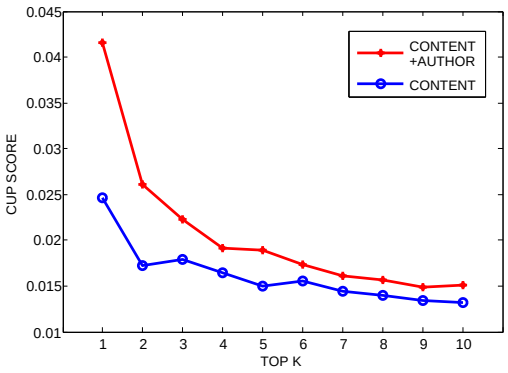


Figure 2.8: Plot to compare the co-usage probability (CUP) score in the top-k datasaet recommendation.

2.6 Conclusions and Future Work

In this chapter, we proposed an algorithmic approach to enahnce retrieval of research datasets using information from user profile. We also proposed an unsupervised approach using web intelligence for finding dataset names. The performance of the dataset finding algorithm was experimentally compared with the state of art. We also performed a comparative evaluation of the two context generating approaches (only content base vs. content+author based). The proposed approach for dataset finding gives significant improvement in the recall ($74\% vs 38\%$). For the overall framework, we found that adding author based ranking improves the recall in top-4 recommendation results by 10%. We also see that the co-usage probability of the datasets in the recommended list was higher when context was generated by including the author information.

Chapter 3

Automating keyword assignment to short text documents

Assigning keywords (or subjects) to documents is extremely important for several practical applications like search engines, indexing of databases of research documents, comparing similarity of documents, document categorization or classification and ontology creation and mapping, to name a few. While this is important for every information retrieval practices in general, the act of classifying or summarizing documents with keywords (index terms) has become very important in maintenance of bibliographic databases. As an example, the number of scholarly documents available on the web is estimated to be 114 million English language scholarly documents[48]. Given the size of the database, several academic indexing services such as Microsoft Academic search[1] , Pubmed[2] and MNCAT[3] (to name a few) are using keywords to index the documents rather than the full text because of several issues with full text indexing[49]. The index terms are mostly assigned by experts but in several cases author keywords are also used.

Although keyword assignment is a well studied problem in the field of information sciences and digital libraries, there are several practical challenges that cannot be addressed with the state of art techniques for automating keyword assignment. The current approaches for keyword extraction leverage the local content information of the documents and identify keywords

[1] http://academic.research.microsoft.com/
[2] http://www.ncbi.nlm.nih.gov/pubmed
[3] https://www.lib.umn.edu/

from it. These approaches are suitable for the documents with enough text content since the structural, positional and linguistic information can be leveraged to determine certain words as keywords for the document. In the literature, these approaches are broadly classified into two sub-categories: (1) keyword extraction and (2) keyword/ key phrase abstraction. Supervised [50, 51] and unsupervised approaches[1, 2] for keywords extraction use the local text content of the document. These approaches, at best, can assign keywords from only those present in the content. Other keywords which provide better description of the document will be completely missed out. While the keyword abstraction approaches mainly deal with keyword selection from an external collection of keywords based on their relevance to documents' content. The keywords, therefore, need not necessarily overlap with the text in the document[52]. Gabrilovich et. al.[53] proposed an innovative approach for document categorization which uses Wikipedia knowledge base to overcome the limitation of generating category terms which are not present in the documents. However, this approach uses the entire content of the document and extend the context using Wikipedia. While, the proposed approaches are useful to assign keywords to full text documents, but they are not necessarily applicable to documents with low text content. By low text content, we mean the documents with length ranging from anything between 5 to 100 words. Scientific publications which contain only a small abstract, documents over the web to which full-text access is withheld, social snippets from media such as Facebook[4] and Twitter[5] , short news texts and product reviews on online stores are a few examples of items with low text content. Keyword assignment to such low content document is very important for the purpose of information management as illustrated in the following few examples.

Example 1: Keyword assignment for items with 5-10 text words: Sometimes in conference submission portals, the number of manuscript submissions are exceedingly high. In general authors are provided an option to categorize their document at the time of submission. Based on personal experiences of several researchers, category selection for the manuscript is a confusing process due to conceptual overlaps among several categories. There are several limitations with this approach. Given the evolving nature of various research topics, category selection for a document can be more appropriately done with the global perspective of various topics instead of the local perspective of the author. Another challenge from the side of the conference organizers is to categorize the documents by reading through the abstract of all the

[4] https://www.facebook.com/
[5] https://twitter.com/

Table 3.1: A few examples of document titles which do not try to capture the essence of the document's content.

Document titles
Sic transit gloria telae: towards an understanding of the web's decay
Visual Encoding with Jittering Eyes
BuzzRank ... and the trend is your friend

submissions. Categorization would be much more simplified if keywords could be assigned simply by glancing over the title of the documents. This would reduce the computational cost significantly. However, using only document titles may be troublesome if the authors provide a catchy title to the document (as shown in table 3.3). In this chapter, we demonstrate that this can be accomplished using our proposed approach that leverages information from several open source web resources to assign relevant keywords.

Example 2: Removing non-relevant keywords: One of the main challenges with automatic or even manual keyword assignment to documents is the problem of irrelevant keywords causing drift from the main topic of the document. Often authors may insert keywords which are too specialized for their own work but such keywords may neither be useful for the purpose of categorization or retrieval of the document from a database. In an automated keyword assignment approach, irrelevant keywords may pop-up either due to the nature of the document or the applied algorithm. In either case, it is important to identify and remove keywords which are irrelevant. We propose an automated approach to handle this key challenge.

Example 3: Keyword extraction from short summary text: The problem of keyword extraction from short summary text of a manuscript is important for several practical purposes. One use case is the example of a conference submission system. In this case, it will be useful if keywords can be recommended to authors based on the information from the summary abstract. Another use case is to assign keywords to the permission protected documents on the web. Often such documents do not have full text access. Thus providing keywords, using only the summary abstract information would be very helpful for the authors. Another interesting

use case where the summary text based keyword extraction is useful are very long text documents. It is computationally inefficient to assign keyword by scanning the entire text of the document. However, short summary text has its own challenges. In this work, we identify two challenges associated with keyword extraction from short text. Firstly, the text content may not have sufficient occurrences of words to determine their importance in the structure of the short text. Secondly, the short text available for keyword extraction may not have keywords in the text. We solve these problems by incorporating global information about keywords from web resources. So even if the short text does not contain keywords, the proposed approach can generate keywords from the extended text content from the web resources.

The performance of the proposed approaches for the above mentioned challenges, have been evaluated on several real world datasets. We have mainly used scholarly research articles published in peer reviewed computer science conferences. We conduct extensive experiments to compare the performance of the proposed approach with several relevant baselines. We find that the proposed approach using web resources to enhance the information in the short text of the document outperforms the baseline keyword extraction approaches for keyword assignment. In particular, we highlight the success of appending the short summary text with text content from web resources to extract more relevant keywords in comparison to the keywords extracted from the local content of the document. The proposed approach also shows promising results for keyword assignment using only the title of the document.

3.1 Related Work

As described earlier, the literature under document annotation can be divided into two broad classes. The first class of approaches study the problem of annotation using extraction techniques [54, 55]. The main objective of such techniques is to identify important words or phrases from within the content of the document to summarize the document. This class of problem is studied in the literature by several names such as “topic identification” [56],“categorization”’ [57, 58], “topic finding” [59],“cluster labeling” [60, 61, 62, 63]and as well as “keyword extraction”[54, 55].

Researchers working on these problems have used both supervised and unsupervised machine learning algorithms to extract summary words for documents. Witten et al. [50] and Turney [51] are two key works in the area of supervised keyphrase extraction. In the area of

unsupervised algorithms for key phrase extraction, Mihalcea and Tarau [1] gave a textRank algorithm which exploits the structure of the text within the document to find key-phrases. Hasan and Ng [64] give an overview of the unsupervised techniques used in the literature.

In the class of key phrase abstraction based approaches. There can be two approaches for document annotation or document classification: single document annotation and multiple document annotation. In the single document summarization, several deep natural language analysis methods are applied. These strategies of document summarization use ontology knowledge based summarization [65, 66]. The ontology sources commonly used are wordNet, UMLS. The second approach widely used in single document summarization is feature appraisal based summarization. In this approach, static and dynamic features are constructed from the given document. Features such as sentence location, named entities, semantic similarity are used for finding document similarity.

In the case of multi-document strategies, the techniques incorporate diversity in the summary words by using words from other documents. However, these techniques are limited when the relevant set of documents is not available. Gabrilovich et. al.[53] proposed an innovative approach for document categorization which uses of Wikipedia knowledge base to overcome the limitation of generating category terms which are not present in the documents. However, this approach uses the entire content of the document and extend the context using Wikipedia.

3.2 Background of open source web resources

In this section we give a brief overview of the different open source web resources used in this chapter.

3.2.1 WikiCFP

As the name suggests, WikiCFP is a semantic wiki for calls for papers in science and technology fields. There are about 30,000 CFPs on WikiCFP. The knowledge source is used by more than 100,000 world wide researchers every month. WikiCFP is a well organized resource to browse through the CFPs of conference using keyword search. The CFPs are categorized using genealogy information of research topics. In addition to all this, WikiCFP contains the most updated call for papers and expired CFPs are automatically pushed down. Thus the CFP information is timely updated. In this work we have used the “topics of interest” section of the CFPs

for mining up-to-date topic information about research.

3.2.2 Crowd-sourced knowledge

Wikipedia[6] is currently the most popular free-content online encyclopedia containing over 4 million English articles since 2001. At present Wikipedia has a base of about 19 million registered users including over 1400 administrators. Wikipedia is written collaboratively by largely anonymous internet volunteers. There are about 77,000 active contributors working on the articles in Wikipedia. Thus the knowledge presented in the articles over the Wiki are convinced upon by editors of similar interest.

DBpedia[7] is a crowd-sourced community effort to present the information available in Wikipedia in a structured form. This information can be used to answer sophisticated queries on the Wikipedia database, for instance 'Give me all cities in New Jersey with more than 10,000 inhabitants' or 'Give me all Italian musicians from the 18th century'. The English version of the DBpedia knowledge base currently describes 4.0 million things, out of which 3.22 million are classified in a consistent ontology.

Similar to DBpedia is Yago which in addition to Wikipedia combines the clean taxonomy of WordNet. Currently, YAGO2s[67] has knowledge of more than 10 million entities (like persons, organizations, cities, etc.) and contains more than 120 million facts about these entities. Moreover, YAGO is an ontology that is anchored in time and space as it attaches a temporal dimension and a spacial dimension to many of its facts and entities proving a confirmed accuracy of 95%.

Freebase[8] is another online collection of structured data harvested from sources such as Wikipedia as well as individually contributed data from its users. Its database is structured as a graph model. This means that instead of using tables and keys to define data structures, Freebase defines its data structure as a set of nodes and a set of links that establish relationships between the nodes. Because its data structure is non-hierarchical, Freebase can model much more complex relationships between individual elements than a conventional database, and is open for users to enter new objects and relationships into the underlying graph.

[6] www.wikipedia.org/
[7] http://wiki.dbpedia.org/About
[8] https://www.freebase.com/

3.2.3 Academic search engines

Academic search engines provides a universal collection of research documents. Search engines such as Google scholar and similar other academic search engines have made the task of finding relevant documents for a topic of interest very fast and efficient. We use the capacity of search engines to find relevant documents for a given query document. We have used the Google scholar search engine[9] and University of Minnesota's MNCAT library search engine[10] for this purpose.

3.3 Overview of our work

The unique challenges associated with keyword assignment for short text documents are explained earlier. As a result, computational techniques are required to automatically assign relevant keywords to such low content documents.

In this chapter, we focus on the problem of automating keyword assignment for low text content documents. In particular, we address the three fold challenges described earlier. Before moving forward about the details of the proposed approaches, we summarize the flow of the overall framework.

The problem of keyword assignment for low text content documents is addressed in two ways. In figure 3.1, we compare two different models for keywords assignment problem, one for full text (conventional model) and another for short text (proposed model). In the rest of the paper, we will discuss the model for short text document. We will describe the proposed approaches to leverage content from open source web resources to improve (i) keyword abstraction and (ii) keywords extraction. In the category of keyword abstraction, we first present an approach that leverages information from web resources like WikiCFP and Wikipedia for automating keyword selection for such document. Generation of keywords from the web resources helps to generate keywords which are up-to-date with the current scientific terminologies. The problem of low text content of the document is overcome by generating additional content using the web resources like academic search engine. The additional content is known as 'web context' because the information is generated using external global resources rather than the 'local' text content of the document. The new text content of the document is then used to

[9] scholar.google.com/
[10] https://www.lib.umn.edu/

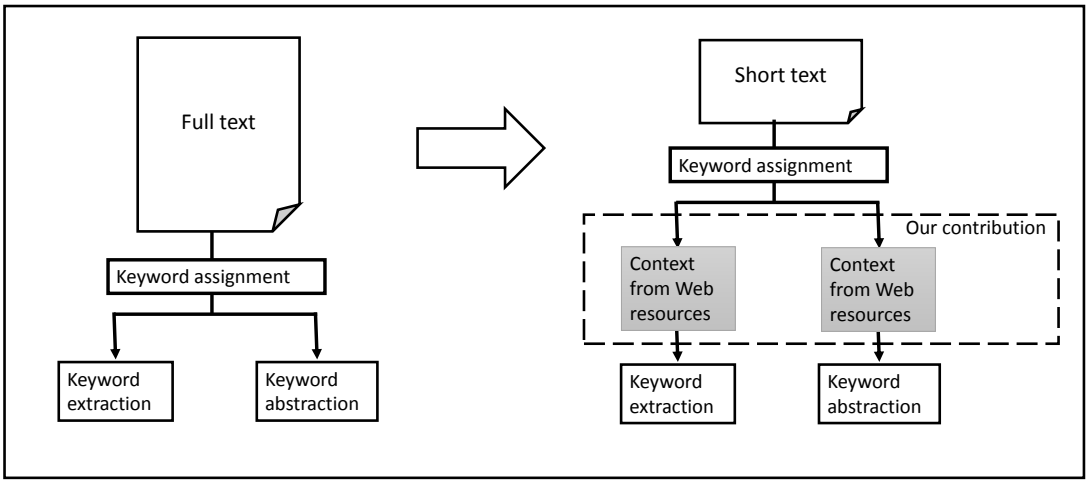


Figure 3.1: Figure explaining the big picture of the contributions of our work. Figure (a) is a pictorial summary of the keyword assignment approaches for full text documents, (b) shows proposed model of keyword assignment for short text documents.

select keywords from the list of keywords generated using crowd sourced web resources. In the second approach, we fully automate the step of keyword assignment by assigning keywords from crowd sourced topics available in crowd sourced knowledge bases like Dbpedia, Yago and Freebase. The relevance of the keywords to the document is automatically inferred using a web-distance based clustering approach. Non-relevant keywords are detected and removed in an unsupervised manner. The proposed approaches are tested on real world dataset from DBLP, the approach, however, is generic enough to be used for various types of documents like news articles, patent documents and other documents where keywords needs to be assigned using only the short text content.

In the category of keyword extraction, we propose a novel model for automatic keyword extraction from the low text content of the document. We show that the quality of the assigned keywords is improved by incorporating relevant context from the web. In addition to the text content of the document, the text content generated from the web adds global information about the document's topic and thus improves keyword extraction. The hypothesis is tested on a real world document corpus. We evaluate the performance of three well known keyword extraction techniques and compared the impact of adding the proposed web-context (described in Section 3.4) to the local content of the document.

Since, text content generation using open source web resources is used to enhance both the keyword abstraction and extraction, we first describe this step in the next section. For the sake of convenience, we refer to this step as web context extraction.

3.4 Web context extraction

Given the 'short text' S' (which can be 1-10 word length), the expanded context is generated using open source web resources such as the academic search engine. As shown in figure 3.5, the context of the 'short text' is expanded using the results obtained by querying the web corpus using an academic search engine. The 'short text' is used as a query to the search engine. For the query S', the search engine retrieves 'relevant' ranked results. The web context for the input 'short text' is created from the text in the titles of the retrieved documents. There are several other resources in the retrieved results that can be used (like snippets, author information etc) but those are not included in the web context because not all academic search engines offer these features. The final context is created by concatenating the text in the titles of the top-n documents. The value of n is not fixed and can be a parameter to the approach. In the later section, the results are evaluated by varying the value of k. The text is tokenized using space delimiter. Duplicate titles are excluded from the web context. As a basic step in text mining, the tokens are pre-processed by applying stop-word filter, non-alphabetic character removal and length-2 token removal. In the rest of the paper, we will refer to this context for a 'short text' (S') as web-context ($WC(S')$) for the sake of convenience and consistency.

3.5 Keyword abstraction

3.5.1 Finding keywords for documents with 5-10 text words

In this section, we describe a three step approach taken to automatically assign keywords to a text document given only the title information for the document. The three steps (shown in figure 3.2 are enumerated in the following subsections.

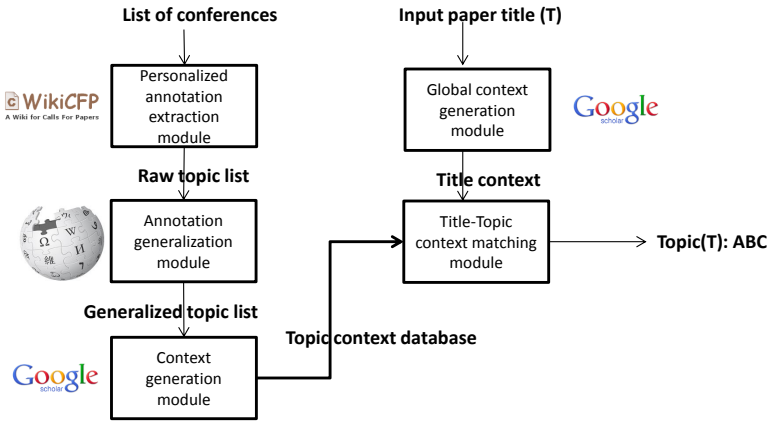


Figure 3.2: A systematic framework of the proposed approach

Topic/Keyword Generation

Automatic topic generation is a challenging task. In most of the literature, this step of document annotation involves expert knowledge to generate topics to be assigned to documents. Here we show a simple and novel approach to automate the task of topic generation using open source media such as WikiCFP and Wikipedia. The proposed topic generation involves two step (as shown in figure 3.2)

Personalized topic retrieval: In this step, we first identify the human created labels for scientific research topics. At the first step, a list W consisting of various venues of research publications (conferences and Journals) is created in an unsupervised manner. We have narrowed down the domain to data mining research venues. For each venue in this list W, personalized annotations are retrieved from its CFP (Call for papers) from WikiCFP (highlighted with blue box in figure 3.3. As mentioned earlier, WikiCFP provides a feature for obtaining the CFPs for any research venue. The organized knowledge in this database helps to retrieve various forms of information about the venue such as its categories, deadline, date, CFPs and related venues. We use the last two features to create a database of personalized annotations for research venues in


WI 2013 : IEEE/WIC/ACM International Conference on Web Intelligence

SHARE

Conference Series : Web Intelligence

Link: http://cs.gsu.edu/wic2013/wi

When	Oct 17, 2013 - Oct 20, 2013
Where	Atlanta, USA
Submission Deadline	May 1, 2013
Notification Due	Jul 1, 2013
Final Version Due	Sep 1, 2013

Categories web artificial intelligence web information retrieval world wide wisdom web

Call For Papers

WIC 2013 provides a leading international forum to bring together researchers and practitioners from diverse fields, to increase the cross-fertilization of ideas and explore the fundamental roles, interactions as well as practical impacts of Artificial Intelligence engineering and Advanced Information Technology on the next generation of Web systems. Web Intelligence has been recognized as one of the most important as well as promising direction for scientific research and development in the era of Web and agent intelligence to bring in the next generation Web systems. Furthermore, WI-IAT 2013 will include workshops providing in-depth background on subjects that are of broad interest to Web intelligence and Intelligent Agent Technology communities. The workshop programs will focus on new research challenges, initiatives and applications WIC 2013 is an excellent opportunity for researchers who wish to examine design principles and performance characteristics of various approaches in web intelligence technology, and increase the cross-fertilization of ideas on the development of web intelligence systems among different domains.

WIC 2013 topics and area include, but not limited to:

Web Intelligence Foundations

- Brain Informatics for WI
- Human Level WI
- New Cognitive Models and Computational Models for WI
- Granular Computing (GrC) for WI
- Autonomy-Oriented Computing (AOC) for WI
- Complex Networks for WI
- Cyber Individual and Personalization

Figure 3.3: A snapshot from WikiCFP showing the personalized topic list in the conference CFP.

list W. For the venues where the CFP information is unavailable its related venues are queried in the similar manner. The output of this step is a database D consisting of human generated topics for paper submission. Database creation from WikiCFP assures that the research topics are up-to-date with the research trend. It also enhances the diversity of topics in the database.

Wikifying personalized topics/keywords The result of the previous step is a database D consisting of human generated topics. However, the database contains noise in several forms such as repetition of research topics, irrelevant topics, highly specialized topics. As an example, "mining of web data" and "web based mining" refer to the same topic "web mining" however they were expressed differently in different CFPs due to personalization by different individuals. Given the complications of natural language interpretation, construction of a refined list of topics from the topics in the database D cannot be done by fusion or intersection of the duplicate topics. In order to remove such noise from the database D, we have used the crowd sourced intelligence of Wikipedia to refine and generalize them. This step of using Wikipedia for refining the personalized topics is termed as Wikifying. The following steps are used for Wikifying.

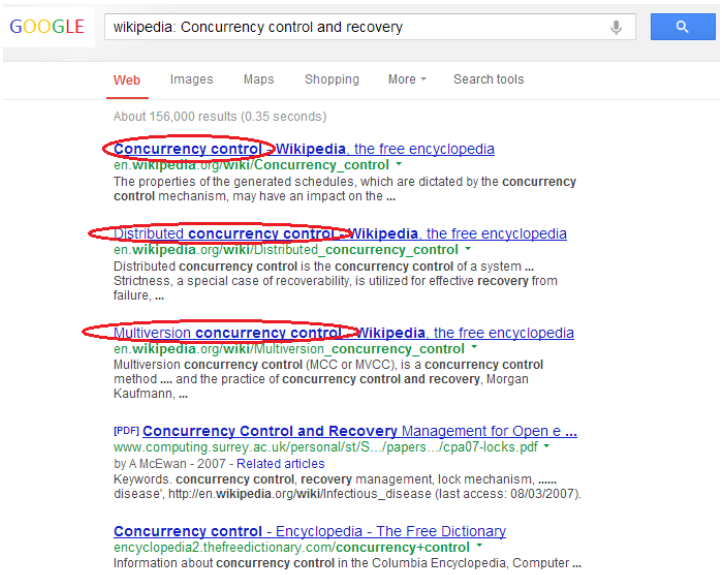


Figure 3.4: A snapshot of Google web page showing the Wikification of the input query.

- *Wiki Querying*- Each topic in D is queried as a Wikipedia item in the Google search engine. Given a query q in the Google search engine as $wikipedia : q$, it returns a list of results from Wikipedia titles with which are related to q.

- *Top-N extraction*- Next, we extract the top-N Wikipedia results(document titles) for the query q from the Google search page (as shown in the figure 3.4).

The idea behind wikifying the original topics/keywords from CFPs is to bring in an element of consensus from crowd sourced knowledge of Wikipedia. As an output of this step, we get a refined database D' which contains refined topics after wikifying the topics in database D. We call these topics as wikified topics/keywords.

Using the web context

Using the approach discussed in section 3.4, we derive the additional text content for each keyword (topic) in the list of keywords (D') by using the keyword as the input to generate the web context (WC_{topic}). Similarly, the additional text content for the input document (WC_{d_i}) is derived using its title as the input 'short text' for web context generation.

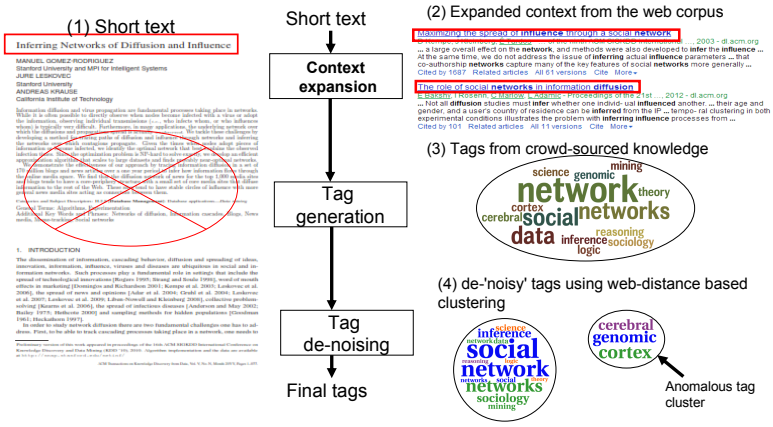


Figure 3.5: A systematic framework of the proposed approach. An example is illustrated to explain the proposed approach.

Topic Ranking

As stated in the previous step, each topic in the corpus D' is described by its web context WC_{topic}. Similarly, the title of a given document is described by its web context WC_{d_i}. The topics in the corpus D' are ranked for a input document title S' using three text similarity computation models namely, TF-IDF[68], LDA-TFIDF model [69] and LSI-TFIDF model [70].

Ranking : The ranking of the topics in D' for a given document is obtained by computing the cosine similarity between the features derived from the web context of input document and the keywords in D'. For a given document, the topics $\in D'$ are ranked in the decreasing order of their cosine similarity scores.

3.5.2 Automating keyword abstraction and de-noising

In this section, we describe a different approach for keyword abstraction. Here, we also propose an approach to de-noise the predicted keyword list. Similar to the previous section, the only text information about the document is the text information in its title.

The task of fully automating keyword assignment is accomplished by selecting keywords from the topics/keywords in the crowd sourced knowledge bases like Dbpedia, Yago and Freebase and finally the non relevant tags are removed by using a fully unsupervised approach. There are three main components of this approach: (1) Context expansion using academic search engine (similar to web context generation in previous section), (2) candidate keyword generation using crowd-sourced knowledge and (3) de-noising keywords using web-based distance clustering technique.

Using the web context

Given the title text of a document (d_i) as the only text information for the document, its additional text content is generated using the title text as the input for web context extraction (described in section 3.4). It is referred as web context ($WC(S')$).

Tag generation

In this section, we describe the procedure to utilize crowd-sourced knowledge to generate keywords from the expanded context $WC(S')$. As described earlier, the crowd-sourced knowledge is available in well structured formats unlike the unstructured web. The structured nature of knowledge from sources such as DBpedia, Freebase, Yago, Cyc provides opportunity to tap in the world knowledge from these sources. The knowledge of these sources is used in the form of concepts and named entity information present in them, since the concepts and named entities consists of generic terms useful for tagging. We have used the AlchemyAPI[3] to access these knowledge bases. A tool such as this provide a one-stroke access to all these knowledge bases at once and returns a union of results from all the various sources.

Given the expanded context ($WC(S')$) as the input to the AlchemyAPI, which matches the $WC(S')$ against the indices of these knowledge sources to match $WC(S')$, using the word frequency distribution, with concepts and *named entities* stored in the knowledge bases. The output for an API query $WC(S')$ is a list of concepts and named entities. Using the open source knowledge bases and the word frequency information from the input, the API returns a list of concepts related to the content. The named entity list returned for a query $WC(S')$ consist of only those named entities of type 'field terminologies'. There are other type of named entities such as 'person name', 'job title', 'institution' and a few other categories but those are not

generic enough to be used as keywords. The concepts and named entities for $WC(S')$ together form a *keyword cloud* T.

Figure 3.5 highlights a keyword cloud consisting of keywords generated using the above described technique. As shown in the figure, the keywords are weighted based on the word distribution in $C(S')$. This example also shows a few keywords like 'cerebral', 'cortex', 'genomic' that appear to be inconsistent with the overall theme of the cloud for $C(S')$. The next step describes an algorithm to handle such situations in the tagging process.

Keyword de-noising

As described in the previous step, the keyword cloud T for $C(S')$ may contain some inconsistent or 'noise' keywords in it. Here, we describe an algorithm to handle the problem of 'noise' in the keyword cloud. This step is therefore termed as keyword de-noising.

Given the keyword cloud T for $C(S')$, noisy keywords are pruned in the following manner. The keywords in T are clustered using a pairwise semantic distance measure. Between any two keywords in T, the semantic distance is computed using the unstructured web in the following way. For any two keywords t_1 and t_2 in T, $dis(t_1, t_2)$ is defined as the normalized Google distance(NGD)[45]:

$$NGD(t_1, t_2) = \frac{max\{logf(t_1), logf(t_2)\} - logf(t_1, t_2)}{logM - min\{logf(t_1), logf(t_2)\}}$$

where M is the total number of web pages indexed by the search engine; $f(t_1)$ and $f(t_2)$ are the number of hits for search terms $t_1$and t_2, respectively; and $f(t_1, t_2)$ is the number of web pages on which both t_1 and t_2 occur simultaneously.

Using the NGD metric, a pairwise distance matrix(M) is generated for the keyword cloud T. The pairwise matrix M is then used to identify clusters in the keyword cloud. The cloud is then partitioned into two clusters using different hierarchical clustering techniques. Here, we assume that there is at least one 'noise' tag in the tag cloud T. Out of the two clusters identified from the keyword cloud T, the one cluster with majority tags is called a normal cluster whereas the other cluster is called as outlier cluster (or noisy cluster). In case of no clear majority the tie is broken randomly.

The algorithm is illustrated through an example shown in figure 3.5 step 4. This step shows that the keywords generated in step 3 are partitioned into two clusters as described above. The keywords in one clusters are semantically closer than the keywords in the other clusters. As

shown in this example, the outlier keywords 'cerebral', 'cortex', 'genomic' are clustered together while the remaining normal keywords cluster together. Since the former is a smaller cluster, it is pruned out from the keyword cloud. Lastly, the final keywords consists of only the keywrods in the larger cluster.

3.6 Keyword extraction from the summary text of a document

In the domain of document-oriented keyword extraction, the recent methods focus on three approaches namely, (1) natural language processing based techniques (2) statistical co-occurrence based techniques (3) world knowledge based techniques. In order to familiarize the readers with the above mentioned approaches, we present a brief description about the popular algorithms in each of the above categories. Although these techniques are not the contributions of this work but we will extensively use these techniques in the proposed model for keyword extraction.

3.6.1 TextRank[1]

Under the natural language based techniques, TextRank is the most popular technique for document-oriented keyword extraction. TextRank is based on term selection from the text based on the part-of-speech (POS) tagging of the terms and then applying a series of syntactic filters to identify POS tags that are used to select words to evaluate as keywords[2]. Using a fixed-size sliding window approach, co-occurring words within the window are is accumulated within a word co-occurrence graph. The candidate keyword are selected from the co-occurrence graph by ranking the words based on a graph based ranking algorithm (TextRank- similar to pageRank[71]). The top-ranking words are selected as keywords. Multi-word keywords are formed by combining adjacent keywords.

3.6.2 Rake[2]

Under the statistical co-occurrence based techniques, the Rapid automatic keyword extraction (RAKE) is most popular for document oriented keyword extraction. RAKE is an unsupervised, domain-independent and language independent method for extracting keywords from individual documents. Keyword extraction is done by means of ranking the non stop-words (a list of most common English words). The phrases in the non stop-words are identified across the document

and then scored based on the proposed metrics. The input parameters for RAKE comprise of the document text and a set of stop-words, a set of phrase delimiters and a set of word delimiters. RAKE uses stop words and phrase delimiters to partition the document in candidate keywords. Based on the co-occurrence of words within the candidate keywords, the word co-occurrences are identified. This approach for identifying co-occurring words save the computation cost of using sliding window technique used in TextRank. Finally, the word association is measured in a manner that adapts to the style and content of the text and therefore important for scoring the candidate keywords.

3.6.3 Alchemy[3]

There are several techniques used for leveraging world knowledge to extract keywords. In addition to the local content and style information from the document to extract keywords, these approaches utilize information from crowd sourced corpus like Wikipedia, Freebase, Yago and other similar corpus to identify keywords. The AlchemyAPI[3] uses statistical algorithms and natural language processing technology, in addition to the world knowledge using various crowd-sourced corpus, to extract keywords from the text of the documents. The scoring of the candidate keywords is influenced by incorporating statistical information from the world knowledge corpus.

3.6.4 Adding web context

In addition to the local content of the document, the additional information with the author to assign keywords comes from the various other related literature. An author, generally assign keywords that highlight both the local content of the document as well as the relation of the document with the other topics in the category of his/her document. So the author assigned keywords are a balanced mixture of both the local content and the global content. Here, we generate the web context for a document (d_i) using its title text as the input for the web context extraction step (described in section 3.4).

3.7 Experimental analysis for keyword abstraction for documents with 5-10 text words

In this section, we discuss the experiments and results for the keyword abstraction framework for both the approaches described in section 3.5. We first describe the test dataset, the ground truth, the baseline and the evaluation metric used for evaluation of the proposed approach.

3.7.1 Test dataset description

For the purpose of evaluating our approach, a test set consisting of 50 research documents from top tier computer science conferences was constructed. The 50 papers were selected to capture the variety of documents in computer science research. Several of the documents had catchy titles (examples given in table 3.3) and the titles were never intended to convey the core idea of the document. In our algorithm, we used only the title information as the input to the algorithms.

For the proposed approach, the aggregated keywords, from the conferences in which these 50 papers were published, totals to 777 keywords. The keywords for the test documents was selected from these keywords.

3.7.2 Ground truth

In absence of any gold standard annotations for the test documents, the ground truth for the documents was collected from the author assigned keywords to these documents. We collected this information by parsing the 50 documents in the test set. We assume that the keywords assign by the authors are representative of the keywords for the document. The proposed approach and the baselines were evaluated on this ground truth.

3.7.3 Baseline approach

We have compared the performance of the proposed approaches with two baselines which use only the local content information for topic assignment. The first baseline uses only the title and abstract information about the document to identify relevant topics from a given list of topics. For the keyword selection algorithm, the keywords are selected from a list of keywords available in faceted DBLP project [72]. This list contains the most popular author keywords assigned to a minimum of 100 research articles in the DBLP dataset [46]. The keywords in the

list, unlike those used in the proposed approach, are actually human generated keywords/topics. The second baseline uses the entire content of the documents (title, abstract and all the other section). While taking into account the full content of the documents, we removed the keywords assigned to the paper. We used a tf-idf model to vectorize each of the test document in terms of the keywords. Using the top-10 highest tf- idf values, we assigned 10 keywords to each document from the list of given keywords.

For the fully automated keyword abstraction algorithm, we compare the performance of the proposed approach with a baseline using the full text content of the test documents. The full text information is generated from the pdf versions of the test documents. The PDF documents were converted to text files using PDF conversion tools. As a basic pre-processing step, stop-words, non-alphabetical characters and special symbols were removed from the text to generate a bag of word representation of the full text. For the purpose of comparison, the full text context was used to generate keywords using the proposed approach and at the final step, de-noising of tags was done using the proposed algorithm. The purpose of this baseline is to see the effectiveness of using the global context as a replacement for the full text content of the document.

3.7.4 Evaluation metrics

Given that evaluation by experts is resourceful and time consuming, hence challenging, we evaluate the results of the proposed approach for keyword selection using the following metrics.

Recall

Recall is used to compute the proportion of the ground truth results that was correctly identified in the retrieved results. Each document is marked to be labeled correctly if atleast one of the retrieved/predicted keywords exactly matches with any one keyword in the ground truth for the document. The proposed approach and the baselines were evaluated on the same metric.

Jaccard index

The Jaccard similarity between two sets A and B is defined as the ratio of the size of the intersection of these sets to the size of the union of the sets. It can be mathematically stated as:

$$J(A,B) = \frac{|A \bigcap B|}{|A \bigcup B|}$$

Table 3.2: Table shows comparison of recall for the baselines (BS_{abs}, BS_{whole}) using only the local context information for document summarization and the proposed approach $(PA_{TFIDF}, PA_{LDA}, PA_{LSI})$ which uses global context generated using search engines in the top-10 results.

Annotation techniques	BS_{abs}	BS_{whole}	PA_{TFIDF}	PA_{LDA}	PA_{LSI}
Recall for top-10	0.50	0.50	0.56	0.50	0.50

We compare the Jaccard index of the predicted tags (with the ground truth tags) and the baseline tags (with the ground truth tags). The Jaccard index is averaged over the total number of documents in the test dataset.

Execution time

The final metric for comparing the proposed approach with the baseline is the execution times. Since the main overhead of the approach is in the first step of tag generation due to difference in sizes of the input context. The execution time is computed as the time taken in seconds to generate tags for the 50 test documents given their input context. For the proposed approach the context is derived using web intelligence whereas for the baseline the context is the full text of the test document. Pre-processing overheads are not taken into account while computing execution timings.

3.7.5 Results and discussion for keyword abstraction without de-noising

This section is sub-categorized into two sections. The first section discusses the quantitative evaluation of the proposed work. In the next section, we qualitatively discuss our results for a few examples of documents with catchy titles.

Quantitative evaluation

In this section, we discuss the results of the experiments performed in the previous section. Table 3.2 gives a summary of the recall in the top-10 results obtained using the proposed approaches and the baselines. As shown in the table, the recall in the top-10 improves (0.56 vs 0.50) in case of the proposed approach (using TFIDF model for comparing content similarity). It can also be seen that the recall using the proposed approach is comparable to the baseline

approaches (approx. 0.50 in both cases). The comparable results signify the effectiveness of the fully automated approach to identify topics for research documents. Thus, given only the title information about a research document, we can automatically assign topics to the document which are very close to human assigned topics.

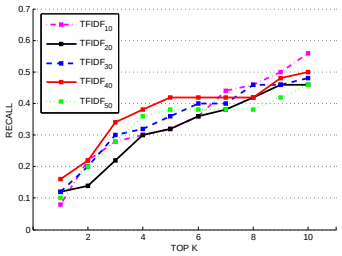


Figure 3.6: Recall@k using TFIDF.

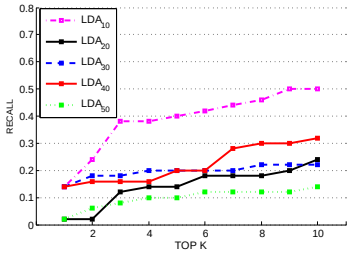


Figure 3.7: Recall@k using LDA.

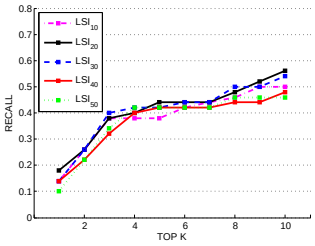


Figure 3.8: Recall@k using LSI .

In the next set of experiments, the global context size was varied from 10 through 50 in steps of 10. In the figures 3.6, 3.7, 3.8, we show the variation of recall in the top-k results of the different proposed approaches. In these figures, we also compare the recalls when the global context size is varied. In the figure 3.6, the recall increases monotonically in all the different global context sizes. The highest recall (0.56) using TFIDF model occurs in the top-10 and using the global context of size 10. The context size of 40 gives the next best recall. However, we also observe that adding extra context (e.g green curve for context size 50-$TFIDF_{50}$), the recall is lower in comparison with other context sizes.

In figure 3.7, we observe a significant difference in the performances by varying the context size. The highest recall attained using LDA model is 0.50 and it happens when the context size is 10. From this figure, we see that the LDA model is very sensitive to the content information. As shown in the figure, adding extra content (green curve- LDA_{50}), the recall is very low (< 0.20).

In the case of LSI model for similarity evaluation, figure 3.8 shows a similar trend in performances using different context sizes. As shown, the recall for context sizes of 40 and 50 (shown in red and green curves respectively), is lower in comparison to the recall using context size of 10, 20 or 30. However, the highest recall (0.58) is attained when the context size is 20.

Table 3.3: Table showing the topics assigned to the documents with catchy titles.

Document titles	**Our approach**	**Ground truth**
Sic transit gloria telae: towards an understanding of the web's decay	link analysis, adversarial information retrieval, bayesian spam filtering	Link analysis, Web information retrieval, Web decay, dead links
Visual Encoding with Jittering Eyes	semantic memory, bag-of-words model, motion analysis, computer vision	Information retrieval, personal information management, human-computer interaction, World Wide Web use
BuzzRank ... and the trend is your friend	Pagerank, web community, e-social science	Web graph, Web dynamics, PageRank

Qualitative evaluation

In this section we discuss the results of the proposed approach by qualitatively analyzing the results of the proposed algorithm. Using a quantitative measure like recall fails to account for the subjective accuracy of other topics assigned to a document in our top-10 results when compared to the ground truth by excluding the exact match scenario. Here we analyze the results in a subjective manner.

Table 3.3 shows the results of summarization using the proposed approach and the ground truth keywords. This is special category of documents where the titles of the document are "catchy" and do not intend to display the core idea of the document. Our algorithm uses only this title information to generate the context and find topics suitable for these documents. The column two shows some of our results in the top 10. For the document titled as "Sic transit gloria telae." the ground truth keywords are very closely related to what we find using our approach. The term "information retrieval" is common in both. In the next example - "Visual encoding with Jittering eyes"- the topic "motion analysis" and "computer vision" are very closely related with the term human-computer interaction. Similarly, "bag-of-words model" is frequently used in "information retrieval" and "semantic memory" uses "World Wide Web". In the third example- "BuzzRank.. and the trend is your friend"- the topics "web community" and "e-social science" are highly relevant for "web graph" and "web dynamics".

3.7.6 Results and discussion for keyword abstraction with de-noising

This section is sub-categorized into two sections. The first section discusses the quantitative evaluation of the proposed work. In the next section, we qualitatively discuss our results for some of the test documents.

Table 3.4: Table showing Jaccard Index measure for the proposed approach (varying k in context expansion) and the full content baseline

clustering algorithm	k=10	k=20	k=30	k=40	k=50	Full Text*
unpruned	0.054	**0.059**	0.052	0.058	0.052	0.044
single	0.054	**0.057**	0.050	**0.057**	0.056	0.040
complete	**0.058**	0.055	0.043	0.047	0.052	0.034
average	0.052	**0.059**	0.052	**0.059**	0.054	0.034

Quantitative evaluation

In order to compare the quality of keywords generated by both the approaches, we evaluate the results of the proposed approach and the baseline approach using the ground truth keywords for the test documents. The results of this experiment are shown in table 3.4. The first five columns correspond to the expanded context extracted using k as 10, 20, 30, 40 and 50. The Jaccard index of the baseline(Fulltext) with the ground truth is 0.044 whereas the Jaccard index for all the expanded context (proposed approach) over all values of k is greater than 0.50. The highest Jaccard index is 0.059 at k=20.

When we use the single hierarchical clustering algorithm for de-noising, the Jaccard index is only reduced to 0.040 for Fulltext baseline. The Jaccard index for the expanded context with k=20,40 is 0.057 which is clearly higher to the baseline results. Similarly, for the complete hierarchical clustering based de-noising, the Jaccard index is 0.058 for k=10 whereas it is only 0.034 for the full text baseline. The same scenario is found for average hierarchical clustering based de-noising. The Jaccard index is 0.059 for k=20,40 while it is only 0.034 for Fulltext baseline.

The experiment described above shows a quantitative approach for comparing the quality of resultant keywords from the proposed and the baseline approaches. The results shown above surprisingly favor the keywords generated by the proposed approach. The baseline approach uses the full text of the document in order to generate keywords. An explanation for the observed results can be attributed to the fact that context derived from the web contains a wide spectrum of terms useful for generating generalized tags for the document. While on the other hand, the full text approach uses only the terms local to the specific document which might not be diverse enough to generate generalized keywords.

Table 3.5: Table showing results for a few of the sample documents. This table shows that several of the topics in the second column (our approach) are very closely related to the keywords in the ground truth (column 3).

Document titles	**Our approach**	**Ground truth**
iTag: A Personalized Blog Tagger	web search,semantic technologies,semantic metadata,tag,meta data,computational linguistics, social bookmarking,data management	Tagging, Blogs, Machine Learning
Advances in Phonetic Word Spotting	speech recognition,language,linguistics,information retrieval,mobile phones,phoneme,speech processing, natural language processing,consonant,handwriting recognition,neural network	Speech recognition, synthesis Text analysis, Information Search and Retrieval
Mining the peanut gallery: opinion extraction and semantic classification of product reviews	linguistics,supervised learning,book review, unsupervised learning,review,parsing, sentiment analysis,machine learning	Opinion mining, document classification
Swoogle: A Search and Metadata Engine for the Semantic Web	world wide web,search engine,web search engine, internet,social network, semantic search engine, search tools,semantic web,social networks,search engine optimization,ontology,web 2.0,semantics	Semantic Web, Search, Meta-data,Rank,Crawler
Factorizing Personalized Markov Chains for Next-Basket Recommendation	cold start,matrix, recommender systems, collective intelligence,markov chain, collaborative filtering, markov decision process	Basket Recommendation, Markov Chain, Matrix Factorization

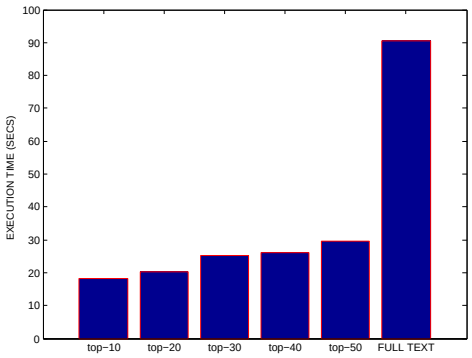


Figure 3.9: Figure showing execution time comparison for the tag generation step using the expanded context (varying k) vs the full text for 50 documents.

One of the challenges described about using the full text approach is the issue of time consumption for reading the full text in case the document is large. In figure 3.9, we show the results of an experiment conducted to compare the execution time of the keyword generation step for the proposed and the baseline approaches. The x-axis in the figure shows the expanded context (using different values of k) and the baseline(Full text). The y-axis corresponds to the total execution time(in seconds) for 50 documents. As shown in the figure, the execution time for the baseline is approximately 90 seconds for 50 documents whereas the maximum execution time is only 30 seconds for the expanded context where k=50. As shown earlier, that the quality of keywords generated using expanded context with k=10 or k=20 is as good as higher values of k. This implies that good quality keywords for a document can be generated 4.5 times faster using the proposed approach than using the full text of the document. This shows the effectiveness of the proposed approach to be useful in real time systems.

Qualitative evaluation

In this section, we discuss the results of the proposed approach by qualitatively analyzing the results of the proposed algorithm. The last section highlighted the performance of the proposed algorithm and quantitatively compared the results with the baseline using the Jaccard index. However, using a quantitative measure like Jaccard fails to account for the subjective accuracy of the tags other than those which do not match the ground truth exactly. Here we analyze the results subjectively.

Table 3.5 shows the keywords predicted by the proposed approach and the ground truth keywords for a few sample documents from the test dataset. For the first document in the table ('iTag: A Personalized Bog Tagger'), the keywords (our ground truth) assigned by the used contains terms like 'tagging', 'blogs' and 'Machine learning'. Although there are no exact match between the proposed keywords and the ground truth keywords yet the relevance of the proposed keywords is striking. Keywords such as 'semantic meta data','social bookmarking', 'tag','computational linguistics' are similar others in this list are clearly good tags for this document. Another example is shown in the next row. The ground truth keyword 'speech recognition' exactly match the keyword in the proposed list. However, most of the other keywords in the list of proposed keywords are quite relevant. For example, tags such as 'linguistics','natural language processing' are closely related to this document. A few tags such as 'mobile phones','consonant,'hand writing recognition' may not be directly related. The

third example shown in this table also confirms the effectiveness of the proposed approach. The ground truth consists of only two tag: 'opinion mining' and 'document classification' while the proposed tag list consist several relevant tags though there is no exact match.

The last two examples shown in this table demonstrate the effectiveness of the approach to expand the keywords. The fourth example is originally tagged with keywords like 'semantic web','search','meta-data', 'rank' and 'crawler'. But the proposed list consists of highly relevant keywords like 'ontology', 'search optimization' which capture even the technique used in the particular research document. Similarly, for the last example the non-overlapping tags are relevant for annotating the research document.

3.8 Experimental analysis of the keyword extraction approach

The experimental setup is divided into two subsections. In the first part, we describe the preliminary analysis of the datasets used and explain the motivation for using the proposed approach. In the second part, we discuss the experimental design for evaluating the performance of the proposed approach. Before going into the details of the two sections, we first discuss about the datasets used in this work.

3.8.1 Dataset description

There are two datasets used for this approach. The datasets consists of research documents from SIGKDD conference series[11] , top-tier Data Mining conference in the computer science discipline. Each of the datasets consists of information about three attributes of the documents, namely, the title of the document, the abstract/summary text and the author assigned keywords. The information was obtained from the ACM digital library. The various statistics about the two datasets are given in table 3.6. Since the keyword vocabulary changes with time, we have selected datasets from different years to evaluate the performance of the proposed approach over time.

[11] http://www.sigkdd.org/

Table 3.6: Table summarizing the datasets used.

Dataset id	Year	Count of documents
$dataset_1$	2008	118
$dataset_2$	2012	112

Table 3.7: Results of preliminary analysis of the datasets.

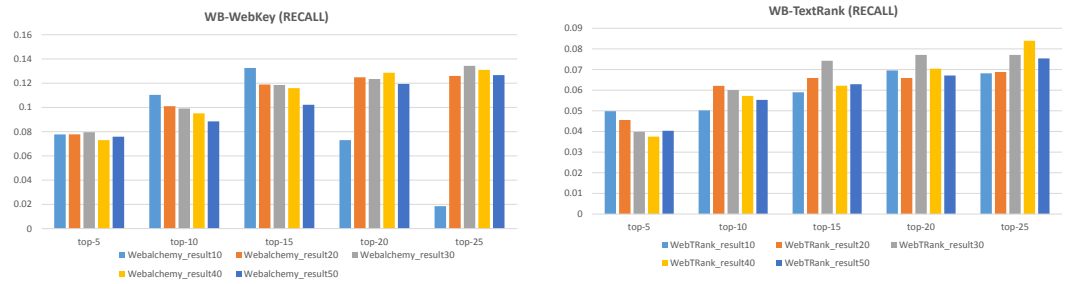
Dataset id	Total keywords	Keywords in titles	Keywords in abstract
$dataset_1$	438	3	208 (47.5%)
$dataset_2$	579	1	261 (45.0%)

3.8.2 Preliminary analysis of the data

In this experiment, we performed a preliminary analysis of the above described datasets. The aim of the experiment is to derive statistics about the keywords used in the research documents. Table 3.7 summarizes the statistics collected from this experiment. As shown in this table, the $dataset_1$ consists of 438 keywords and 47.5% of these keywords appear in the abstract/ summary text of the document. However, the number of total keywords in $dataset_2$ are higher than in $dataset_1$. But even in this dataset, we find that 45% of the keywords appeared in the abstracts of the document. It should also be noted that negligible number of keywords appear in the title of the documents. From this analysis, we find that any keyword extraction algorithm can extract keywords from the abstract of the documents with lesser than 50% recall. So, in order to improve the recall of any algorithm, we need extra content information in addition to the text in the abstract. Using full content of the document is one approach, however it has challenges such as (1) the computational time; and (2) risk of adding noise keywords[49]. In such a scenario, the proposed approach provides a computationally efficient and superior solution (as shown in the following sections).

3.8.3 Experimental design parameters

In this section, we discuss the experimental parameters like the ground truth, the baselines, the evaluation metrics and various experiments to evaluate the performance of the proposed approach.



(a) (b)

Figure 3.10: Quantitative evaluation of n in the top-n titles used to create web-context. The plots shows evaluation for (a) Alchemy and (b) TextRank techniques

Ground truth

The ground truth is prepared from the author assigned keywords for the research documents. We assume that the author assigned keywords are the best description of the summary of his document. In order to improve the comparison with the predicted keywords, the ground truth keywords are reduced to their root form using the stemming techniques. We used nltk (natural language toolkit) in python to stem the words to their root forms.

Baselines

Since, the proposed approach creates extra content to improve keyword extraction, the approach can be used as a pre-step to any keyword extraction algorithm. Thus, the baselines in our experiments is the keyword extraction from the local text content of the document.

Evaluation metrics

We use precision and recall metrics to compare the performance of the proposed approach with the baselines. The precision and recall values are averaged over the number of documents in the datasets.

The experiments conducted in this work are aimed at bringing out three main understandings in relation to the performance of the proposed approach. We briefly describe the aim of the experiments in this section. The detailed analysis of the results in presented in the next section.

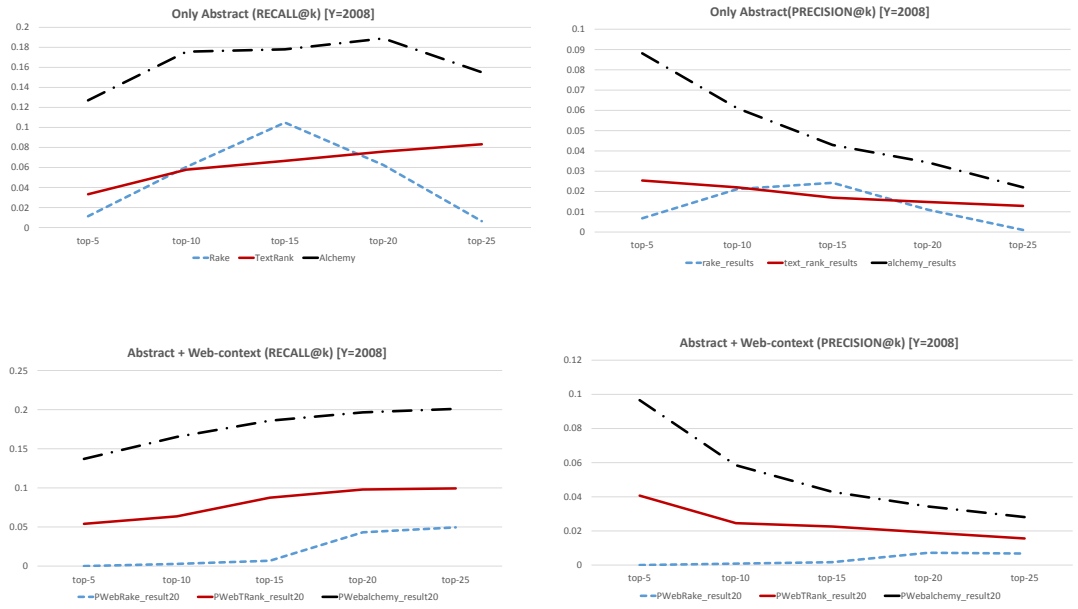


Figure 3.11: Quantitative comparison of Rake, TextRank and Alchemy keyword extraction techniques for $dataset_1$. Plots (a,c) show the comparison via the recall@k metric. Plots (b,d) show the comparison via the precision@k metric. The value of k is in the range {5, 10, 15, 20, 25}. The results are shown for different document content (only abstract, only web-context and abstract+web-context). These figure show that the Alchemy keyword extraction technique (shown in **black dash-dot**) performs the best on both the metrics in all the cases.

Experiment 1 The aim of the experiment is to quantitatively evaluate the influence of n in the $top - n$ titles used in creating the web context used for a document.

Experiment 2 The aim of the experiment is to quantitatively compare the performance of the three approaches for keyword extraction (Rake, TextRank and Alchemy).

Experiment 3 The aim of the experiment is to quantitatively evaluate the impact of the web-context in improving the performance of keyword extraction techniques. The impact is evaluated by comparing with baselines.

3.8.4 Results and discussion

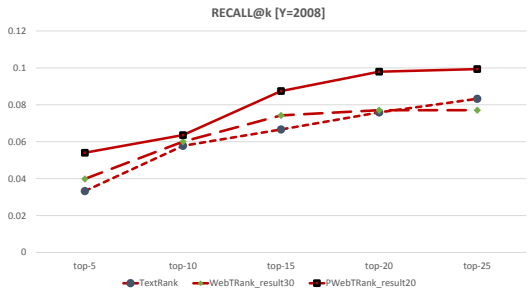
Figure 3.10(a) (b) shows the variation of recall@k for different values of n. As shown in this figure, the recall@k for k={5, 10, 15, 20, 25} is most higher when n=20,30. For other values of n, the recall is not consistently higher. It varies drastically when n=10. With n=50 also, the recall@k is not consistently high. Similar trends are observed in both the plots. From this analysis, we observe the variation of recall on the values of n selected to create the web-context. The analysis helps in determining the stable values of n that can be used for creating the web-context for optimal performance. For the experiments hence, we use n=20,30 for creating the web-context.

Figure 3.11 (a-f) shows plots for quantitative comparison of Rake, TextRank and Alchemy keyword extraction approaches. Plots (a,c) show the comparison via the recall@k metric. Plots (b,d) show the comparison via the precision@k metric. The value of k is in the range $\{5, 10, 15, 20, 25\}$. The results are shown for different document content (only abstract and abstract+web-context). These figure show that the Alchemy keyword extraction technique (shown in **black dash-dot**) performs the best on both the metrics in all the cases.

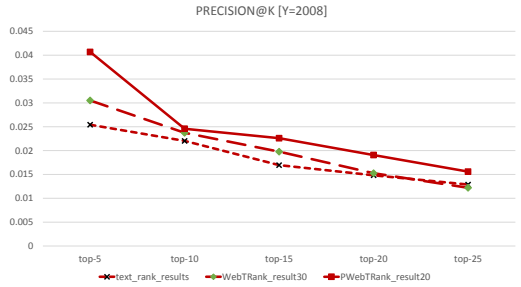
As shown in the figure, the performance of the Alchemy technique is significantly higher than the Rake and TextRank techniques. These results shown in the figure are for $dataset_1$. However, the results for all the test datasets were consistent with these findings. This experiments reveals that the world knowledge based approach i.e. Alchemy technique is the best approach for keyword extraction. These results clearly demonstrate the advantage of incorporating information from the world knowledge corpus like Wikipedia to improve keyword extraction. Comparatively, the TextRank approach is clearly performing better than Rake in abstract+ Web context based keyword extraction. Therefore, in further analysis we will consider only TextRank and Alchemy techniques to monitor the impact of using web-context in addition to the summary text of the documents. findings in Rose et. al [2].

Figures 3.12 and 3.13 show the precision and recall results for comparing the performance of the proposed approach with the baselines for all the keywords extraction techniques for $dataset_1$ (KDD 08) and $dataset_2$ (KDD 12).

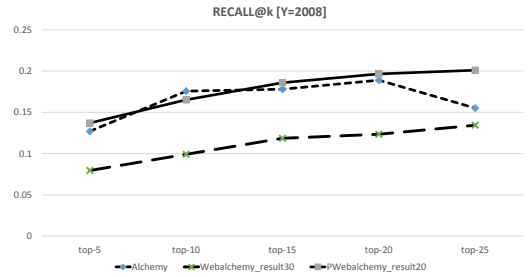
As shown in figure 3.12, the analysis is done to quantitatively evaluate the impact of adding web context to the summary text to enhance various keyword extraction approaches. In the plots (a) and (b), we compare the performance of the TextRank algorithm by varying the content used for keyword extraction. As shown in these plots, the proposed approach to add web



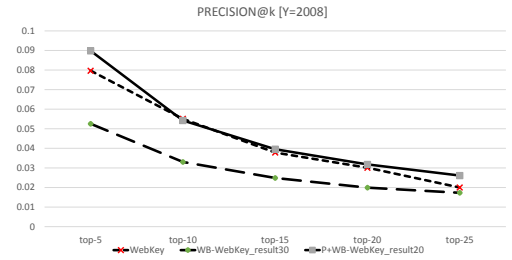
(a) Recall@k for TextRank.



(b) Precision@k for TextRank.

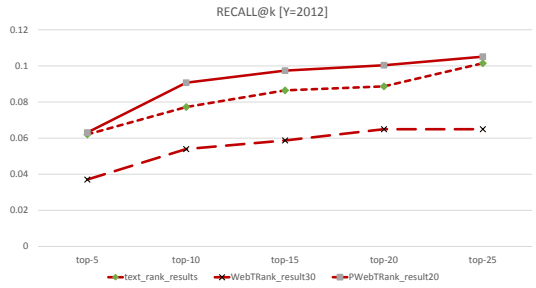


(c) Recall@k for Alchemy.

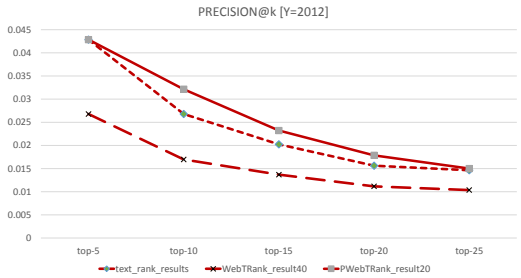


(d) Precision@k for Alchemy.

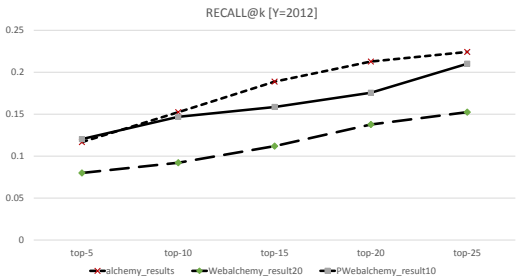
Figure 3.12: Quantitative comparison of adding web-context to the local content for $dataset_1$ (2008). The figure shows the comparison using recall@k and precision@k metrics for TextRank and Alchemy techniques.



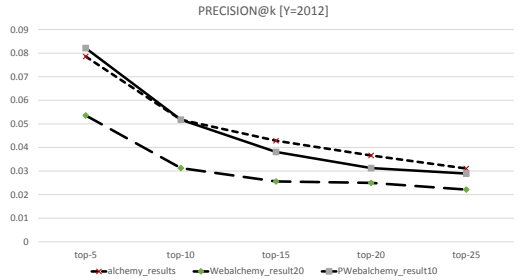
(a) Recall@k for TextRank.



(b) Precision@k for TextRank.



(c) Recall@k for Alchemy.



(d) Precision@k for Alchemy.

Figure 3.13: Quantitative comparison of adding web-context to the local content for $dataset_2$ (2012). The figure shows the comparison using recall@k and precision@k metrics for TextRank and Alchemy techniques.

context to abstract (Abstract + web-context) gives a 57% improvement in recall@5 and 60% improvement in precision@5. The proposed approach is shown with **solid square marked red** curve. The baseline is shown in **short dashed red** curve. From the analysis in the previous experiment, we know that the performance of the TextRank is better than the Rake algorithm for the datasets used in this work. Therefore the improvements obtained using the proposed approach are significant.

In plots (c) and (d), we compare the performance of the Alchemy technique by varying the content used for keyword extraction. The **solid and square marked black** curve denotes the proposed approach and the **short dashed black** curve denote the abstract only baseline. As shown in the figure, the web-context improves the recall@5 by 11.3% and recall@25 by 33%. The precision@5 is also improved by 12.5% by using the web-context in addition to the text in the abstract of the documents.

These results on $dataset_1$ show interesting results demonstrating the significance of adding web-context to improve keyword extraction. However, we validate the performance of the proposed approach for $dataset_2$ as well. Figure 3.13 shows the results for $dataset_2$. In plots (a) and (b), we compare the performance of the TextRank algorithm by varying the content. The **solid red and square marked** curve denotes the performance of the proposed approach while the baseline (abstract only) approach is denoted by **short red dashed and diamond marked** curve. As shown in these plots, the recall@10 is boosted by approximately 17% and precision@10 is boosted by approx. 10%. These results demonstrate the significance of using the web-context to improve keyword extraction using the TextRank algorithm.

In plots (c) and (d), the performance of the Alchemy technique is compared by varying the content information. The performance of the proposed approach is demonstrated by a **solid black and square marked** curve while the baseline (abstract only content) is denoted by **short black dashed and cross** marked curve. The recall@5 shows an improvement of only 3% and the precision@5 improves by 4.6% over the baseline. The recall and precision values at k=10 are comparable for both the baseline and the proposed approach.

3.9 Conclusions and future work

In summary, there are three main conclusions in this work. Firstly, we showed that the proposed keyword selection approach using only the title information of the document performs comparable with the baseline approach using the full text content of the document. The comparison was done using recall@k metric. Secondly, the fully automated approach for keyword assignment performs better than the baseline approach in terms of the Jaccard index comparison. The proposed approach is atleast 3 times faster than the baseline approach using the full text content of the document to assign relevant keywords. There are several areas in this work to extend in the future. On of the areas of improvement in the current work is the de-noising algorithm which uses hierarchical clustering to pruning. However, hierarchical clustering has its limitations and it is worth to explore other algorithms such as density based clustering and some other novel anomaly detection algorithms. We would also test the proposed approach for other document corpus like news, patents etc.

Finally, we proposed a novel approach to improve the performance of the keyword extraction techniques. We performed several experiments to evaluate the improvement in the performance of popular keyword extraction techniques by appending the web-context to the abstract of the document. By adding the web-context to the abstract of the document, we find that recall@5 and precision@5 are boosted by (approx.)60% for the TextRank algorithm. For the Alchemy algorithm, the proposed approach enhances the recall@5 by 11.3% and precision@5 by 12.5%. The performance was tested on different datasets.

Chapter 4

Automating annotation for research datasets

With their broad real world applications such as in security [73], social networks [74, 75], recommendation systems [76, 77], medicines [78]and climate [79, 80], development of efficient data mining and machine learning techniques have become a hot area of research in the present time. The main factor driving the diverse research in data mining and machine learning is the availability of multitudes of public datasets on the web which was not so abundant a decade ago. By providing free access to datasets from various domains, several interdisciplinary scientists have benefited by utilizing these datasets for generation and validation of their hypothesis and algorithms. Several un-validated theories can now be tested on real world datasets. Recently, Backstrom et al[81] re-evaluated the famous 'six degrees of separation' hypothesis over the Facebook network, the largest available network of human relationships, and found that this degree has now shrunk to just 4.74.

As shown in the above example, to answer the research questions of inter-disciplinary nature finding a suitable dataset is of prime importance. While in several cases the quality of development of any algorithm or hypothesis is enhanced if their initial choice of datasets is apt. However, the excessive volume of research datasets available and lack of consistent schema to summarize a research dataset has made it difficult for the search engines to search for the datasets of interest. In most cases a user wants to explore a small set of datasets related to his research interest by getting rough idea about their various properties, application and other

related descriptions to understand their utility in their work. This is very similar to literature review where the reader first looks at a short summary or some annotations of a paper before going deep into the content. Moreover, a structured and semantic version of the annotations is highly useful for search engines to search research datasets from full collection of all existing datasets. Just like documents, annotated datasets become indexable and hence searchable.

In this chapter, we study the problem of automatically generating structured semantic annotations for research datasets. The idea about structured and semantic annotations can be understood with an example of a dictionary entry.

Example 4.0.1 *An example of dictionary entry "data"*

- ***Definition of term***

  1. *information output by a sensing device or organ that includes both useful and irrelevant or redundant information and must be processed to be meaningful*
  2. *information in numerical form that can be digitally transmitted or processed*

- ***Examples of the term:***

  - *Smith,.. mixes accessible summaries of social-science data ..... children.*

- ***Origin of term***

  Latin, plural of datum (see datum)

  First Known Use: 1646

- ***Other computer-related terms***

  adware, flash, kludge, phishing, recursive, router

- ***Rhymes with the term***

  beta, eta, theta, zeta

The above example gives a structured information of the term "data" by having different categories for providing description of the term. In the above example the annotation is structured in five categories namely, definition, usage of the term, origin of the term, related terms and

rhyming terms. Within all categories some semantic description is provided for the term. Similarly, structured and semantic information for research datasets will greatly help researchers. Example 2 shows an example of annotation for a research dataset.

Example 4.0.2 *Example of research dataset:"LiveJournal"*

- ***Data type:***

  1. *directed*
  2. *undirected*

- ***Concepts:***

  Communities, Social networks, Sociology, User profile

- ***Semantically similar datasets:***

  Orkut, Friendster, DBLP, Slashdot, Epinions

In the above example, the "LiveJournal" dataset is summarized in three categories, namely, the data type, the concepts and datasets similar to this dataset. However, there can be innumerable ways to design a schema for summarizing the dataset, the above stated schema for semantic annotation captures a broad spectrum of information about the dataset. While it might also be of interest to know about the numerical properties of the dataset, the purpose of semantic annotation of datasets is to provide used with a high level summary of the dataset avoiding the low order details of the dataset.

Despite the importance of the problem, to the best of our knowledge, the proposed problem has not been well studied in existing literature. As mentioned above, the purpose of semantic annotation is to provide a high level summary of the dataset, the approach therefore does not require to know about the actual content of the datasets. With the web intelligence from open sources of knowledge such as academic search engines and crowd sourced knowledge bases, the proposed framework can automatically generate semantic annotations under a pre-defined schema. Starting with the dataset name, we propose a general framework to extract context information and generating semantic and structured annotations for the dataset. Our approach consists of three components:1) labelling data types 2) generating concept descriptors and 3) ordered set of semantically similar datasets. We have evaluated the proposed approach on two real

world datasets. Experimental evaluation shows the effectiveness of the approach for generating semantic annotations for the test datasets. We also provide a case study where we use queries to compare results from Google web search with the search over the annotated databases.

4.1 Problem description

In this section, we formally define the problem of semantic annotation of datasets.

Let $D = \{d_1, d_2, ...d_n\}$ be a list of 'referred as' name of n datasets. For example, in the example 1.2 the 'referred as' name of the dataset is "LiveJournal". Although LiveJournal is a name for a social networking site but for the purpose of scientific research the datasets is also referred as LiveJournal dataset. For the purpose of this work we will refer to a dataset only by its name and not its actual content.

The research task is to annotate each d_i in D with the following structured semantic information:

Data type label: Let the list of possible data types labels for datasets in D be defined as $T = \{t_1, t_2, ..., t_k\}$. The problem of assigning data type labels to a dataset d_i is to select a set of labels from T such that each label is a data type descriptor of d_i dataset.
Example: The data type for famous "reuters" dataset is 'text'. Similarly, the data type for "iris" dataset is 'multivariate'.

Concept generation : The problem of concept generation is to find k descriptor terms for d_i dataset to describe the application concepts for the dataset d_i. These terms are n-grams where $n\epsilon\{1, 2, 3, 4...\}$.
Definition 1(Application): A specific use of the dataset in academia or industry
Example 1: 'Iris' flower dataset is used for validating classification algorithms. Thus classification is an application of the 'Iris' dataset
Example 2: 'Twitter' dataset is used for sentiment analysis, social networking, sociology, text mining. These are the application of 'Twitter' data.

Similar datasets assignment: Given the list D, the problem of extracting semantically similar datasets is to define similarity measures between the context of two dataset (d_i, d_j). For each d_i, similar datasets in D are extracted based on the similarity measure.

This problem is challenging in various aspects. Firstly, there is no universal schema for describing the content of a dataset therefore it is very hard to extract information from the

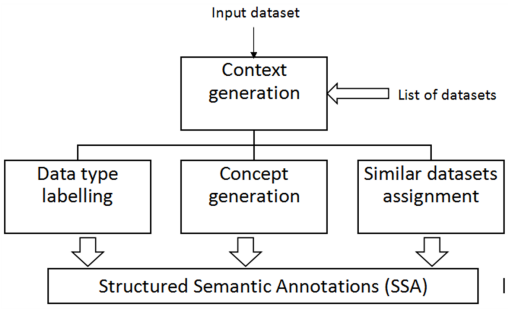


Figure 4.1: Overall framework of the proposed approach.

content of the dataset for semantic annotation. The semantic annotation have to be derived using only the dataset name which, in general, are available for most public datasets. Secondly, there is no well known structure for representing the semantic annotations for research datasets. The proposed structure for semantic annotation has to positively impact the dataset summarization for user's search.

4.2 Proposed Approach

In this section we discuss the proposed approaches to the three sub-problems discussed in the previous section. Figure 4.1 shows the overall framework of the proposed approach. As mentioned earlier, the only other source of information to derive semantic annotation for research datasets is its 'referred as' name. The content of the dataset is not easy to use to generate annotation since there is not universal schema to represent the content of the dataset. In absence of any universal schema for representing content of the dataset, developing an automated framework for annotation is extremely challenging.

In this section the approach to extract information only from the dataset name is discussed. The claim here is that the dataset name along with intelligence from web sources (academic search engines and crowd sourced knowledge sources) will generate information relevant for automating annotation for research datasets. In the following subsections, we describe the approach to generate information using only the dataset names and then turning this information for generating semantic annotation for various components of the semantic annotation structure.

4.2.1 Context generation

As described earlier, the main challenge for annotation of datasets is derive relevant information using only the dataset name. In order to generate information for a dataset with 'referred as' name d_i the following web intelligence was used. The web intelligence is used through academic search engines.

The information is derived in the following manner. Given a dataset name d_i and access to the database of an academic search engine, the required information is extracted from the search results when the dataset name d_i is given as the query input to the search engine. The web intelligence of search engines is particularly useful to generate relevant information for the following reason. The database used for search in academic search engine consists of research articles from various conference proceedings, scientific journals, book and newsletters. If the input query (the dataset name) is unambiguous then the intelligence of search engines can be used to extract all the documents which are relevant for the input query. For datasets, the motivation for using academic search engines is to find research documents which have referred to the dataset in some form. It is expected that the top-k results of such query consists of research documents which use the input dataset (referred by its name d_i). The information extracted using the search engine's results consists of the following text information from the results: (1) the titles/headings of the returned documents and (2) the snippet text provided under the heading. The text information in (1) and (2) for the top-k results are converted to a bag of words model for a input dataset name d_i. As a basic natural language pre-processing step, all the stop words, special characters and numbers are removed from the bag of words. Both the information sources (1) and (2) were given equal weights in the bag of words representation of the information for a dataset.

For convenience, we will refer to this information generated for a dataset name d_i as the context $C(.)$ for d_i. Next, we describe the different components of the proposed approach for semantically annotating a dataset using the context $C(.)$.

4.2.2 Identifying data type labels

In this section we propose an algorithmic approach to identify the data type labels for a given dataset name. Given a list of data type labels T and a dataset name d_i, our goal is to assign a subset of type labels from T to the dataset name d_i.

Given the context $C(.)$ for a dataset name d_i, assigning a subset of labels from T is posed as multi-label classification problem. The problem of multi-label classification is to assign multiple target labels to each instance in the data. Given a feature set for a dataset instance, the classification task is to find a subset of labels of data types from T for the dataset instance d_i. A detailed description about the multi-label classification can be found in a recent work by Tsoumakas et al [82].

The first step in any classification problem is to construct a feature set to represent the each instances in the data. As described earlier, the dataset name d_i is represented with its context $C(.)$. In order to construct features for a dataset name d_i, the bag of word representation of the context $C(.)$ is converted into two vector space models, namely, a bag-of-words(BOW) representation and term-frequency inverse term frequency representation(tfidf) [57]. The dimension of the resulting feature sets (BOW and tfidf vectors) is reduced in an unsupervised manner using the principal component analysis(PCA) technique. The reduced feature set consists of the first k most significant principal components which explain 98% of the total variance of the original feature set.

For this work, the multi-label classification is carried out using AdaboostMH algorithm available in Mulan library[82]. Using Adaboost model gives an advantage in the case of the class imbalance problem in the dataset. AdaboostMH is an extended version of the Adaboost algorithm using binary relevance transformation for the multi-label data. The classification was performed using 10-fold cross validation technique.

4.2.3 Concept generation

In this section, we discuss our approach to model concepts for a given dataset d_i. Unlike the data type labelling of datasets, there is no prior information regarding the labels which can be used as concepts for the datasets. Therefore we propose an unsupervised approach to this problem.

In the unsupervised setting, we extract the concept information from open sources like OpenCyc, DBpedia, Freebase, Yago and other linked data sources. The information about concepts is stored in the following manner in the above mentioned databases. For example, DBpedia, Freebase and Yago contain information about people, companies and concepts in a structured manner. DBpedia stores information from Wikipedia in a structured manner to answer complex queries on Wikipedia data. Freebase stores crowd-sourced information about entities and concepts using a graph model. Instead of using tables and keys to define data

structures, Freebase defines its data structure as a set of nodes and a set of links that establish relationship between the nodes. This enables complex modelling between individual elements. Yago, on the other hand, is a combination of Wikipedia and WordNet. It contains more than 120 million facts about various entities and concepts with spatial and temporal dimension attached to these facts and entities.

Given the context $(C(.))$ for a dataset name d_i, the research task is to extract relevant concepts from the aforementioned knowledge bases. Based on the word frequency distribution in the context $C(.)$ of d_i different concepts can be extracted from the knowledge bases. In order to extract a unique list of concepts for a dataset name d_i from its context $C(.)$, AlchemyAPI[3], an automatic natal language processing tool, was used. The advantage of using this tool is that it generates a ranked list of concepts by combining results from the above mentioned linked data sources. However, the returned list is not the final list of relevant concepts. The next step is to prune our irrelevant concepts.

The pruning was done in two steps. In the first step, the list is refined by pruning out the concepts that were used in the original query to create the context $C(.)$ for a dataset name d_i. If the concept list contain (1) the dataset name itself, (2) the term 'data', or (3)the term 'machine learning', these terms were removed from the concept list. The second step of pruning involves removal of concepts containing named entities. As described earlier, the knowledge bases contain structured information about named entities which might match to the words in the context $C(.)$ of any d_i. The removal of named entities was done using the named-entity tagging feature of the knowledge bases like Freebase adn Yago. Given a term (concepts), AlchemyAPI was used to recognize if the term is a named entity in the following categories: 'person','company','organization' or 'job title'. Any concept which is tagged with these named-entity categories were pruned out from the list of concepts. The final concepts list for a dataset name d_i is obtained after the two pruning steps.

4.2.4 Finding similar datasets

In this section we describe the task of finding a set of similar datasets for a given dataset name d_i. Given a list of dataset names D and the targe dataset name d_i, the research task is to find the k(a parameter) most similar dataset names from the given list D for the target dataset name d_i .

In order to find similar datasets, we first define a similarity metric $sim(C(.), C(.))$ for finding similarity between two datasets. Let d_a, d_b, d_c be three datasets in D and $C(d_a), C(d_b), C(d_c) \in$

V_k be their context representations as described earlier in the proposed approach (V_k is vector space model). Let $sim(C(.), C(.)) : V_k \times V_k \longrightarrow \Re^+$ be a similarity function of two context vectors. We say that d_a is semantically more similar to d_i than d_b and d_c w.r.t. $sim(C(.), C(.))$, if $sim(C(d_i), C(d_a)) > sim(C(d_i), C(d_b))$ and $sim(C(d_i), C(d_a)) > sim(C(d_i), C(d_c))$.

In the information retrieval literature, the cosine similarity is a widely used measure to compute similarity between two given vectors. It is well explored in several applications such as measuring document similarity when the documents are represented as a vector space model. In this work, we use the cosine similarity of two global context vectors to find the semantic similarity between datasets represented by the vector space model of their derived global context. The cosine similarity for two contexts is formally defined as:

$$sim(C(d_a), C(d_b)) = \frac{\Sigma_{i=1}^{k} a_i * b_i}{\sqrt{\Sigma_{i=1}^{k} a_i^2} * \sqrt{\Sigma_{i=1}^{k} b_i^2}}$$

where $C(d_a) = \langle a_1, a_2, ..., a_k \rangle$ and $C(d_b) = \langle b_1, b_2, ..., b_k \rangle$.

Having defined the semantic similarity function, we now discuss the approach to find k most similar datasets for a given dataset d.

We have used the *tfidf* vector representation for the vector space model representation of the context $C(.)$ for the given dataset d_i. The dictionary for the *tfidf* model is built from the $C(.)$s of all the dataset names in D. Finally, the *tfidf* vector for (C(.)) of each dataset in D and the target dataset name d_i is calculated using this *tfidf* model. For a given dataset d_i, the top k similar datasets are determined by using the semantic similarity between the context C(.) of d_i and the context C(.) of $d_j \in D$ ($i \neq j$). The datasets d_j in D are then ranked in descending order of their semantic similarity with d_i. Top-k datasets are selected from their ranked list to represent the top-k similar data sets for a given dataset d_i.

4.3 Experiments

In this section, we discuss the experimental setup designed to evaluate the performance of the proposed approach. Here we show the effectiveness of the proposed semantic annotation techniques on two real world datasets.

Table 4.1: Table showing cardinality of data type labels for UCI and SNAP datasets.

Dataset	Instances	Label count	Label density	Label cardinality
SNAP	42	5	0.34	1.6904
UCI	110	4	0.275	1.1

4.3.1 Dataset Used

In the following section, we describe in detail the real world dataset used to perform the experiments for annotating datasets. Our datasets comes from two sources, namely, (1) the UCI machine learning repository [20]; (2) the Stanford Large Network Dataset Collection (SNAP) [21]. The details are mentioned below.

(1) UCI dataset: The original repository consists of 244 datasets used in machine learning and data mining research as per June 1, 2013. The datasets are described with the following metadata on the web page [1] : Name, data type, default task, attribute type, number of instances, number of attributes, application area and year. The name refers to the common usage name for the dataset. The data type field describes its characteristic data type (e.g. multivariate, sequential, text etc.). The default task field describes the category of machine learning task in which the dataset is mostly used. Next, the attribute types describes whether each attribute is categorical, integer or real etc. The number of instances, the number of attributes and year are numerical description of the dataset. Finally, the application area field describes the broad area of application of the dataset (e.g. Life Sciences, Physical sciences etc.). For the purpose of our research, we selected only those datasets which consisted of more than three characters in their name field. We use a total of 132 datasets from this collection. The input data used consists only of the dataset names without any other information from their meta-data. Other meta-data information is used as ground truth for validation which is described later. Apart from the metadata information, the repository also provide well curated information about each dataset. This information is also used for the purpose of validation (described later).

(2) SNAP dataset: SNAP is a collection of large graph networks. This collection consists of different varieties of network datasets. The following description is available for each dataset in this collection: Dataset name, Data type, Number of nodes and, number of edges. Apart from the meta-data information, each dataset is also individually described by the owners of the dataset. The input data is constructed from the dataset name information for each dataset.

[1] http://archive.ics.uci.edu/ml/datasets.html

All the numeric strings and strings with less than four character were removed from the dataset name. Our final dataset consists of 45 dataset names. The meta-data information for each dataset is utilized as ground truth for the purpose of validation (described later).

As described earlier, the proposed approach of obtaining the semantic annotations of a dataset has three different components in it. The semantic annotations were generated for datasets contained in the UCI and SNAP dataset collections. In the following section we describe the experimental setup, ground truth, the baselines used and the evaluation metrics used for validating the proposed approaches for structured semantic annotation.

4.3.2 Experiments for data type labelling

First we describe the experiments and evaluation of the data type labelling problem.

Experimental setup

Section 3.1 describes the algorithm for identifying the data type labels for datasets. The experiments for data type identification for SNAP and UCI are performed separately. As mentioned earlier, we have used two different models (*tfidf* and BOW) of vector space representation of the context of each dataset.

Table 4.1 shows the information about the two datasets used in classification. As shown in the table, the label cardinality and label density [82] are higher for the SNAP data in comparison to UCI data. In the UCI data we find a high skew in the distribution of instances on the labels for one of the labels. There are 5 labels for SNAP data: {'directed', 'undirected', 'temporal', 'content','reviews'}. The UCI data consists of 4 labels : {'multi-variate','content', 'temporal/time series or sequential','univariate'}. The SNAP data consists of 30 principal components in the BOW feature set and 33 principal components in the *tfidf* feature set. The UCI data consists of 100 principal components both in BOW and *tfidf* feature set. The classification was performed using 10-fold cross-validation using the AdaboostMH classifier.

Ground truth

The ground truth in this case is the original data types for each dataset. As mentioned earlier, both the UCI and the SNAP datasets provide the meta-data information about the data type

of each dataset. Thus each instance was labeled with the original labels available from these datasets.

Baseline

The baseline used in this case is the zeroR classifier. ZeroR is the simplest classification method which relies on the target and ignores all predictors. ZeroR classifier simply predicts the majority category (class). Although there is no predictability power in ZeroR, it is useful for determining a baseline performance as a benchmark for other classification methods.

Evaluation metrics

In order to evaluate the performance of the multi-label classifiers we have used typical metrics used in multi-label classification [82]. We have used 17 metrics to evaluate the performance of classifiers.These metrics are shown in table 4.4.

4.3.3 Experiments for concept generation

Secondly, we discuss the experiments performed to evaluate the efficiency of the concept generation approach for the UCI and SNAP data.

Experimental setup

The task of validating the concept terms is non-trivial. As mentioned earlier, the concepts are generated in an unsupervised manner with no prior information about the application categories. In order to validate the results of the proposed approach, the concepts generated for each dataset in the UCI and SNAP collection are compared against a baseline using Turing type human subject evaluation. For this user study, 6 users[2] were provided with the results of both the proposed approach and the baseline for the UCI and the SNAP datasets. In the interest of the users, the total dataset instances for both the datasets (UCI and SNAP) were partitioned into 6 segments. Each user was given only a single segment for evaluation[3] . For each dataset instance the user was shown three sets of information :(1)the dataset name (hyper linked to the web); (2) results of the baseline, and; (3) results of the proposed approach. It should be noted that the

[2] users are graduate student at University of Minnesota

[3] With the limitation of users to evaluate only a few results, this technique was adopted

Table 4.2: Table showing the meaning of different user ratings.

User rating	User's expression/intent
0	None of the results are satisfying.
1	Both are equally satisfying, hard to compare.
2	Algorithm 'A' is better.
3	Algorithm 'B' is better.

users did not know the source of the results (baseline or proposed approach). The sources were referred as Algorithm 'A' and Algorithm 'B'. The users provided one of the following integer rating (Table 4.2) against each of the dataset instance in their results set.

Baseline

The baseline for comparison is constructed in the following manner. Each dataset is provided with a short description by the owner of the dataset. This description was used to create the context $C(.)$ for a dataset instance. Special characters and stop words are pruned from the text. We have also pruned out the dataset name and the terms which were used in querying the context of the dataset such as the dataset name, 'dataset', 'machine learning'. The refined bag of words represent the baseline context for a dataset instance. The concepts were generated using the approach discussed in Section 4.2.3. The baseline differs from the proposed approach in terms of the context generation. The context for the baseline is derived from the owner's description whereas the context in the proposed approach is automatically generated using web intelligence.

Evaluation metric

The UCI and SNAP data are evaluated separately. In order to compare the results of the baseline and the proposed approach based on the user rating, the average user score(or average relevance score) are computed by averaging the number of user votes in each of the four categories of user ratings. The performance of each of the approaches is compared based on this average scores.

4.3.4 Experiments for similar dataset assignment

Finally, in this section we describe the evaluation methodology for the similar datasets assignment approach. The following are the experimental setup, ground truth, baseline and the evaluation criteria used for the experimental analysis.

Ground truth and experimental setup

There are two different approaches used to construct ground truth for this experiment. For the SNAP dataset, the repository already provides a categorization of dataset in different groups according to their usage and properties. So the ground truth information about the similarity of datasets is obtained from the existing grouping of the datasets. All the datasets in a group are similar to each other but not to the datasets in the other groups. However, there may be cases when a dataset appears in two groups. This issue is addressed by assuming the datasets of each group to be similar to that dataset. This similarity is non-transitive in nature.

For the UCI dataset, there are 4 ways to group the datasets based on the different categories of attributes. Based on the different groupings, the similarity between any two datasets can be determined in the following manner. A dataset d_a is similar to dataset d_b if these two datasets are grouped together in 1 or more groups. Similarly the degree of similarity between two datasets d_a and d_b is 2 if they are grouped together in 2 or more groups. The degree is 3 if they are grouped together in 3 or more groups and the degree is 4 if they are grouped together in all the 4 groupings.

Baseline

Given the list of datasets D, the baseline is constructed in the following manner. Similar to the baseline discussed in Section 4.3.3, the context $C(.)$ for the baseline is generated using the owner's description. Next, the feature vector for computing the similarity score as discussed in Section 4.2.4 is derived from the baseline context $C(.)$. Except for the context generation step, the remaining procedure for computing the similar datasets is same as discussed in Section 4.2.4.

Table 4.3: Table showing results for multi-label classification algorithm for SNAP dataset. The ↓ and ↑ signs indicates lower the better and higher the better marks respectively.

Measure	Zero classifier	AdaBoostMH (BOW)	AdaBoostMH (TFIDF)
Hamming loss↓	0.349	**0.343**	**0.343**
Accuracy↑	0.025	0.066	**0.146**
Precision↑	0.025	0.083	**0.213**
Recall↑	0.025	0.078	**0.158**
Fmeasure↑	0.025	0.074	**0.172**
Subset Accuracy↑	0.025	0.045	**0.050**
Micro Precision↑	0.025	0.200	**0.288**
Micro Recall↑	0.014	0.066	**0.164**
Micro F_1 ↑	0.018	0.098	**0.203**
Macro Precision↑	0.105	0.170	**0.238**
Macro Recall↑	0.120	0.147	**0.243**
Macro F_1 ↑	0.108	0.153	**0.234**
Micro AUC↑	0.636	0.630	**0.637**
Macro AUC↑	0.500	0.550	**0.555**
Coverage↓	2.165	2.005	**2.003**
Ranking loss↓	0.354	0.351	**0.346**
Average Precision↑	0.657	0.660	**0.663**

Evaluation metrics

The evaluation metric used in this experiment is precision@k. *Precision@k* corresponds to the number of relevant results in the top-k retrieved results retrieved. We have average the precision@k metric over all the test instances.

4.4 Experimental results and discussion

In this section, we discuss the results of the above mentioned experiments for the various components of the proposed approach. Following the pattern in section 4.3, we have divided the section into three sub-sections to discuss the results for each component individually.

4.4.1 Data type labelling

Table 4.3 and 4.4 show the results for multi-label classification experiment for SNAP and UCI data respectively. In table 4.3 the three columns corresponds to the different classification scheme. The column titled Zero classifier is the baseline classifier. The other two columns show results for AdaboostMH classifier using the two different feature sets, BOW and *tfidf*

respectively. These classifiers are compared via several metrics used to evaluate multi-label classifiers. As shown in the table, the AdaboostMH classifier using *tfidf* feature set gives the best performance over all the metrics. In comparison to the baseline, we find a 12% improvement in terms of accuracy and 15% improvement in terms of the Fmeasure value. For other metrics also we find that the improvement are significant. Based on these results, it becomes clear that using tfidf vector representation (reduced by PCA) is a useful feature representation for identifying data type labels for datasets.

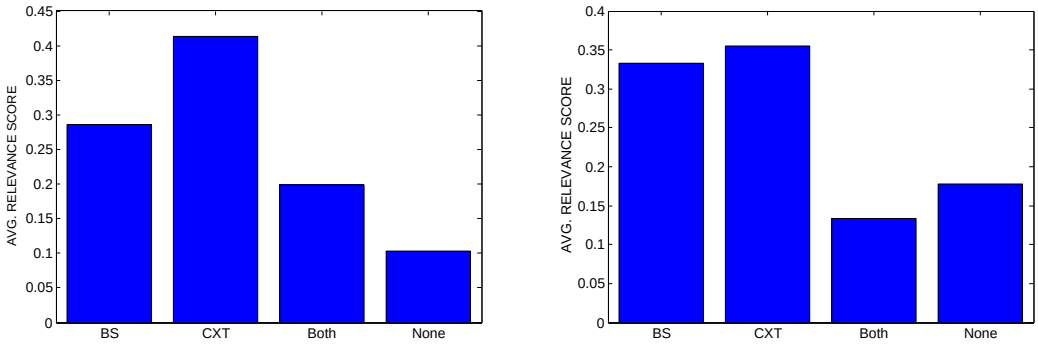
In table 4.4 we show the results of multi-label classification for UCI dataset. Unlike, the SNAP dataset, here we find that the second column(in bold), corresponding to AdaboostMH results using BOW features , gives significant improvements over the baseline classifier. For the UCI dataset, the distribution of the instances over the labels is very skewed towards one label. Therefore we find that the baseline classifier shows seemingly good performance. However, in this table, we show that using the feature set represented by *BOW* vector model gives significant improvement over a zero classifier even when the skew is so high. Using BOW vector representation we get a 5% improvement in Micro AUC, Macro F_1, Macro precision, Macro recall. Using tfidf representation we get approx. 10% improvement in the Macro AUC. These results show the potential in the features derived from the context ($C(.)$) of the datasets to identify data type labels for the dataset instances.

4.4.2 Concept generation

The results of this approach on the UCI and SNAP data are shown in figures 4.2(a) and (b) respectively. These plots summarizes the user ratings for the concepts generated by the baseline(BS) and the proposed approach(CXT). The x-axis shows the 4 category labels to which the users assigned the ratings for the concepts generated for each of the dataset instances. The y-axis shows the average relevance score obtained by each category based on the user ratings. As shown in figure 4.2(a), the concept generation results for the proposed approach is liked by the users for approximately 41% of the total dataset instances in the UCI dataset. In comparison to the proposed approach, the baseline results were liked by the users for only 28% of the total dataset instances in the UCI dataset. Out of the remaining dataset instances, the users selected both the proposed approach and the baseline results for approx. 19% of the total instances. For the remaining dataset instances neither of the approaches satisfied the users' expectation of the concepts for the dataset instance.

Table 4.4: Table showing results for multi-label classification algorithm for UCI dataset. The ↓ and ↑ signs indicates lower the better and higher the better marks respectively.

Measure	Zero classifier	AdaBoostMH (BOW)	AdaBoostMH (TFIDF)
Hamming loss↓	0.084	**0.073**	0.084
Accuracy↑	0.840	**0.858**	0.840
Precision↑	0.883	**0.904**	0.883
Recall↑	0.840	**0.857**	0.840
Fmeasure↑	0.854	**0.873**	0.854
Subset Accuracy↑	0.800	**0.810**	0.800
Micro Precision↑	0.883	**0.904**	0.883
Micro Recall↑	0.803	**0.820**	0.803
Micro F_1↑	0.840	**0.860**	0.840
Macro Precision↑	0.371	**0.426**	0.371
Macro Recall↑	0.400	**0.450**	0.400
Macro F_1↑	0.384	**0.437**	0.384
Micro AUC↑	0.864	**0.917**	0.884
Macro AUC↑	0.500	0.540	**0.596**
Coverage↓	0.473	**0.408**	0.417
Ranking loss↓	0.115	**0.092**	0.098
Average Precision↑	0.908	**0.924**	0.915



(a) (b)

Figure 4.2: User validation results of concept generation experiment on the UCI and SNAP dataset for (a)UCI dataset. (b)SNAP dataset.

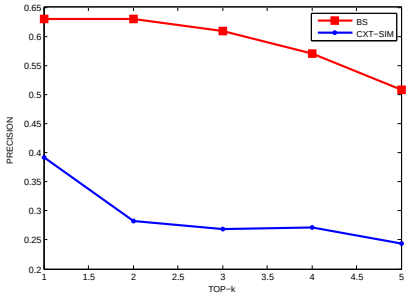


Figure 4.3: The plot shows the comparison of dataset similarity using the proposed approach(blue) and the random baseline (red) for SNAP data.

The results for the analysis on the SNAP dataset are shown in figure 4.2(b). For the SNAP dataset, it can be seen that the results of user ratings are only slightly higher for the proposed approach as compared to the baseline approach. These results are interesting because they provide assurance about the quality of the automatically generated content by the proposed approach. Based on this user evaluation, it becomes clear that the proposed approach for content generation is comparable to the human provided description with respect to the concept generation results.

concepts matching. The random baseline cannot produce more than two matching concepts whereas we find that using the proposed approach we can get upto 6 concepts to match exactly with the ground truth concepts though the proportion of such datasets is smaller. Similarly in figure 4.2(b) we compare the proposed approach(CXT) with the 'random' baseline. For atleast 1 match between predicted and the ground truth, we find that the proposed approach finds it for approx. 50% of the dataset instances whereas the baseline finds this for approx. 42% of the instances. But moving towards the left, the performance of the baseline for atleast 2 match degrades to less than 5% whereas the proposed approach still gives an accuracy of 25%. Similar to the results of UCI data, here also we find that the baseline cannot predict more than 2 exact matching concepts for any instance(0% accuracy) whereas the proposed approach predicts 3 exact match for approx. 10% of the instances. These results show that the proposed approach is efficient in finding concepts for datasets. The comparison with baseline confirms the significance of the predictions over random predictions.

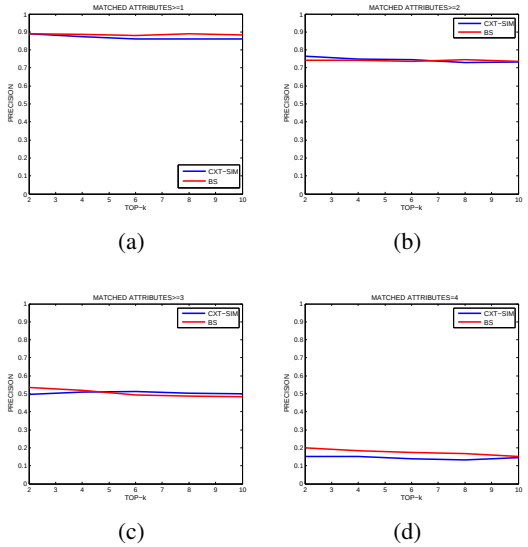


Figure 4.4: Similar dataset validation results for UCI dataset. Plots(a-d) show the comparison of the proposed approach with the baseline for different degrees of matching.

4.4.3 Similar dataset assignment

Figures 4.3 and 4.4 show the evaluation results for SNAP and UCI datasets respectively. The figure 4.3 compares the prediction results of the proposed approach (denoted by CXT-SIM) with the baseline for the SNAP data. As shown in the figure, the baseline approach (BS) outperforms the proposed approach (CXT-SIM) with respect to the Precision@k metric for all values of k. The higher value of precision for the baseline can be attributed to significant overlap between the owner's description for similar datasets. It is expected that the baseline precision is higher because the owner description are repetitive/duplicate for datasets in the same group. Since the grouping of the datasets is done by the owner of the datasets, some of the datasets share the same owner's description. While the baseline outperforms the proposed approach, the precision of 38% for the proposed approach is still indicative of a useful performance.

Figure 4.4(a-d) shows the comparison of the proposed approach (CXT-SIM) against the baseline (BS) for the UCI dataset collection using different degree of matches. As shown in figure 4.4(a), the precision@k is greater than 80% both for the CXT-SIM and the baseline and this trend is decreasing as the value of k is increasing. The high precision in this case is due to the majority of the instances are grouped in a single group. Both the proposed approach and

the baseline results are comparable in this case. Figure 4.4(b) shows the results in the case of 2 degree matching between datasets. The CXT-SIM approach is as good as the baseline approach for all values of k. On increasing the degree to 3, the difference between the performance of the proposed approach and the baseline is not significant (figure 4.4(c)). It should be noted that a higher precision is expected when using the baseline approach since it uses the owner's description as the context $C(.)$. The owner's description by default contains the category terms which were used to define groups for the dataset instances. Therefore the efficiency of the proposed approach is significant if the precision is close to the baseline precision. Finally, in figure 4.4(d) we compare the performance in the case when the datasets match to a degree 4. In this case we find that the precision of the baseline approach is approximately 5% higher than the proposed approach precision for all values of k. However, the difference is not very significant. The following set of evaluation for the UCI dataset collection demonstrates that finding similar datasets using automatic context generation is comparable with the similar datasets when the we use owner's description as context for a dataset instance. Since obtaining owner's description is not a computationally efficient technique, using an automatic context generation technique is computationally efficient with comparable accuracy.

4.5 Use case study

In this section, we discuss some of the use cases of the annotated database generated using the proposed approach. The utility of a database is known if it returns relevant results for input queries. We generated the input queries for research datasets synthetically. Each query is has the following three components: data type, concept and ''similar to" fields. For the SNAP and UCI datasets, the actual labels for the data type field is already known. So for each query the data type field is randomly selected from the pool of known data type labels. In order to generate the concept field of the query, we prepared a list concepts which are closely related to the application areas of the datasets. The concepts field in each query is generated by random selection from the list of concepts. The "similar to" field in each query was obtained by random selection from the pool of all the research datasets i.e UCI and SNAP. A total of 50 queries each was generated to validate the SNAP and UCI annotations. Few examples of the synthetic queries for SNAP dataset is shown in Table 4.5. For the sake of convenience we will refer to these query sets as Q_{SNAP} and Q_{UCI}.

Given the queries, the efficiency of the annotated database was validated in the following manner. Each query in Q_{SNAP} and Q_{UCI} was first searched on the Google search engine and the top-10 results of these queries were archived. For each query, all the dataset names (excluding the dataset name in the query itself) appearing in these top-10 results were extracted. The results for the Google search for a sample set of queries are shown in column 2 of table 4.5. This was done for all the 50 queries and the full results can reviewed here [4] . Secondly, these queries were searched on the annotated database. The search task was performed by means of simple *tfidf* based cosine similarity between the input query and the annotation of the datasets in the database. For an input query, the extracted datasets were ranked in descending order of their cosine similarity. From this ranked list, top-5 datasets are considered as the final output of the query. Outputs for a few queries are shown in the column 3 of the table 4.5. The purpose of this table is to present the subjective comparison between Google search and the search on annotated database. It should be noted that the comparison between the Google search results and the proposed approach is relevant since we are comparing the dataset names extracted by both the retrieval systems.

4.5.1 Qualitative evaluation

In the table 4.5, for the first query, 'twitter' and 'facebook' dataset in the Google results seems to be relevant for the query. However, for the search results on the annotated database 'reddit','beeradvocate' are more relevant because these datasets contain user reviews which can be used for sentiment analysis. The 'pokec' and 'gplus' dataset are similar to slashdot dataset since all them are social network dataset. For the second query, the Google gives only 'dblp' dataset which is not a good fit for the query. The results of the proposed approach contain only 'reddit' as a good fit while the other results are not appropriate for the query. In the Google search results for the third query we find 'stanford' dataset as the only output. However, 'stanford' dataset is relevant for web type of datasets and not for recommendation problems. Using the proposed approach we get 'movies', 'beeradvocate', 'ratebeer', 'cellartracker' and all of these contains reviews which are used in recommendation systems[83]. For the fourth query, we find that the Google search result is 'google' dataset which is relevant for problems dealing with web graphs and the hyperlinks although it is temporal in nature. In the search output of the proposed approach we find 'reddit' which is used for sentiment analysis while the other outputs

[4] https://www.dropbox.com/sh/5tf9mwpa1lohlbn/oZikAVL4go

are temporal in nature. For the last query we find that Google search gives no output while the results of the proposed approach consist of datasets that are used for centrality computation such as 'roadnet' a road network dataset, 'astroph' is temporal collaboration network, 'berkstan' is a temporal web graph.

4.5.2 Quantitative evaluation

In order to make the quantitative evaluation, we use the results of the queries in Q_{UCI}. We have also provided the entire table for reviewing the output for Q_{UCI} queries here[5] . In this experiment the outputs of Google search and proposed approach are tested for relevancy for the input query by comparing the 'context' of the input query (C_{q_i}) with the context of the output datasets (C_{o_i}). The 'context' of the input query consists of the data type and application field of the query and also the ground truth annotations of the dataset in the"similar to" field of the query. Similarly, for the output datasets the context consists of the ground truth annotations of those datasets. The ground truth annotations include the original data type, application type and other meta-data information available from the UCI repository. The relevance score of the output for a query q_i in Q_{UCI} is computed using the cosine similarity between the BOW model vector representation of the query context and the vector representation of the generated output. The relevance score(RS) is:

$$RS(q_i) = \sum_{j=1}^{5} cossim(BOW(C_{q_i}), BOW(C_{o_j}))$$

where $cossim(a, b)$ corresponds to the cosine similarity between two vectors a and b.

Figure 4.5 shows the distribution of the relevance score for the two types of searches (Google and search on annotated database) for 50 test queries. The horizontal line inside the box marks the median relevance score and the edge of the upper and the lower edge of the boxes are the 75th and 25th percentile respectively. The end of the whiskers(horizontal bars outside the box) corresponds to the end data points not considered as outlier. As shown in the figure, the median of relevance score for search on the annotated database is approx. 0.65 whereas it is nearly 0.45 for Google search. This figure clearly shows the efficiency of the proposed approach over the Google search. We also find that several of the queries did not find any datasets. Thus it is clear that the annotations provide great advantage over the general purpose search engine

[5] https://www.dropbox.com/sh/v8xdme937w3pzmm/CppA0an0WW

Table 4.5: Table showing the search output of few synthetic queries for research datasets using a Google search engine and a search engine indexed on the semantic annotations.

Input query for dataset	**Results from Google search(in top-10)**	**Results from semantically annotated database**
find dataset of content type and used in sentiment analysis similar to slashdot data	twitter, google, facebook, flickr	reddit, beeradvocate, gplus, pokec, flickr
find dataset of temporal type and used in sentiment analysis similar to wiki talk data	dblp	reddit, berkstan, oregon, astroph
find dataset of reviews type and used in recommender system similar to amazon data	stanford	movies, beeradvocate, ratebeer, cellartracker, reddit
find dataset of temporal type and used in sentiment analysis similar to amazon meta data	google	reddit, wiki talk, oregon, astroph, gnutella
find dataset of temporal type and used in centrality similar to oregon data	-NA-	astroph, roadnet, wiki talk, berkstan, oregon

even with the entire world database. This experiment also demonstrates the usefulness of the automatically generated annotations by our proposed approach.

4.6 Related work

Annotation of documents or artifacts is studied in two different categories. One branch of literature focuses on manual annotation research. The central idea of research on manual annotation is to facilitate users to write down notes for any artifact in a collaborative manner. The second branch of literature focuses on developing techniques for automating the task of annotation generation. The goal of this field of research is to develop good quality machine readable annotations.

In the manual annotation category Annotea[84] is the representative tool for manual annotation. Annotea enhances collaboration of the users via shared meta-data based Web annotations, bookmarks, and their combinations. This shared meta-data can be used to organize any web document under different topics. Besides Annotea, other manual annotal tools used for semantically annotating the web are: CritLink [85], Futplex[86], ComMentor[87]. However this list is not an exhaustive list.

In between the manual annotation methods and automatic approaches, there are a few semi-automatic approaches/tools which are developed to annotate web pages. Ont-O-Mat[88], MnM[89], S-CREAM (Semi-automatic CREAtion of Metadata)[90] are a few systems which provide annotations in a semi-automatic fashion. Semi-automatic means the annotations are

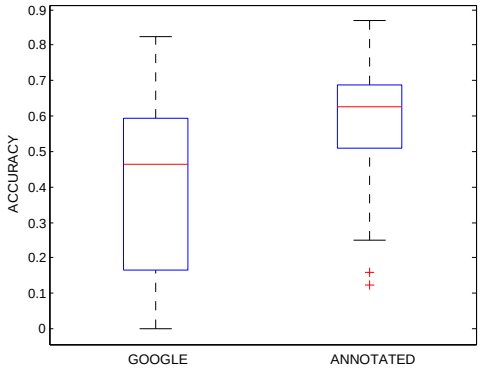


Figure 4.5: Comparison of dataset search using Google search engine, random search and search on annotated datasets. This figure evaluates the relevancy of the various search output with the input queries.

based on automatically generated suggestions.

In the automatic annotation category, systems like KIM[91] and SemTag[92] are quite popular. In such systems the process of annotations involves finding annotations from a given list of ontologies. These ontologies corresponds to several entities like location, music, movies, authors, sports and other popular objects. Thus the goal is to find these entities in the given corpus and map them to their respective ontologies.

Apart from the above mentioned popular tools for semantic annotations, several recent works have explored the application of semantic web in several important domains. In the field of computational biology, researchers have developed techniques to annotate DNAs or genomes to identify the locations of genes and the functions of the genomes. Currently there are several databases like genomic database[93], Flybase[94], WormBase[95] and a few others which store the annotation of biological data. In medical imaging community, the annotation task is to find the region of interest in the image[96]. Annotations is also studied under the category of tagging and this research is popular for sites like Flickr, a photo sharing site[97]. Interestingly, techniques for automatic annotation generation for patterns in transaction data logs have been proposed in a work by Mei et al.[98]. In a recent work, Zhang et al. [99] describes an efficient approach to manually annotate document pertaining to archaeological domain.

4.7 Conclusions and future work

In the current data mining practices, real world datasets play a very important role to test and validate hypothesis and performance of the algorithms. Especially, with the online world the amount and varieties of datasets have become humanly unmanageable. While the popular search engines provide access to documents by searching for key terms within documents, it is for search engines to find research datasets because of lack of annotations.

In this chapter, we propose a novel problem of semantic dataset annotations (SDA)-generating structured and semantic annotations for research datasets. A semantic annotation consists of three components- a set of characteristic data type descriptors, a set of application context descriptors and a set of semantically similar datasets. We propose algorithms to exploit the global context for a dataset to generate the above mentioned annotations. The global context was extracted from the World Wide Web corpus using an academic search engine.

We evaluated our proposed approach on two real world datasets. The results show that the proposed approach generates semantic annotations for datasets effectively. Since the approach is based on generating the global context for a dataset from the web, this can be used to generate annotation for objects with low text content. The strength of the approach lies in the utilization of the web based global context and generate structured annotations using this context. The annotations are usable for the purpose of searching a research dataset suitable for user's interest.

Although, the semantic annotation approach provides useful annotations yet the annotation may not give an exhaustive representation of a research dataset. A major goal of future work is to identify an overall encompassing structure of annotations which can provide full information about the research dataset. We faced several challenges while evaluating the performance of the proposed approach due to lack of standard ground-truth data. In the future, we plan to extend the analysis to dataset of different domains and develop ground truth data for these domains to make it useful for future researchers in their research for research dataset annotations.

Chapter 5

Annotation expansion for low text content items

The search-ability of a resource over the web depends heavily on the content of the resource on the web. Often the resources require additional content (meta data) or description for efficient retrieval, categorization and information management over the web. Tagging or assigning keywords is one of the most popular approaches for creating useful content or semantic descriptions for such resources over the web[100]. For example, tagging text documents is used for categorization (clustering) of documents for information management in the web[101]. Similarly, tagging multimedia items such as images, songs, and videos using online tools has led to the development of specialized search engines like Google image, Bing images etc.

With the popularity of various tagging or bookmarking sites such as Delicious[1] , Bibsonomy[2] , Flickr, YouTube and similar other social activity e-sites, several web resources are appended with meaningful information in the form of tags. Images, videos, songs and similar multimedia items are the most common resources which are tagged either manually or in a semi-automatic manner. In general the content structure of such resources is easy to interpret and hence the tagging process is easier. However, the tagging process becomes complicated when the content structure of a resource is not easily interpretable (e.g. visual or audio features). Such a problems occurs in resources like scientific research datasets or documents with

[1] https://delicious.com/
[2] http://www.bibsonomy.org/

very little text content. While manual tagging by experts is desirable but not feasible, automating tag expansion with minimal content information is a desirable alternative.

In this chapter, we study the problem of tag expansion for low text content items on the web e.g research datasets. The problem of tag expansion is related to the generation of new tags given some user provided initial tags for an item. As mentioned earlier, one of the main challenges in automating tag expansion for such items is that of the lack of usability of the content of the resource. So the task of tag expansion has to completed independent of the item's raw content. In order to alleviate this problem, we leverage intelligence from the World Wide Web to create a secondary content for the purpose of selection new tags for the resource. The generalized framework overlooks the actual content of the resource and still generates a list of relevant tags. In the proposed framework, we remove the noisy tags from the list of predicted tags using web-distance based clustering. Removing noisy tags help to minimize the problem of topic drift i.e. removing tags which do not represents the theme or topic of the items.

The effectiveness of the proposed approach is validated against a real world dataset consisting of a collection of scientific research datasets and their corresponding user assigned tags. The performance is evaluated using both quantitative and qualitative comparisons with several baseline approaches.

The baselines consist of Wikipedia based nearest neighbor approach (WikiSem) to find related tags and tag prediction using non-negative matrix factorization (NMF) based collaborative filtering approaches. For the single tag prediction experiment, we find that the proposed approach (MRR=0.27) is at least twice as accurate as the WikiSem baseline (MRR= 0.13) and the NMF baselines (MRR=0.116) in terms of the MRR values. A user study about the quality of predicted tags support out quantitative findings. We find that the proposed approach is performing better than the baselines both in terms of the NDCG@k and MAP metrics when multiple new tags are predicted. We also find that approximately 50% of the test instances were given very high ratings by the human subjects for the quality of the predicted tags. One of the main reasons for the proposed approach performing better than the baselines is the hybrid nature of the framework that overcomes the limitation of the WikiSem baseline (nearest similar tag without incorporating information about the relationship between items) and NMF baselines which do not use any information about relationships among the tags. We also analyzed the performance of the noisy tag removal technique and found that we can remove 10-17% of noisy tags within an error rate of 1.5%.

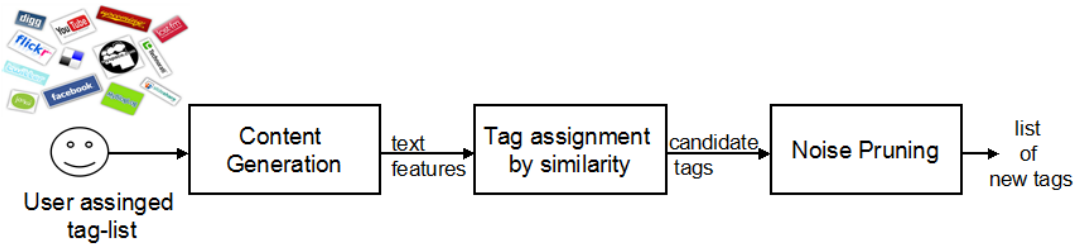


Figure 5.1: Overall framework of the proposed approach.

5.1 Problem Setting

In this section, we formalize the problem of tag expansion. Given a list of user given tags $L_i = \{t_1, t_2, .., t_K\}$ for an item (i), the problem of tag expansion is to find a list of new and relevant tags $N_i = \{n_1, n_2...n_m\}$ for the item (i). K and m are not known apriori.

relevant: A tag is relevant if it summarizes the content information of the item. In case of a URL, the tags could summarize the properties of the web pages.

For example: *Wikipedia.org* can have the following relevant tags- graph; links; text; concepts; encyclopedia.

5.2 Proposed Approach

In this section, we discuss the proposed approach to the tag expansion problem. As shown in figure 5.1, there are three main components of the proposed approach. Given an items (i) along with user assigned tags, the first step is to generate a secondary content information using the initial user provided tag list. In the next step, this content information is used to generate a list of candidate new tags. In the final step, we use a web-distance based clustering technique to remove noisy tags from the list of candidate tags. These steps are discussed in detail in the following sections.

5.2.1 Secondary content generation

In a manual tagging system, one of the important requirements about the item to be tagged by the user is the visible content of the item. The users provide tags after interpreting the content of the item. For tagging images and videos, the content is available in form of the visual or

audio features. Similarly, in document tagging users can read the text content to provide tags. However, the problem of tagging becomes challenging when the content of the item is not easily interpret-able or very minimal as explained earlier. The first step is tag generation for such items is content generation. In this section, we describe a fully automated approach to generate secondary content for such items.

Given the user assigned tags/labels L_i for an item (i), the secondary content(text features) for i is generated using information from the web. The web corpus provides the universal source of all the text content. In this work, we have used the search engines to mine information from the web corpus to generate the secondary content for the items. The information from the search engine is used to create text features which is used to find related tags (described in the following section).

As illustrated in figure 5.2, the text features for the item i are generated using its user given tags(L_i) as a input query to a search engine. In the figure, the input query is the tag list "api, corpus, dataset" (the tag "dataset" is added to the original list of tags since the item under consideration is a dataset). The text features are generated from the results of the query on the search engine. The text features are constructed from the (1) the 'title' of the retrieved results for the input query; and (2) the 'snippet' text of the retrieved results. The items in the results that are used to create text features are highlighted with a red box (in the figure). Other items such as the URLs are not included in the text features. The final text features are generated by pre-processing the extracted information using the basic data cleaning process such as HTML tag stripping, text tokenization, stop wording, uniform case representation and non-alphabetical character removal. We refer to this unstructured text features as the secondary content of an item i and it is denoted $C(i)$ in the rest of the paper.

We used two different search engines to generate the text features, namely, Google search engine and the Google Scholar search engine. The query session were in terms with the terms of service of these systems. For both the databases, we have used only the top-50 results of search output[3] .

5.2.2 Candidate tag generation

For an item i and its user assigned tag list L_i, the list of new candidate tags can be generated using the secondary content information generated in the previous step. We use a database

[3] Based on a user survey- the best search results are within the top-50 hits.

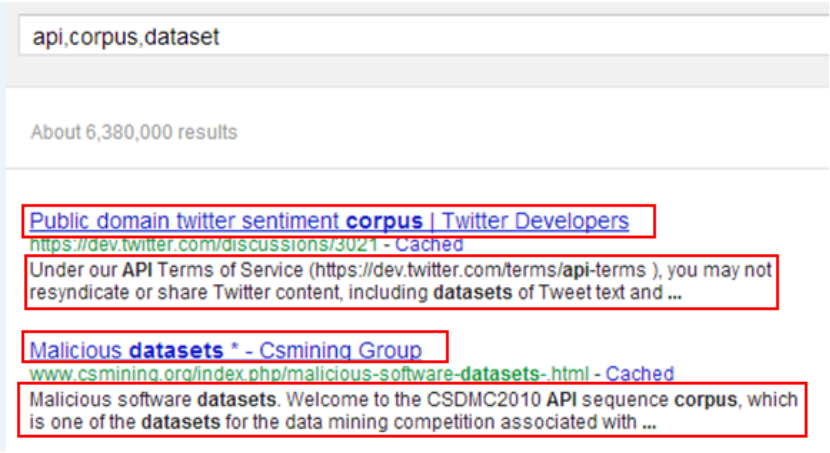


Figure 5.2: An illustration of content generation using Google search engine. The search box shows the input query and the text features are extracted from the information highlighted in red boxes.

consisting of tagged dataset instances. Note that the tagging may or may not be complete for all the dataset instances in the database.

Given the database(DB) of the tagged datasets, the set of new candidate tags for i are selected from the list of the tags available in DB using the following similarity metric. Just as the input item i is represented by its tag list L_i, each item instance in the DB is also represented by its corresponding tag list (L_j, $i \neq j$). The new candidate tags for i are selected from the $\bigcup (L_j)$ of the instances in the DB where $L_i \bigcap L_j \neq \phi$ and $i \neq j$ i.e. the target tag list and the tag list of d_j in DB have some common tags. The final candidate tag list(P_i) is a non-repetitive list of tags generated from the $\bigcup (L_j)$.

The tags in the final candidate list are ranked using the approach shown in figure 5.3. For the input item instance i, a matrix M is constructed from the dataset instances in the database DB. Matrix M is a $n \times k$ matrix, where n is the number of item instances in DB having at least one overlapping tag with L_i and k is the number of total unique tags. The presence of each tag for an item is denoted by 0 or 1 in the matrix. The weights W_i against each item instance in the DB are assigned by computing the similarity score based on their secondary content ($C(j)$ and $C(i)$) of $j \in$ DB and i respectively. The similarity is computed using the cosine similarity score between the tfidf vector representation of ($C(j)$ and $C(i)$). The final weight of each tag(t_i) in the list of candidate tags for the dataset instance d_i is given as $s(t_i)$:

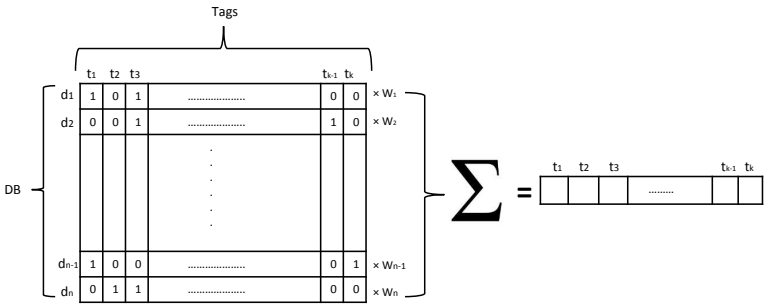


Figure 5.3: The figure shows the computation of weights for the candidate tags for an item. The weights are used to assign a relevance order to the tags.

$$s(t_i) = \Sigma_{y=1}^{n}(M_{y,i} \times W_y)$$

Finally, the tags in the list are ranked in decreasing order of their weights. For future purposes, we refer to the list of candidate new tags for an item i as Q_i.

5.2.3 Removing noisy tags

As a conventional approach, the problem of tag expansion can be easily addressed by ascribing new tags from similar items. However, assigning tags based on similarity between the items can lead to introduction of noisy tags if the similarity computation is not perfect. In the proposed approach, the similarity between the items is computed using the content generated from the web which may introduce some unwanted content in the text features for the dataset instance. One way to overcome this problem is to remove unwanted content before computing similarity score in W_i. The second approach is to prune out noisy tags from the final list of predicted tags. In this work, we describe the latter one in detail.

As a final step, identifying and pruning out noisy tags is an essential step to prevent the problem of topic drift (i.e. the tags do not represent the theme or topic of the item). There are not many automatic tagging approaches that address this problem. Recently, techniques using tag similarity metrics [102] and Folk-LDA [103] were used to deal with the problem of topic drift for the expanded tags. Since these approaches use the content of item (images and documents respectively), they are not directly applicable in the present problem. In this section,

we describe a novel approach to detect the noisy tags from the assigned list of tags without using any content information of the item.

Given the user assigned tag list L_i for an item and the list of candidate new tags Q_i obtained in the previous step, the noisy tags are pruned by using the technique of clustering based on a web based distance metric. In the first pruning step, the pairwise distance between all the tags (the user assigned tag list L_i and the tags in Q_i) is used to cluster the tags into two clusters using the hierarchical clustering technique. The distance function between tags is defined as the Normalized Google distance (NGD) [45] between the tags.

$$NGD(t_1, t_2) = \frac{max\{logf(t_1), logf(t_2)\} - logf(t_1, t_2)}{logZ - min\{logf(t_1), logf(t_2)\}}$$

where, t_1 and t_2 are tags, Z is the total number of web pages indexed by the search engine; $f(t_1)$ and $f(t_2)$ are the number of hits for search terms $t_1$and t_2, respectively; and $f(t_1, t_2)$ is the number of web pages on which both t_1 and t_2 occur simultaneously.

The distance function defined above is a measure of semantic relatedness of any two words. The problem of computing semantic relatedness between two words is addressed in several ways in the literature[104]. Based on a human judgment based evaluation, Cramer et. al.[104] showed that distributional measures such as the NGD perform better among all the semantic relatedness techniques. One of the main findings of this work is that using the NGD metric as the distance function has two main advantages: (1) it works with flat text input since it does not require any vector representation of the words; and (2)the choice of words is practically infinite since it uses the world wide web corpus to compute word frequencies.

The noisy tags are identified by means of clustering around the user assigned tags L_i. The tags which fall in any other cluster than the cluster(s) that contain the original user assigned tags are considered noise tags. Clustering using semantic relatedness of tags gives groups of tags which belong to the same topic/theme. When the user assigned tags are distributed in different clusters it signifies a lack of single topic in the tags in which case none of the candidate tags is pruned. The tags in the noisy cluster are pruned to get the final list of relevant tags from the candidate list Q_i.

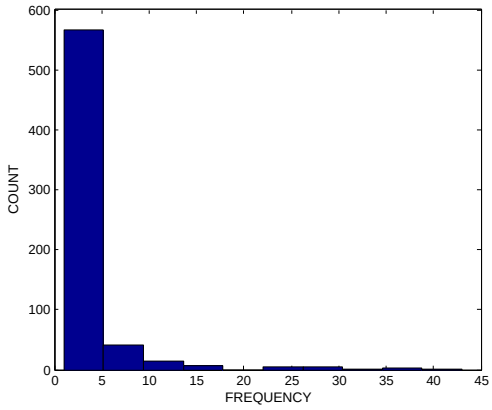


Figure 5.4: Histogram showing the distribution of tag frequency.

5.3 Experiments

In this section, we discuss the experimental setup designed to evaluate the performance of the proposed approach. The following components of the experimental design are discussed: dataset description, evaluation metrics and the baseline approaches.

5.3.1 Dataset description

As a test dataset for this work, we used tagging dataset about scientific research datasets. As mentioned earlier, the scientific datasets are one category of items whose content is not easy to interpret and there is no standard representation across datasets from various sources. The dataset for this work was collected from a data mining blog[105]. The blog consists a total of 398 research dataset instances along with user provided tags (via the del.icio.us tag subscriptions). Most of the dataset instances in this collection are related to machine learning but there are also a lot of government, finance, and search datasets as well. We extracted a sample of this data for this study. The data sample was extracted based on the following analysis.

Data preparation

The entire dataset consists of about 710 unique tags. However, several of the tags were morphological variants of words with similar semantic interpretations. The morphological variants

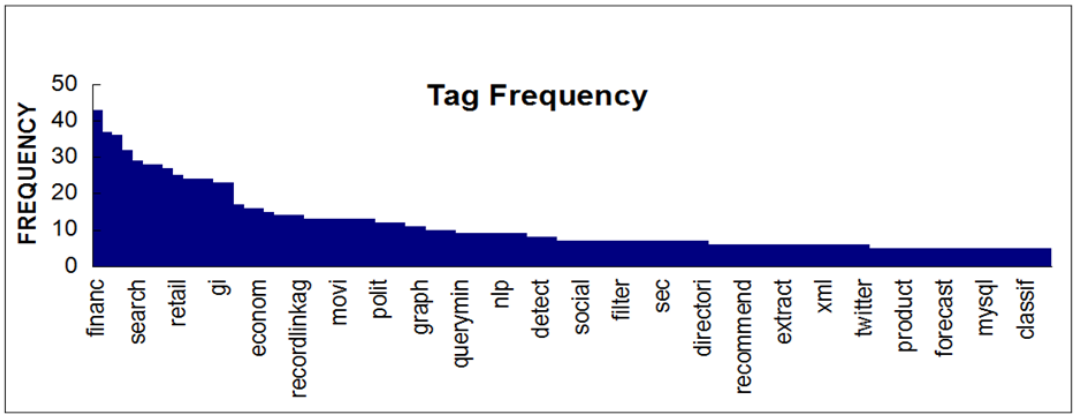


Figure 5.5: Plots showing the frequency of different tags.

were removed using the process of stemming using the nltk Porter algorithm[4] to get a reduced tag set of 657 unique tags.

The figures 5.4 and 5.5 show the distribution of the tag counts and the frequency of the tags. As shown in the figure 5.4, most of the tags were used only once. For the sample data, we used instances in the dataset with high frequency tags i.e tags which were used at least 10 times for tagging instances in the dataset. Figure 5.5 shows the frequency of tags against the tag labels. In total, there were 36 such high frequency tags. The less frequent tags were removed from further analysis. The sample data was divided into the training and the testing sets.

Training and testing sets

Each item instance in the dataset is represented by few (say n) tags. For each item instance, the testing instances is generated by using the leave-one-out technique[106]. In the leave-one-out technique for selecting the testing set, one tag from the user given list of tags L_i is randomly selected as the test tag (tag to be predicted). The remaining tags for the instance are considered as the training set. In order to remove any bias, we constructed three sets of training and testing data by performing leave-one-out sampling three times. The tags in the training set are called the user assigned tags and the tags in the test set are the expected tags or the ground truth tags for the instances in the dataset.

[4] http://nltk.org/api/nltk.stem.html

5.3.2 Baselines

As mentioned earlier, the problem of tag expansion has not been studied for items with non-interpretable (useful) content (e.g. research datasets). In absence of any prior work on tag completion for research datasets, we have used six baselines for the comparison, namely, (1) random tag generator (RandTag);(2) Non-negative Matrix factorization using Multiplicative update (NMF-mm/CF-mm); (3) Alternative non-negative least squares using projected gradients (NMF-cjlin/CF-cjlin); (4) Probabilistic Non-negative matrix factorization (NMF-prob/CF-prob); and; (5) Alternating least squares (NMF-als/CF-als) (6) Wikipedia based semantic similarity (WikiSem). These baseline approaches make tag prediction using the same input as used for the proposed approach. The state of art approaches in the tag expansion problem assumes the content availability of the item. Therefore such baselines are not applicable in the current evaluation framework.

(1)**RandTag**: The random tag generator randomly selects 10 tags from the pool(P) of the high frequency tags for each dataset instance. The tags generated by the random generator do not overlap with the user assigned tags for the dataset instances. The random baseline helps to compare the usefulness of the proposed approach in terms of the various metrics as compared to random predictions.

(2) **CF-mm**: This is the most commonly used algorithm to solve collaborative filtering(CF) problems. The details of the algorithm are available in Lee and Seung et al[107]. The NMF approach is commonly used in matrix completion problems such as collaborative filtering techniques. In order to use this algorithm, for each test case a $M \times N$ matrix was generated. M corresponds to the datasets instances such that there are M-1 tagged datasets instances in the system and one test dataset with user assigned tags. N is the total number of tags such that a dataset instance (a row) is filled with 1 in the column corresponding to its assigned tags. New tags for a test instance were predicted using the matrix completion approach. The algorithm was run with the following parameters: matrix factorization using 12 components and maximum iterations is 1000. These parameters are identical for all the other NMF based algorithms. The same of matrix construction is used for all the NMF based baselines. For each instance in the resultant output matrix, top-10 high scored columns were used as the new predicted tags for the dataset instance.

(3)**CF-cjlin**: This algorithm was proposed by Chih-Jen Lin[108] to solve the NMF problem by alternative non-negative least squares using projected gradients.

(4)**CF-prob**: We have used the algorithm provided by Lars Hansen[109]. The assumes the input matrix as a sample from a multinomial.

(5)**CF-als**: This algorithm solves the NMF problem by alternatively solving the least square equations for input matrix factors. All the negative elements are set to zero. The algorithm[110] implementation is available here[5] .

(6)**WikiSem**: Works in the past[111] have used Wikipedia as a way to compute semantic similarity between words. In order to create a meaningful baseline for comparison, we use semantic similarity finding tool, DISCO[6] with Wikipedia database (the English Wikipedia dump from 3rd April 2013). For each dataset instance d_i, new tags are assigned from the pool of high frequency tags(P) by computing the 1^{st} order semantic similarity of the tags with the user assigned tags for the dataset instance. A tag t_i in P is ranked based on its aggregated similarity score with all the user assigned tags for the dataset d_i. A tag with higher aggregated similarity score is ranked higher.

5.3.3 Evaluation metrics

In order to evaluate the performance of the proposed approach and the baseline approaches, each of the approach was used to predict new tags for the instances in the dataset. The predicted set of tags were compared with the ground truth tags. The following evaluation metrics were used to compare the effectiveness of all the approaches.

Mean Reciprocal Rank(MRR): In the information retrieval domain, the Reciprocal Rank (RR) measure calculates the reciprocal of the rank at which the first relevant document was retrieved. RR is 1 if a relevant document was retrieved at rank 1, RR is 0.5 if relevant document is retrieved at rank 2 and so on. We have averaged the RR across of the instances in the test data to calculate the Mean Reciprocal Rank (MRR). Since there is only one relevant tag in our ground truth, for quantitative evaluation, the MRR metric is suitable. In case of multiple relevant tags in the ground truth (Section 5.5), we use the Mean Average Precision (MAP) metric for comparing the approaches.

Percentage reduction: The MRR represents the efficiency of the proposed approach to retrieve relevant tags. However, it does not account for proportion of the noisy tags retrieved by the proposed approach. The percentage reduction measures the percentage of the predicted tags

[5] http://cogsys.imm.dtu.dk/toolbox/nmf/index.html
[6] http://www.linguatools.de/disco/$disco_en$.html

Table 5.1: Table showing the MRR results for tag expansion using different pruning criteria.

Pruning	Clustering Algorithm	Google DB	GScholar DB	RandTag	WikiSem	CF-mm	CF-cjlin	CF-prob	CF-als
Unpruned	unpruned	0.308	0.167	0.079	0.131	0.113	0.114	0.072	0.116
NGD based	complete	0.264	0.127	–	–	–	–	–	–
	single	0.250	0.133	–	–	–	–	–	–
	average	0.255	0.131	–	–	–	–	–	–

that were removed by the pruning algorithm.

Reduction error (RE): The percentage reduction measure computes the amount of reduction obtained by pruning. However, it does not account for the accuracy in pruning. The reduction error measures the percentage of correct tags (ground truth tags) that were pruned out. It is formally defined as follows:

$$RE = \frac{\sum_{i=1}^{N} |P_i \bigcap GT_i|}{\sum_{i=1}^{N} |GT_i|}$$

where, N is the total number of instances in the test data, P_i is the set of pruned tags for dataset instance i, and GT_i is the set of correct tags for the dataset instance i.

5.4 Results and discussion

The table 5.1 summarizes the comparison of the proposed approach with the different baselines based on the MRR metric. In the table, the different techniques for pruning noisy tags are in mentioned in column 1. The different clustering algorithms are mentioned in Column 2. Column 3 and 4 show the results of the proposed approach using Google database and Google Scholar database respectively. Column 5 to 10 show the results for the baselines. As shown in the table, the MRR values at the unpruned stage show the effectiveness of the proposed approach over the baselines. Using Google database for content generation, the average MRR value is approximately 0.308 i.e. on average bases the exact tag is predicted within top-3.33 of the predicted tag list. The MRR value is 0.167 using the Google Scholar database as source for content generation. The differences in the two results of the two databases can be attributed to the tag names being of general type than being academic terms. However, in comparison to the baseline approaches the proposed approach the MRR values are significantly higher. For the RandTag baseline, the MRR value is as low as 0.079. For the WikiSem baseline, the MRR

Table 5.2: Table showing the reduction(% error) results for tag expansion using different pruning criteria.

Pruning	Clustering Algorithm	Google DB (error)	GScholar DB (error)
NGD based	complete	$17.091\%(1.449\%)$	$14.947\%(3.260\%)$
	single	$11.955\%(1.449\%)$	$12.110\%(2.173\%)$
	average	$12.844\%(2.174\%)$	$13.571\%(2.899\%)$

value is 0.131. In comparison to the WikiSem approach, the proposed approach is performing at least 2 times better in terms of the MRR value.

Among the non-negative matrix factorization based baselines, the MRR values are distinctively lower in comparison to the results of the proposed approach. The highest MRR value for any NMF baseline is only 0.116 (using CF-als) while other NMF based baselines have lower MRR values. Although the NMF techniques are well known approaches for collaborative filtering problems such as rating predictions, such techniques are not efficient for the current problem.

One of the limitations of the NMF techniques is that they do not utilize any semantic information (such as relatedness, similarity) about the tags. On the other hand, the WikiSem approach is limited due to the fact that it does not takes into account the similarity between the items (datasets). It only uses the semantic information about the tags for prediction. However, the proposed approach takes into account both of these limitations to significantly improve the accuracy of the predictions.

After applying the pruning, we find that the MRR value decreases to approximately 0.260 (Google) and 0.130 (Google Scholar) using only the NGD based pruning. However, the pruning approach gives advantage in terms of reduction in noisy tags. Table 5.2 shows the evaluation results for the noise pruning approaches. Column 2 shows the different hierarchical clustering algorithm used. Column 3 and 4 show the % reduction and the reduction error values are shown in parentheses for Google and Google Scholar database respectively. For the Google database we find that the complete hierarchical clustering approach for pruning gives the best results among the three clustering algorithms. Comparing the Google and Google Scholar database, we find that, in general, Google database is more suitable for tag expansion. One of the reasons for this could be the generic nature of tags given to research datasets.

In summary, we find that the proposed approach outperforms the baseline approaches for the

Table 5.3: Table showing some example of tag prediction for well known dataset.

Dataset name	**User assigned tags**	**Expected tag**	**Predicted tags**
Wikipedia link	link, graph, network	wikipedia	web,api,machine learning,corpus
YouTube	web, network	**graph**	**graph**,link,api machine learning
Wikitech	trend, wikipedia	**text mining**	search, api, queries, **text mining**, text, link, google
CiteULike	record linkage, networks	graph	name, text, rdf, corpus, google, economic
Enron Email	network, text	corpus	link

tag expansion. We also find that the proposed approach for pruning noisy tags shows significant reduction in the noise within very small error range (less than 1.5%).

5.5 User study

The quantitative analysis of the results do not capture the full evaluation of the quality of the predicted tags. As an example, the table 5.3 shows the results of tag expansion for a sample instances in the dataset. The column 1 in this table is the dataset name. The column 2 shows the list of user assigned tags. Column 3 shows the ground truth tag for the dataset instance while the predicted tags (using the proposed approach) are shown in the column 4. As shown in the table, the tags predicted for Wikipedia link dataset, although do not include the exact match, yet the other predicted tags such as 'web' and 'corpus' might be relevant for the dataset. For the "YouTube" dataset, we see that the tag 'graph' is an exact match. Moreover, tags such as 'link' and 'machine learning' can also be relevant for its description. Similarly, for the 'Wikitech' dataset, the 'text mining' tag is an exact match whereas other tags like 'search','queries','text' can also be of interest. The predictions for the 'CiteULike' dataset also reflect that the quantitative evaluation for testing tag prediction is not the ideal evaluation criteria.

In this section, we further evaluate the performance of the proposed approach and the baselines based on a user study. The user assessment was performed in two folds. The first part of the assessment was to generate a new ground truth data for comparing the results of the baseline approaches with the proposed approach. In contrast to the earlier evaluation where the ground

truth was constructed from the user given tags, in this assessment multiple human subjects were provided a sample of instances (dataset names) along with the original user assigned tag list. In order to collect new tags for each instance, a list of 36 unique tags was also provided to the subjects. Each subject was asked to assign at most 5 best fit tags for the dataset instance. The dataset names were also accompanied with their URL to assist in tag selection. Due to the inherent challenges in a user study, the study was conducted for 15 instances from a total of 92 instances in the dataset. The assessment was completed by 5 graduate students[7] . The new ground truth tags for the 15 instances were obtained by aggregating all the tag preferences provided by users for each instance. For each dataset instance, the tags in ground truth were ranked based on the aggregation of rater votes given to different tags. Most preferred tag was ranked high in the ground truth. The ground truth did not contain tags present in the original user assigned tag list.

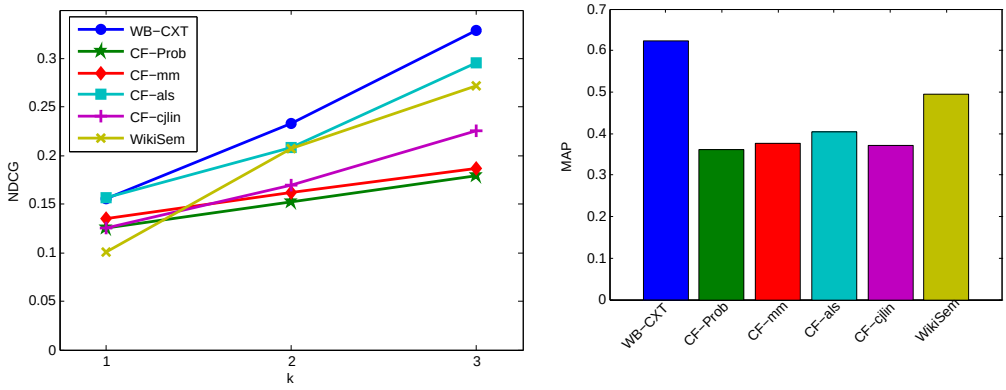
Figure 5.6(a)(b)show the evaluation results for the proposed approach and the baselines using the new ground truth data. Since the ground truth data is a ranked list of expected tags, we have used NDCG@k (k=1,2,3) and MAP metrics for comparing the performance of different approaches. These two metrics are useful for the purpose of evaluating ranked results with multiple ranked entries in ground truth and the predicted results. As shown in the figure 5.6, the proposed approach (WB-CXT) is giving the best performance with respect to the NDCG@2 and NDCG@3. The NDCG@3 for WB-CXT is 16% higher than NDCG@3 for CF-als. The NDCG@1 is comparable for WB-CXT approach and the CF-als approach (0.15). Amongst the baselines, we find that the CF-als is performing significantly better than all other CF based approaches.

In figure 5.6, we compare the different approaches using the MAP metric. As shown in this figure, the WB-CXT approach clearly outperforms all the baselines with respect to the MAP metric. The MAP metric value for the WB-CXT approach is 0.62 while the MAP value for the best baseline (WikiSem) is 0.49. The proposed approach gives a improvement of approx. 12% in MAP value in comparison to the WikiSem baseline.

The above mentioned experiment are supportive of the results in Section 5.4. We find that the performance of the proposed approach is found to be superior to the baseline approaches using two different set of ground truths.

In the next user assessment study, a new experiment to collect a subjective opinion about

[7] @CS dept Univ. of Minnesota



(a) (b)

Figure 5.6: Results for validation using ground truth tags collected using user assessment. Figure shows the comparison of the proposed approach(WB-CXT) with different baselines using (a) NDCG@k; and (b) MAP metric.

the quality of prediction results for the proposed approach was conducted. In this user study, the human subjects were provided with the same set of 15 dataset instances. The questionnaire contains a dataset name, its URL, user assigned tags and predicted tags and a user rating scale. A total of 8 graduate students[8] participated in this assessment. Each instance was rated on a Likert like scale. A rating of 5 indicates highest satisfaction while 1 indicates lowest satisfaction for the rater. In order to improve the understandability of the results, we define the concept of *good* rating as follows. A rating given by a subject is considered good only if it is 4 or above. Rest of the ratings score (1-3) do not reflect good confidence in the quality of predicted tags.

Figure 5.7 show the results of the evaluation. As shown in the figure, 7 out of the 15 instances under evaluation were given very good ratings by the subjects. In these instance the proportion of good ratings is above 60% signifying a higher agreement between the raters about the quality of the tags.

5.6 Related work

The problem of automating tagging of multimedia has been studied in two main domains: image tagging and video tagging. Tagging provides a high level representation of the actual content

[8] @ CS dept. of Univ. of Minnesota

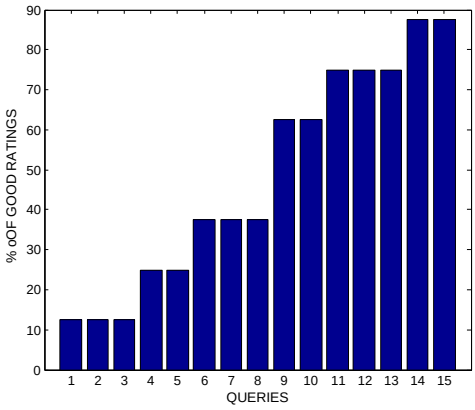


Figure 5.7: The figure shows the percentage of good ratings received by the instances in the sample data used for human assessment. A rating of 4 and above on the scale of 5 is defined as a good rating. 8 out of the 15 instances in the sample data received greater than 60% good ratings.

of the multimedia item. Classically the tagging problem is proposed as a classification problem where the low level features are connected to the high level tags by training classifiers. In the area of video tagging, Cambell et al [112] used several machine learning approaches such as Support Vector Machines, Gaussian Mixture model, Maximum Entropy learning, modified nearest neighbor classifier and Multiple Instance Learning to model concepts(tags) using various combination of low level features. As an extension of this work, Qi et al [113] proposed video annotation as a multi-label classification problem. In addition to multi-label classification, they take into account the correlation between the tags (or labels). Other works for video annotation try to improve the data quality instead of the classification model. In this category Wu et al [114] uses the side information such as surrounding text and existing tags for the social media (photos). Tensor framework was used to represent the image, audio and text content of video by Liu et al [115] and then used for classification. Further in the data representation direction of research, several feature selection algorithms are developed to leverage both sparsity and clustering properties of features and incorporated them in classification models for annotating images[116, 117].

Another direction of research which emerged due to lack of labeled data for training is semi-supervised approaches [118, 119]. For image tagging problem, approaches such as topic modeling[120], discriminative models[121], nearest neighbor based methods[122]are widely

used. However in these approaches the images in the test set can only be tagged with the labels or tags in the training data. Recently, Gilbert and Bowden [123] used web based methods to construct new labels using the internet as an additional information source. In summary, all these approaches for image tagging in general use computer vision techniques to construct features from the images and then assign label either in supervised or unsupervised manner.

The problem of automatic annotation of multimedia has also been studied as a tag/ annotation completion problem. The multimedia object, originally provided with few tags, has to be provided more tags to improve the comprehension of its description. There are two primary ways in which this problem is addressed. Initially, the approaches for completing tags for images used only text-based methods such as co-occurrence information of tags[124]. Recently, the proposed approaches have started using visual information from the images in addition to textual content[125, 126]. However, the focus in these approaches was tag expansion but such approaches might suffer from the problem of topic drift where the dominant topics of the original media are changed. Recently, a few works have addressed this problem of topic drift by using various pruning techniques for removing inconsistent tags [103, 127].

5.7 Conclusions and future work

In this chapter we propose a novel approach to tag expansion for research datasets. We have proposed a novel approach to create secondary content for research datasets using the world wide web and used the secondary content to select new tags. We experimentally evaluated the performance of the proposed approach on a real world dataset using different baselines.The performance of the proposed is compared with Wikipedia based nearest neighbor tagging (WikiSem) and non-negative matrix factorization (NMF) tag expansion approaches. Based on the Mean Reciprocal Rank (MRR) metric, the proposed approach was twice as accurate as the WikiSem baseline (0.27 vs 0.13) and at least 2.25 times the NMF baselines (0.27 vs 0.12). Based on a subjective evaluation using human subjects, we find that the quality of tag prediction for approximately 50% of the instances were given a very high rating by the subjects.

Chapter 6

A keyword based search for research datasets

Research datasets, like any other information need, have become an important part in the research life of a data scientist[128]. At present, any theory or algorithm holds higher value if it can be validated on some real world datasets [129]. Such expectations in research are realistic since technology has made data collection simpler than ever before. However, given the infinite variety of datasets over the web, a common problem that several data mining researchers/ data scientists, especially working in inter-disciplinary areas, face is to identify the most relevant dataset for their research problem. The problem of finding a research dataset is more challenging when the exact dataset of the research need is not known. This situation might also arise in career of a new researcher looking for research datasets with no other information about the dataset except for the 'context' in which the dataset has to be used. It is like a citation search where the user is trying to find the best citation for a particular topic [41].

In the above mentioned scenario, possible approaches to ascertain an appropriate dataset include reading several research papers or getting some expert recommendation. While these approaches are useful but there are number of limitations to them. Firstly, reading through all the literature surrounding a research problem is a time consuming task. Secondly, a manual search of research articles will never be exhaustive and it might lead to a narrow understanding of a dataset usage. A more efficient approach to search for datasets would be to use search engine like Google and Bing. However, these general purpose search engines use the text content of

Table 6.1: Search results for a dataset query from a general-purpose search engine

Query	**"Dataset for link prediction in evolving social networks"**
Results 1	[pdf]Probabilistic Approach to Structural Change Prediction in ...
Result 2	[pdf] Link Prediction on Evolving Data using Tensor Factorization
Result 3	[1312.6122]Shadow networks: Discovering hidden nodes with ...
Result 4	LINK PREDICTION IN SOCIAL NETWORKS - Svitlana Volkova
Result 5	A probabilistic approach to structural change prediction ...

the resources over the web to retrieve and rank them. The main problems with datasets to be retrieved successfully over the web is their inadequate text content. The raw content of datasets is generally not indexed in the databases. Moreover, the text content associated with datasets is generally short and limited to the owner's description about its properties. So a context based search for a dataset might not return interesting results (see Table 6.1). As shown in the table, the top-5 results retrieved for the given query contain only text documents and no dataset sources. This example demonstrates the limitation of general purpose search engines for context based queries. Although, a lot of research is done on query understanding [130, 131] but such context based queries with specific information need item cannot be handled appropriately by this approach. Other attempts include data repositories such as UCI machine learning repository [20], Stanford graph data [21] and few other open source repositories created and maintained by researchers. However, the search paradigm for data repositories is developed for context based querying. The alternative and efficient solution for this problem is a specialized search engine for research datasets.

For non-text content type resource like images, videos or research datasets, building a specialized search engine is non-trivial. Over and above the task of index separation, content creation is a major challenge. For multimedia like images and videos, this challenge is overcome by assigning tags (or annotations). The approaches for developing context for images range from manual to automatic tagging. These approaches include both supervised [116, 123] and semi-supervised tagging algorithms[124, 126]. Similarly for videos, the approaches include use of crowd-sourcing[1] , visual features and audio features to generate content [132]. However,

[1] http://www.youtube.com/

the task of content creation is more challenging for research datasets. Crowd-sourcing based approaches to create content for research datasets are not popular because it requires expert knowledge. Automating content creation for research datasets is even more challenging than that for multimedia items like images and videos. Firstly, unlike images and documents, with their content being represented in the standard forms (visual and text features), research datasets do not yet have a standard representation of their content. Moreover, research datasets used in different scientific disciples have variable data representation or schema. Secondly, the owners's description about the dataset is very short and restricted to raw content of the data rather than the context in which it is used. Lastly, it is hard to create extra content for datasets if it has no 'visible' features (except for its raw content).

In this chapter, we propose a novel 'context' based paradigm for searching datasets to search for research datasets. Our search systems is called *DataGopher*. We overcome the problem of content creation for research datasets using open source information sources like academic search engines for generating content. We have developed algorithmic approaches to populate content for research datasets. The content includes different types of fields. The database consists of datasets from a wide range of scientific disciplines such as sociology, geological sciences, text analysis, social media, medicines, public transportation and various other disciples.

6.1 Related work

In this section, we discuss some of the literature that highlights the innovations or directions of research in relation to search engine technologies. Over the last few decades, the science behind search engines has made several breakthroughs. The conventional search systems started with simple index of names of files over the web[133]. The approach was soon extended to search within full text files over the web and a hypertext paradigm was created in 1991[134] (later publication). However, the Archie and Gopher systems did not used natural language keywords. The real use of indexing was done later in Wandex[135] and WebCrawler[136] when the full text of web pages were indexed instead of just the title of the web pages. These were one of the earlier web-crawling based search engines. As the indexed based systems of searching the web gained popularity, several search engines were developed including Magellan, Alta Vista, Yahoo, Google within a short span of a decade [135].

With the natural language paradigm of searching the web, real world challenges were identified in conventional search systems. There were number challenges which appeared with the use of natural language querying. While using natural language for querying is convenient for users, it increases the rate of making spelling errors in the queries causing search failures. This problem was overcome by employing thesaurus based spell checkers parsing the queries. Features such as spelling suggestion were incorporated in search systems[137]. In addition to the problem of misspellings, another major problem in keyword based natural language querying is the expression of the information need. It was found that the search system has no way to understand the user's information need in case the query does not contains the indexed terms[131]. This problem might be due to different morphological variants being used in querying while only a particular form of the word is indexed in the database. In order to alleviate this problem, techniques such as stemming[138] and N-gram analyzers[139] were developed. Although stemming and N-gram based indexing and query parsing helped to remove the problem of morphological differences, these approaches did not work when the query terms were ontological variants of the index terms. For example, for a query 'winter clothes', search results like 'woolen jacket' and 'scarf' are also relevant but the search systems had no way to identify these relations. This area of problems led to development of creating ontological dictionaries for indexing the database as well for parsing the queries[140].

The database search was no longer limited to document search. As the web became a store house of all sorts of multimedia resources, indexing multimedia items became a challenge[141]. Unlike documents and web pages, multimedia items had the challenge of content representation. Since a search systems is a popularly a text (as keywords) based search, the challenge was to create text content for multimedia items. Being general purpose resources, efforts were undertaken to annotate multimedia items with tags and descriptions about their content. Recently, the research in multimedia annotation has been moving towards automation of multimedia resources using feature based supervised and unsupervised techniques. The approaches for developing context for images range from manual to automatic tagging. These approaches include both supervised [116, 123] and semi-supervised tagging algorithms[124, 126]. Similarly for videos, the approaches include use of crowd-sourcing[2] , visual features and audio features to generate content [132].

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

Some of the industrial effort to solve this problem are Quandl[3] . It is a collaboratively curated portal to millions of financial and economic time series dataset from over 250 sources. While Quandl is a good source for time series dataset but it does not serves the purpose of search engine for public datasets based on context. Moreover, search systems such as this are based on massive data curation efforts.

6.2 Problem Formulation

The problem of searching dataset can be formulated as an interest to item relevance matching problem. The interest of the user is expressed in form of some keywords to describe his/her information need. In the information retrieval domain, the input keywords are often referred to as query. For the rest of the paper, we will refer to the user given keywords as query(Q). The search task is to retrieve the k-most relevant items for the given query(Q) from a database of items. Although, the parameter k is not fixed in information retrieval systems, however, from a practical standpoint only the top-10 results are most determinant towards the performance of any search system.

As mentioned earlier, the dataset search on the proposed search engine is more effective for context based queries. A query for dataset can be done to excavate (1) the source of the dataset using the dataset name or (2) dataset satisfying some numerical criteria about the dataset (e.g. size, number of features) or (3) dataset to best fit the context of its use (research problem). In this chapter, we address the dataset search for type (3) because the general purposed search engine cannot effectively address these kinds of searches.

6.3 Proposed Framework

Before going into the details of the various components in the system's design, we provide an overview of the system. As shown in the figure 6.1, the search engine basically consists of two components, namely, database creation and relevance search. For the database creation module, we describe some of the techniques used to automate the task of database creation. While dealing with special items like research dataset, the database creation is not trivial due to lack of the text content for research datasets. We present novel techniques for creating extra content

[3] http://www.quandl.com/

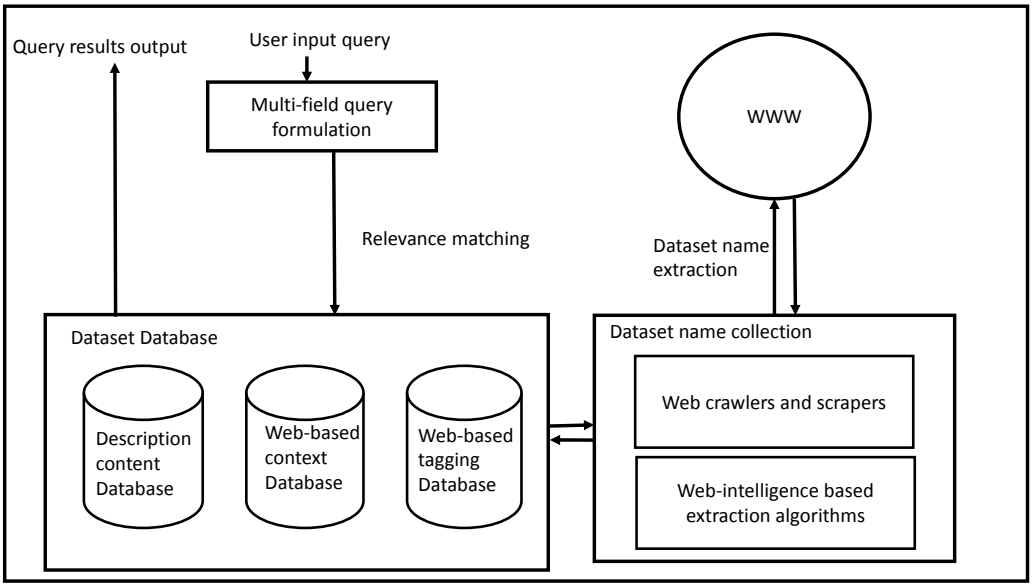


Figure 6.1: A schematic of the proposed search engine model.

or 'context' using open sources for the purpose of automation and adding useful information about dataset in the database. In the relevance matching module, we describe the relevance matching algorithm used to obtain the most relevant results for a user query(Q). The details of each module are mentioned in the following subsections.

6.3.1 Database creation

While general purpose search items such as digital documents, web pages and blogs are indexable on their content represented as text, multimedia search items like images and videos require extra text information (or context) to enable indexing on these items. In the multimedia information retrieval literature, several techniques have been developed to reinforce the task of context creation based on the non-textual content (visual and audio features). However, the task of creating database for searching datasets is still a big challenge. As, mentioned earlier, the two main reasons for the absence of any systematic approach for indexing research datasets is the lack of availability of text content for the dataset. The text content available about a research dataset is not more than the dataset description provided by the dataset owner. Owing to the shortage of text content for identifying a dataset over the web corpus, general purpose search engines are

not effective in searching datasets.

In this section, we discuss the proposed approach to construct a new database for research datasets. The database creation is accomplished in the following two steps:

Dataset name collection

In order to represent the resource in the database, an identifier is essential. In general practices, the identifier is a title or the name of the resource indexed in the database. In our database, the research datasets are represented by their referred names i.e the names by which they are referred in research articles. The names were collected in two ways:

1. Automated extraction: Leveraging upon the description of datasets in research articles, we developed an automated approach to extract dataset names from research articles[142, 129]. The dataset names are identified using natural language processing techniques and co-occurrence information from the web for identifying dataset names.

2. Web scraping: Several of the dataset names were extracted from the web via automated crawlers and scrapers. Most of the dataset names were collected from various data repositories open sourced over the web. The database consists of datasets user in diverse research areas like climate data, sensor data, medical data, finance data and other research areas. Although it was our best attempt to increase diversity in the dataset collection for the database, but we also accept that it is not practical to be able to collect all dataset resources available in the real world.

Other meta-information about the datasets such as their URL and description were also scraped from the web.

Hidden content ('context') creation

As mentioned earlier, the only content information available about a research dataset in the text form is the owner's description about the dataset. Since search engines, in general, use only the textual content of any resource to index them, the lack of relevant content in the owner's description might not help in creating good index for the dataset. Indexing a dataset with terms which are not used by users while searching for a dataset will result in inaccurate hits for the query. While the resource is available in the web corpus, a lack of content information about the

dataset in text form leads to poor retrieval results. We overcome this problem of lack of relevant content for a dataset by creating a hidden content or "context" for it. The "context" adds more information about the dataset in the database. Although the context might not be user readable (therefore "hidden") but it is used to index the dataset in the database.

In order to create useful "context", the text information contained in the "context" should include terms which are used by users while searching for a dataset. In the scientific research, a user may search for a dataset (1) looking for dataset that suits his research topic or (2) dataset based on some specification about its features or (3) finding datasets similar to what they have. In this work, we address the first approach for searching for a dataset by adding use-case information about the dataset as its "context".

One of the approaches to accomplish use-case based "context" is to have an expert based annotation of datasets. However, owing to the huge volume of the datasets beings used in scientific research, a manual expert based annotation is a infeasible task. As an efficient alternative to this approach, we propose a novel automated approach for creation "context" for datasets. The proposed approach leverages information from academic search engines to create the "context" for a dataset. Since the database used by academic search engines consists or research articles, scholarly reports and other scientific documents, these resources can be used to create the "context" for a dataset based on the assumption that the datasets are referred by a certain name in some research documents. Using the search engine capacity to retrieve documents from the database based matching the input query, we use the dataset names collected in the last step to query the academic database to retrieve documents which refer to the dataset within its content. The "context" for the dataset is constructed form the retrieved results in the following manner.

We use the MNCAT library search engine[4] as the academic search engine for the purpose of "context" creation. Given a dataset name d_i as the input query, the search engine returns documents titles ranked in order of their relevance to the input query. In addition to the titles of the documents, the library search engine returns the various subject categories (extracted from the search results). These subject categories corresponds to the various research topics and keywords for the retrieved results. Thus, the "context" for a dataset contains two components, namely, (1) title text and (2) the subject tags.

The title text context is derived from the top-50 titles of the retrieved documents by parsing using natural language parsers. The NLP parsers remove stop words, special characters and

[4] University of Minnesota- MNCAT

converts the text into lowercase. The text context, therefore, consists of words extracted from the titles of the retrieved documents. All the titles in the top-50 results are given equal weights. Weighing the titles in order of the rank in which they appear will give weights to the different words in the title context and might be helpful to create a better context. However, for this work we have ignored putting using different weights for the titles.

The subject tags are a prerogative of the MNCAT library search engines. We use this feature to create subject tags for research datasets. The subjects are assigned as a list of tags for the research dataset. The subjects tags are not mixed with title text context of the research dataset, since both the title context and subjects tags are used to create separate indexes for the research dataset (as described in the following section).

Database indexing

The database indexing is carried out using the Whoosh library[5] . Each dataset instance is indexed in the database using its various content types, namely, (1) the owner's description(extracted fro the web) (2) title-based context (hidden context) (2) subject keywords (tags). We also index the name of the dataset but it is not expected to contribute significantly to the index due to its minimal text content. The three content types used in indexing are referred as different fields of the dataset. Each of the field is indexed separately. For example, there is separate index for the author's description. Then there is separate field for the title based "context" and a separate index for the subject tags. While the owner's description and the title-based "context" are indexed as text fields, the subjects tags are indexed as full keywords. In order to create indexes to include different morphological variants of the words, we have used the N-gram parser to create N-grams for each word. The text content in each of the fields is tokenized using the space tokenizer. Each token is then expanded in the different N-grams. The minimum window size and maximum window size used for N-gram generation are 3 and 20 respectively.

In the next section, we discuss how the indexed database is searched for a input query.

6.3.2 Relevance matching

Given a user query(Q) for a dataset, in this section, we discuss the technique used in the proposed framework to search for relevant datasets. The input query Q is first tokenized using the

[5] https://pythonhosted.org/Whoosh/

space delimiter. Each token in the query is used to search datasets in the database. The tokens in the query are OR grouped with a higher precedence to number of token match than to the frequency of a single token. As an example, the query Q can be 'foo bar' and there are two matches- one containing only the token 'foo' 5 times and the second result contain both the tokens 'foo' and 'bar'. It is expected that the result containing both 'foo' and 'bar' should get a higher score than the one with high frequency of 'foo'. Since, we have the datasets indexed by multiple fields, each of the tokens in the query are also searched in the indexes of all the fields. The multi-fields in the database are utilized by ORing the token for each field. An example is shown below.

For a query(Q), "climate prediction data", the query will be parsed in the following manner:

$$
\begin{aligned}
&Or([Or([Term('title', u'climate'), Term('des', u'climate'),\\
&Term('cxt', u'climate'), Term('tags', u'climate')]),\\
&Or([Term('title', u'prediction'), Term('des', u'prediction'),\\
&Term('cxt', u'prediction'), Term('tags', u'prediction')]),\\
&Or([Term('title', u'data'), Term('des', u'data'),\\
&Term('cxt', u'data'), Term('tags', u'data')])])
\end{aligned}
$$

where 'title','des','cxt' and 'tags' are the dataset name, dataset description, title-based "context" and subject tag fields respectively in the database.

For each token, the BM25 algorithm [143] computes a relevance score. Since each of the token are grouped by OR and the search over each of the fields is also grouped by OR, the OR grouping is converted to a mathematical addition of the BM25 scores for search results for each token.

The final relevance score(RS) for a results is computed as a weighted sum of relevance scores for each term in the indices of the result. The search function is a weighted function of the search in the (1) text description (2) hidden context and (3) tags for the items in the database. The optimal weight for each of the fields is determined experimentally (discussion in Section 6.4). The results are ranked in decreasing order of the relevance score.

6.4 Experiments

In this section, we discuss the quantitative evaluation of the keyword based search engine. There are two parts of this evaluation. In the first part, we compare the performance improvement (via precision@k and NDCG@k) when different parts of the context are added to the local text description of the research datasets in the database. This comparison provide evidence about the impact of adding the web context to the database objects. The impact is demonstrated by improvement in relevance search. In the second experiment, we compare the performance of the proposed search engine with a popular general purpose search engine based on a user study.

Quantitative evaluation

In this section, we discuss the experimental setup for evaluating the effectiveness of adding the web context to the owner's description about the datasets. As explained earlier, the owner's text description alone is not adequate for their retrieval for a context based user query. In order to overcome this problem, we proposed an approach to add extra text content from web resources. In this section, we discuss our experiments with 27 user given queries. For each query, relevant results in the retrieved results is determined using expert opinion. We then evaluate the performance of the proposed context based retrieval with the baseline. We discuss the details of the experiments below.

Data preparation The input data consists of 27 natural language queries (in English) from various graduate students in the CS department at University of Minnesota. The targeted students were asked to express their information need for research datasets based on their field of specialization. The data queries belong to categories of biomedical, data mining, social networks and health care. All these research areas extensively require data for the purpose of analysis and testing. The distribution of query lengths is shown in figure 6.2. The maximum query length is 10, minimum is 3 and the median query length is 5.

Survey design In order to collect response for relevant results from the experts in the categories mentioned above, we used an on-line survey tool to create forms with queries along with a set of 10 candidate for relevant results. Each query form consisted of the input query on the top. A set of 10 candidate relevant results were shown below the query. The results were

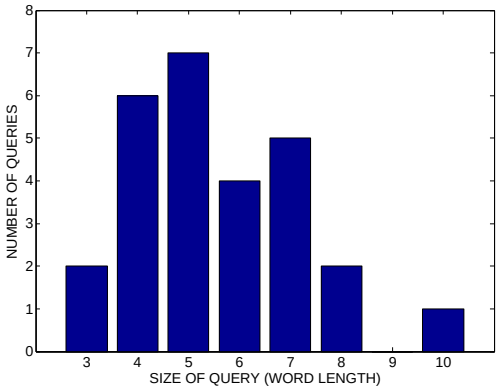


Figure 6.2: Histogram showing distribution of query lengths.

unordered (unranked). The ground truth for relevant results for each query were collected from experts in the following manner.

Ground truth The survey links to these forms were emailed to the respective research domain experts in the department of Computer Science and Engineering at University of Minnesota, Twin Cities. The domain experts were provided with instructions to label the search results as relevant or not relevant for a given user query. The relevance of a search result was determined by consensus of 3 or more experts marking a particular search as relevant. A search result is considered relevant only when 3 or more experts simultaneously mark it as relevant. The search results consisted of dataset name title, a URL link for its web source and a small snippet to provide a quick understanding about the dataset. This approach for collecting ground truth for IR systems is fairly conventional in the IR community. This ground truth is used to compare the performance of the proposed approach with baselines (as described in the following sections).

Evaluation metrics We have used two evaluation metrics to compare the performances.

1. Precision@k:The value of k used for experiments is 5.

2. Normalized discounted cumulative gain (NDCG)@k : The value of k used for experiments is 5.

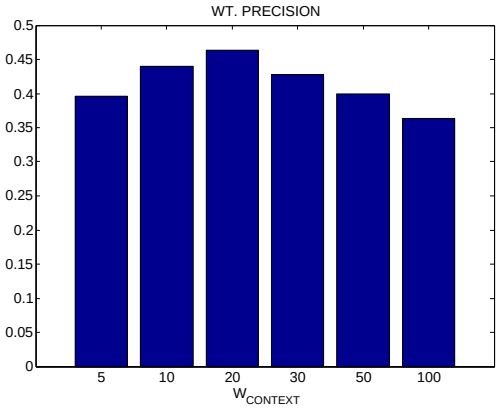


Figure 6.3: Variation of precision@5 with the variation of weight of hidden context field in the search function.

We assume that top-5 retrieved results are more significant while searching for research datasets. Unlike document search where top-10 are considered, research datasets search requires that highly relevant datasets appear within the top-few so that the user need not speculate when provided with more choices. Hence, we take a lower cut-off of top-5.

6.5 Results and discussion

As described earlier, in the proposed approach for enhancing retrieval of research datasets, we add web based context as unstructured text and keywords (or tags) to the original text description of the datasets. In this section, we discuss the results of the experiment. One of the aim of the experiment is to determine the impacts for adding the web based context. Based on the ground truth (described in Section 6.4), we quantitatively measure the impact of adding the two types of web context (discussed earlier) with variable weights.

Given that there are two additional web context that can be added to the baseline content (owner's description) of a research dataset, we study the impact of adding (1) unstructured (hidden) web context by varying its weight in the search function (2) keywords (tags) by varying their weight in the search function and (3) combination of hidden context and keywords by varying each of their weights in the search function.

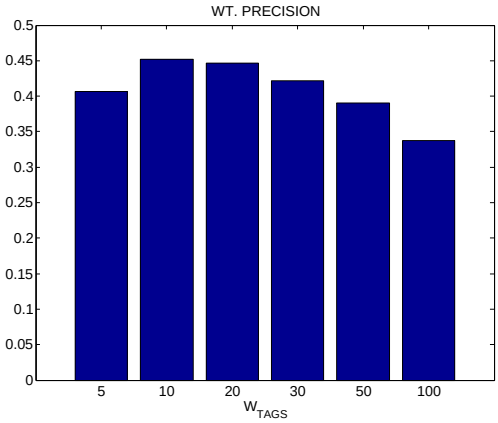


Figure 6.4: Variation of precision@5 by varying the weight of the tag field in the search function.

Figure 6.3 shows the results of varying the weight of the hidden context in the search function. As shown in the figure, the weight for the hidden context are varied from 5 to 100. The highest value of weighted precision, however, is attained at a weight of 20. In this search framework, the baseline (only owner's text description) was given a weight of 20 in the search function. A weight of 20 to the hidden context, therefore, corresponds to an equal weight. So, if only the hidden context is appended to the baseline text, the best search performance is obtained, using the proposed model, when both the hidden context and the baseline content are weighted equally.

Figure 6.4 shows the results of varying the weight of keywords (tags) in the search function. Similar to the hidden context weighing, the weights are varied from 5 to 100. In a standalone mode, we find that highest precision@5 is with a weight of 10 to the keywords. However, this weight is relative to the weight of the baseline text. Since the baseline text is weighted at 20 in the search function, the current analysis of weighted precision signify that the precision is maximized when the weight of the keywords is half the baseline text.

Figure 6.5 shows the variation of NDCG@5 for the various values of weights for the hidden context and the keywords (tags) field in the search function. Similar to averaged precision@5 results, we find that averaged NDCG@5 (across the 27 queries) is showing a similar trend. The highest value of NDCG@5 appears at $W_{CONTEXT}$=20 and W_{TAGS}=10 for the hidden context and the tags field respectively. Each of those were measures independent of each other or to

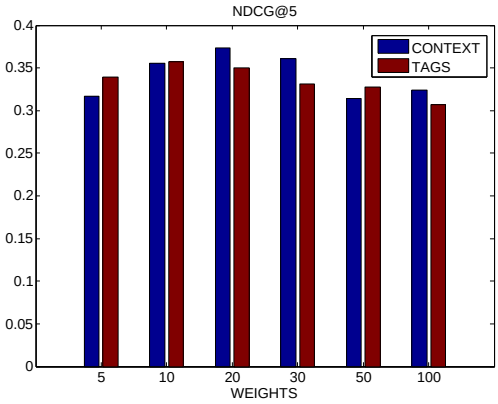


Figure 6.5: Variation of NDCG@5 by varying the weight of the hidden context (blue) and the tag (red) fields in the search function.

say in other words, the search performance was first tested by adding only hidden context to the baseline text to obtain the NDCG@5. In the second case, we only used the tag field in addition to the baseline text to compute the NDCG@5.

Based on the above analysis, it is important to determine what weights should be used when both the hidden context and the keywords are simultaneously appended to the baseline text. In the figure 6.6, we show the variation in precision@5 when both the weights for hidden context and keywords is varied simultaneously. The x-axis in the figure denotes the variation in the weight of the hidden context and the y-axis denotes the variation in the weight of the keywords in the search function. Both are varied from 10 to 100 at an interval of 10. For all the combination of their weights, we determine the values of the precision@5. As shown in the figure, higher values of weighted precision occur in the range $W_{CONTEXT} \leq 30$ and $W_{TAGS} \leq 40$. This region is redder (high precision@5) in comparison to the rest of the heat map. This heat map shows that the search performance is significantly boosted when appropriate weights are assigned to hidden context and the keywords in the search function. Assigning higher weights to web context does not necessarily improves the performance. In order to determine the exact combination of weights of hidden context and the keywords, we further zoom in the range specified above.

In figures 6.8 and 6.9 we zoom in the high precision@5 and high NDCG@5 region shown in figures 6.6 and 6.7. Here the weights of hidden context and tags are incremented by 5. As shown

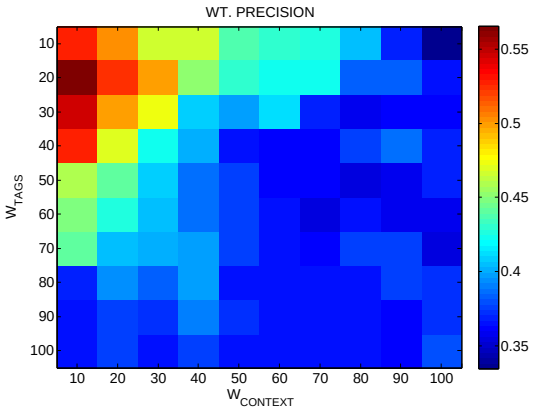


Figure 6.6: Grid search for finding best combination of the weights of hidden context and keywords in the search functions. Averaged precision@5 is used to determine the best combination.

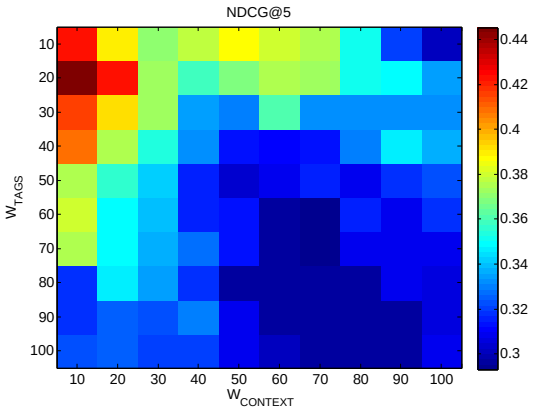


Figure 6.7: Grid search for finding best combination of the weights of hidden context and keywords in the search functions. Averaged NDCG@5 is used to determine the best combination.

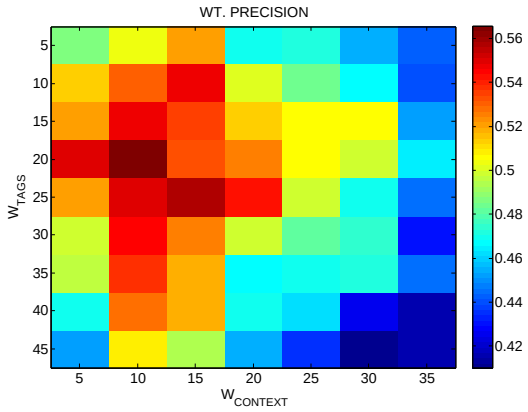


Figure 6.8: Zoom into the heat map of averaged precision@5. The scale of analysis is finer than the previous heat map.

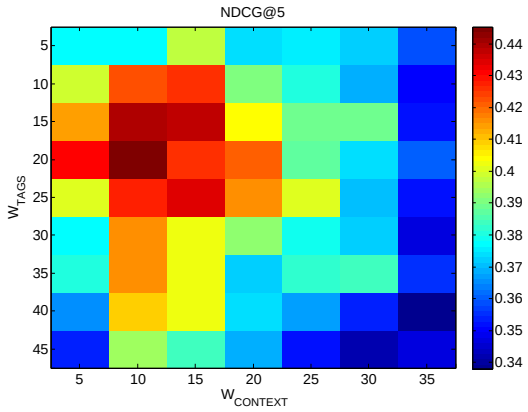


Figure 6.9: Zoom into the heat map of averaged NDCG@5. The scale of analysis is finer than the previous heat map.

Table 6.2: Comparison the performance of search engine with different content of the items in the database.

Metric	DESCP	DESCP+CXT (20)	DESCP+TAGS (10)	DESCP+CXT (10)+TAGS (20)
Precision@5	0.261	0.464	0.452	0.565
NDCG@5	0.228	0.374	0.358	0.445

in the figures, the highest values of both the avg. precision@5 and avg. NDCG@5 occurs at $W_{CONTEXT}$ = 10 and W_{TAGS}=20. Using this analysis, we determine the optimal combination in which the weights of hidden context and tags should be combined for maximizing search performance (measured via weighted precision). It should be noted that these values of weights are relative to the fixed weight of the baseline text. In all the above analysis, the weight for the baseline text was fixed to be 20. Therefore the above analysis is useful in determining the relative proportion of weighs that be assigned to the web context. The web context was appended to the baseline text and not used in a standalone way.

Finally, we compare the performance of the proposed approach with optimal weights for various web context added to the baseline text in table 6.2. As shown in this table, the avg precision@5 and avg NDCG@5 is the highest for the proposed approach when the baseline text (DESCP) with weight 20 is appended with both the hidden context (CXT) and tags (TAGS) using weights of 10 and 20 respectively. In comparison to the baseline approach using only the owner's description (DESCP), the search engine performance measured via precision@5 and NDCG@5 is boosted by almost twice their respective values for the baseline. It is clear from the table that adding additional text content from the web either in form of hidden context or tags or both is significantly helpful in improving retrieval performances over context based queries. However, the results are more significant when both the hidden context and the tags are appended with proper weight proportions to the baseline text. In the next section, we will discuss the performance evaluation of the search engine designed using the parameters derived from above analysis via a user study.

6.6 User study

While the above described experiments helps to understand the quantitative differences between the baseline and the proposed approach, it is even more important to determine the practical utility of the proposed context based search engine over the widely popular general purpose

search engine. In this section, we study the practical utility of the proposed search engine via a user study to compare the performance of the proposed search engine with Bing search engine.

This section is organized in the following manner. First, we discuss the experimental design which provides details about the query formulation, the participants in the evaluation and the details about data collection. Following this, the data analysis from the user study comprises of significance test using two statistical significance test used for evaluating the rankings of two retrieval systems. These experiments for statistical validation as also known as matched pair experiments since both the search engines are compared over same set of input queries [144]. Finally, we discuss some inferences based on the results of these experiments.

6.6.1 Experimental design

Query formulation

For the user study, the input queries for the search engine were collected from three sources:

1. User provided: A set of 9 context based queries were given by users. Here the user base consists of PhD students specializing in the area of data mining. Using an email broadcast of a survey to collect queries for research datasets, several PhD students in the area of data mining were contacted. Thus these queries denote the actual information need of researchers about research datasets.

2. From Internet post/forums: A second set of 4 queries was collected by searching the web for information need about research datasets. The queries were obtained from the forum post at Quora[6] dataset community. In this community, the researchers express various information need about research datasets. For our purpose, we selected queries which were context based (where context of use was provided but not the dataset name).

3. Self constructed: The final set consists of 4 queries which we generated based on our information need for research datasets.

All the queries were context based, where the context of the dataset is know but not its name or source.

[6] www.quora.com/

User study design

Based on the information needs or context based queries (17 in total), we used the queries to retrieve results from our search engine as well as the baseline search engine- Bing.

Bing: The baseline search engine selected for this experiment is the Bing[7] search engine. There are several reasons for the choice of this search engine. Firstly, the best possible comparison for the proposed search engine model is a general purpose search engine which allows natural language querying. Secondly, a general purpose search engine is the most popular choice for searching dataset (based on our experience). While data repositories exists but they are mostly used as dataset look up table and not as search systems for excavating the datasets as per research need. Thirdly, we use one of the most popular search engines in the present context which ensures a lot of technical advances in terms of the state of art research in search engines. Finally, Bing.com provides the API for Bing search in the most convenient form both in terms of price and usability. This flexibility was not available with other search engines like Google. Moreover, the initial system was designed to dynamically obtain queries from users where users can interact with the system and see the results from both the search engines. However, due to technical difficulties the scheme had to be dropped. However, the on-line querying systems was used to collect data for user queries (as discussed earlier).

Each of the 17 queries were ran through both the systems and the retrieval results were displayed as shown in figure 6.10. The users were given the choice to select the queries they want to evaluate. The categorization of the 17 queries was done manually into 7 categories: (i) Text mining/ Information retrieval (2) (ii) Signal Processing (1) (iii) Recommendation systems (1) (iv) Social networks (4) (v) Machine learning (2) (vi) Times series (3) (vii) Link prediction (4). The evaluation also consisted of instructions explaining a context based search and a few samples of relevant and irrelevant results.

User study participants

The user study was done on-line where the advertisement for the study was posted at various forums and communities of researchers. The advertisement was as well emailed to several graduate students at University of Minnesota. We used registration forms to collect information about the participants. Based on the registration information, the participant pool comprised of

[7] http://www.bing.com/

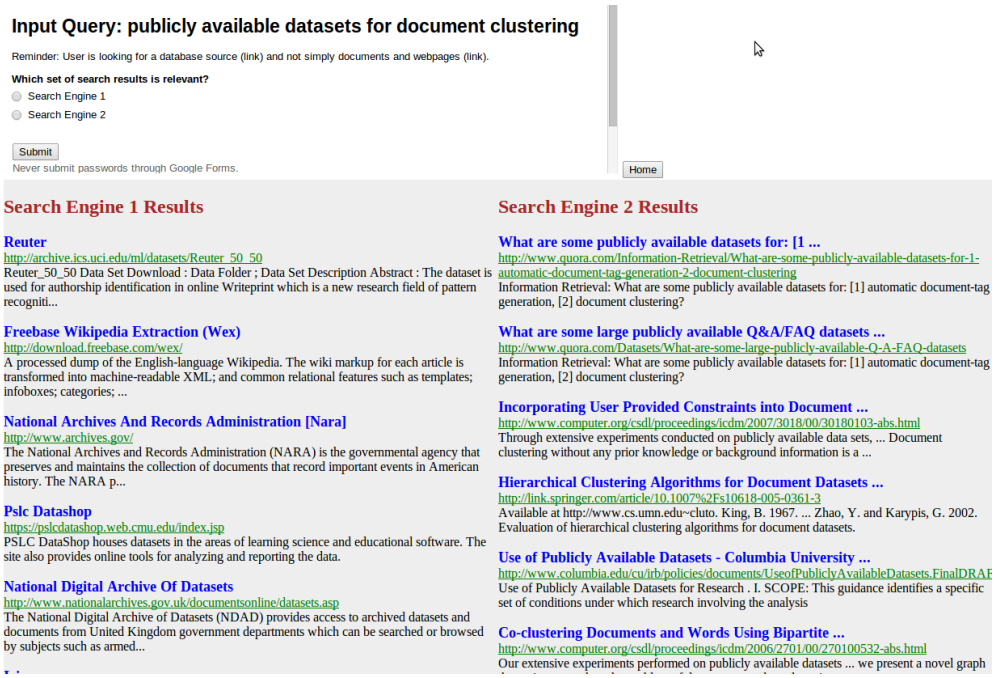


Figure 6.10: A snapshot of the evaluation form for the user study.

3 Professionals, 12 PhD students and 5 MS students. As shown in figure 6.10, the participants were supposed to select either search engine 1 or search engine 2 as there judgment about the relevance of the search results. The identity of the two search engine as well as the origin of the queries were not disclosed to the participants. Moreover, in order to remove duplicate answers from the same user, we encoded the threshold time distance of 10 seconds between each feedback. Multiple responses within a 10 seconds time window were dropped off from the count.

6.6.2 Results and discussion

The purpose of user study is to validate our hypothesis that the proposed search engine outperforms the baseline (general purpose search engine) for context based queries. In the IR literature, this is hypothesis testing is done using statistical significance tests. Since, we are comparing the rankings of two search engine for the same set of queries, these experiments are known as matched pair experiments. The data used for these experiments is the evaluation feedback collected in the user study.

For statistical significance testing, we use the Student t-test and the Wilcoxon signed-rank test. The experimental setup for the hypothesis testing is as follows:

H_o: Search engine 1 and search engine 2 are not different

H_α: Search engine 1 is better than search engine 2

Since, we are interested in testing if search engine 1 is better than search engine 2 and not just search engine 1 is different from search engine 2, we use the one-sided or one-tailed test.

Since the input queries for the user study were collected from three different sources, we conduct the statistical test for each category separately as well as for the entire set of 17 queries. The details of each test is described in the table 6.3 below. For these test we will use α=0.05,0.01 as the significance levels. Certainly, α=0.01 is a much stronger indicator of statistical significance. The p-values are compared with α to reject or fail to reject the null hypothesis.

For the one sided t-test, the p-value for the user provided queries is as low as 2.132 e^{-05} i.e. p-value < 0.05 and p-value < 0.01. Based on this observation, we reject the H_o in favor of H_α. Similarly, we find statistically significant results for all the query sets, since the p-value < 0.01 we reject the null hypothesis in favor of the alternative hypothesis i.e. Search engine 1 (proposed) is better than search engine 2 (Bing) for context based queries. However, there are

Table 6.3: Table showing the p-values for the Student t-test and the Wilcoxon signed-rank test. * the tests were done using R-software packages.

	One-sided t-test*			One side Wilcoxon signed-rank test*	
	t	df	P-value	V	P-value
User Provided (9)	8.0264	8	2.132e-05	45	0.00445
From Internet posts (4)	5.4772	3	0.00598	10	0.04876
Self constructed (4)	5.1962	3	0.00692	10	0.0625
Total queries(17)	11.4876	16	1.931e-09	153	0.000151

some issues regarding use of t-test alone due to the following reasons. Firstly, this test assumes that the t-statistics comes from a normal distribution. This may not be valid for small sample size as is the case with ours. The second problem is that of level of measurement associated with the effectiveness measure. In general t-test assume the measurements to be ordinal but then the difference between them should also be considered differently (e.g. a difference between 80 and 70 is not same as the difference between 20 and 10).

To minimize the influences of the above mentioned assumptions, the non-parametric Wilcoxon signed test is used since it makes less assumptions about the effectiveness measures. From the table, we see that for the user provided queries, the p-value <0.01 which means that there is strong statistical evidence in the data to reject the null hypothesis in favor of the alternate hypothesis. For the internet post queries, we find that there is only mild evidence in the data to reject the null hypothesis in favor of the alternate hypothesis, since the p-value $(0.0487) < 0.05$ but p-value $(0.0487) > 0.01$. Whereas, for the self constructed queries, we do not find enough evidence to reject the null hypothesis, since the p-value $(0.0625) > 0.05$. For the overall dataset, there is strong evidence in the data to reject the null hypothesis since the p-value is $<< 0.01$. Note that we have used α=0.01 as the indicator of strong evidence because the sample size is small.

6.7 Conclusions

In this chapter, we proposed a novel 'context' based search paradigm for research datasets. Research datasets, in general, lack representative text content. In this work, we addressed this problem by developing an algorithmic approach to create additional content or 'context' using the open source information from academic search engines in various forms. We also used web-crawling and web-intelligence based algorithms to prepare the dataset name collection. Using the proposed approach, the database for research datasets was populated with new content in multiple fields. The relevance matching module takes into consideration the content available in various content fields in order to rank the datasets for a given query. The performance of the search engine is evaluated by comparing the impact of adding text content from web resources to the baseline text provided as a description by owners of the datasets. We find that the proposed approach to append the baseline text with additional text content from the web resources significantly improves the performance of the search. In terms of precision@5 and NDCG@5, the proposed approach is atleast twice as better than the baseline approach. We also studied the performance comparison with a state of art general purpose search engine via user study. Using two statistical significance tests, we are able to infer the statistically significant superiority of the proposed search engine over the general purpose search engine for context based queries.

Chapter 7

Conclusions and discussion

In this research, we have made significant contributions to advance the state of art in IR for low text content items.

The proposed approaches allow retrieval of low content items by generating additional text content from several web resources like academic search engines, crowd sourced knowledge from Wikipedia. Low text contents items such as research datasets and short text research documents can be effectively retrieved using the automatically generated text content in form of tags and keywords. Moreover, the current approach to generate additional content helps to retrieve such items with context based queries rather than description queries. The proposed framework of retreival is particularly useful when both the name and source of the item is unknown. We also demonstrated the advantage of using such a retrieval system with a real world user study.

Aside from the particular technical solutions for the low text content challenge in IR, this project has the potential of bringing a context based retrieval paradigm for short text items.The conducted research can be extended to create specialized retrieval systems for items such as news feeds, tweets, product reviews as well in short text document categorization and tagging.

This project also leads several open dimensions for future researchers to work upon and develop them into practical solutions. Some of them are enumerated below:

1. Interested researchers can extend the context based search model for research datasets to feature based search model by annotating the raw numerical content with text information. This would require a basic schema to represent the raw numerical content. Thus

various schemas or classes of schemas can be tested to represent the raw content for various research datasets.

2. The proposed approach can be taken beyond scientific research datasets and be applied to any dataset, for example, in an organization, university or research laboratories.

3. The approaches developed for annotating short text documents can be extended to various short text items such as Twitter and Facebook posts, online product review, news feeds etc. Leveraging web resources would be specially helpful to annotate such items.

Honors and Publications

- Awarded Student travel grant for 15^{th} IEEE International Conference on Information Reuse and Intergration, 2014.
- Awarded Student travel grant for IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2013

International Journal Publications

- Praveen Kumar, **Ayush Singhal**, Sanyam Mehta, Ankush Mittal, *Real-time moving object detection algorithm on high-resolution videos using GPUs*, Springer Journal of Real-Time Image Processing (2013): 1-17.
- **Ayush Singhal**, Jaideep Srivastava, *Data Extract, Mining Context from the Web for Dataset Extraction*, International Journal of Machine Learning and Computing Vol 3(2), pages 219-223.

International Conference Publications

- **Ayush Singhal**, Jaideep Srivastava, *Leveraging the Web for Automating Tag Expansion for Low-Content Items*, In 15^{th} IEEE International Conference on Information Reuse and Intergration, 2014. [**accepted**]
- **Ayush Singhal**, Ravindra Kasturi, Jaideep Srivastava, *DataGopher: A Context based search engine for research datasets*, In 3^{rd} IEEE Workshop on Data Integration and Mining (IEEE IRI-DIM 2014). [**accepted**]
- **Ayush Singhal**, Atanu Roy, Jaideep Srivastava, *Understanding Co-evolution in Large Multi-relational Social Networks*, In 3^{rd} IEEE Workshop on Data Integration and Mining (IEEE IRI-DIM 2014). [**accepted**]
- **Ayush Singhal**, Jaideep Srivastava, *Semantic Tagging for Documents using 'Short Text' Information*, In Proceedings of the 5^{th} International Conference on Web Engineering and Services (InWes 14), p. 337-350, CS & IT-CSCP, 2014.

- **Ayush Singhal**, Jaideep Srivastava, *Generating Automatic Semantic Annotations for Research Datasets*, In Proceedings of the 4^{th} International Conference on Web Intelligence, Mining and Semantics (WIMS 14), p. 1-11. ACM, 2014.

- **Ayush Singhal**, Ravindra Kasturi, Vidyashankar Sivakumar, Jaideep Srivastava, *Leveraging Web Intelligence for Finding Interesting Research Datasets*, In IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2013, vol. 1, pp. 321-328. IEEE, 2013.

- **Ayush Singhal**, Ravindra Kasturi, Jaideep Srivastava, *Automating Document Annotation Using Open Source Knowledge*, In IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2013 , vol. 1, pp. 199-204. IEEE, 2013.

- **Ayush Singhal**, Karthik Subbian, Jaideep Srivastava, Tamara G. Kolda, Ali Pinar, *Dynamics of Trust Reciprocation in Multi-relational Networks*, In Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, pp. 661-665. ACM, 2013.

Journal papers under Review/ in preparation

- **Ayush Singhal**, Atanu Roy, Jaideep Srivastava, *Co-evolution Analysis: a study of Large Multi-relational Social Networks*, in Journal of Data Mining and Digital Humanities (submitted).

- Atanu Roy, **Ayush Singhal**, Jaideep Srivastava, *A Study of dyadic trust*, in ACM Transactions on Internet Technology (submitted).

- **Ayush Singhal**, Jaideep Srivastava, *Meta-content generation for scientific research datasets*, in International Journal of Artificial Intelligence Tools (invited for submission).

- **Ayush Singhal**, Jaideep Srivastava, *A framework for keyword assignment to short text documents*, in Springer Journal on Information Retrieval (in preparation).

- **Ayush Singhal**, Jaideep Srivastava, Ravindra Kasturi. *A context based search framework for scientific research datasets*, in ACM Transactions on the Web (in preparation).

References

[1] Rada Mihalcea and Paul Tarau. Textrank: Bringing order into texts. In *Proceedings of EMNLP*, volume 4. Barcelona, Spain, 2004.

[2] Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. Automatic keyword extraction from individual documents. *Text Mining*, pages 1–20, 2010.

[3] AlchemyAPI. Text analysis by alchemyapi, 2013.

[4] Helen L Brownson. Research on handling scientific information. *Science*, 132(3444):1922–1931, 1960.

[5] Cyril W Cleverdon. The evaluation of systems used in information retrieval. In *Proceedings of the international conference on scientific information*, volume 1, pages 687–698, 1959.

[6] Mortimer Taube, CD Gull, and Irma S Wachtel. Unit terms in coordinate indexing. *American documentation*, 3(4):213–218, 1952.

[7] Hans Peter Luhn. A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of research and development*, 1(4):309–317, 1957.

[8] Paul Switzer. Vector images in document retrieval. *Statistical association methods for mechanized documentation*, pages 163–171, 1965.

[9] Gerard Salton. Automatic information organization and retrieval. 1968.

[10] Joseph John Rocchio. Relevance feedback in information retrieval. 1971.

[11] Gerard Salton and Chung-Shu Yang. On the specification of term values in automatic indexing. *Journal of documentation*, 29(4):351–372, 1973.

[12] Stephen E Robertson. The probability ranking principle in ir. *Journal of documentation*, 33(4):294–304, 1977.

[13] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1):107–117, 1998.

[14] Jon M Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46(5):604–632, 1999.

[15] Silviu Cucerzan and Eric Brill. Spelling correction as an iterative process that exploits the collective knowledge of web users. In *EMNLP*, volume 4, pages 293–300, 2004.

[16] Filip Radlinski and Thorsten Joachims. Query chains: learning to rank from implicit feedback. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 239–248. ACM, 2005.

[17] Fuchun Peng, Nawaaz Ahmed, Xin Li, and Yumao Lu. Context sensitive stemming for web search. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 639–646. ACM, 2007.

[18] Nissan Hajaj Jesse Alpert. We knew the web was big... `http://googleblog.blogspot.com/2008/07/we-knew-web-was-big.htmll`, July 2008.

[19] Pasquale Lops, Marco Gemmis, and Giovanni Semeraro. Content-based recommender systems: State of the art and trends. *Recommender Systems Handbook*, pages 73–105, 2011.

[20] K. Bache and M. Lichman. UCI machine learning repository, 2013.

[21] Jure Leskovec. Stanford large network dataset collection, 2007.

[22] Meiyu Lu, Srinivas Bangalore, Graham Cormode, Marios Hadjieleftheriou, and Divesh Srivastava. A dataset search engine for the research document corpus. In *Data Engineering (ICDE), 2012 IEEE 28th International Conference on*, pages 1237–1240. IEEE, 2012.

[23] M. Balabanović and Y. Shoham. Fab: content-based, collaborative recommendation. *Communications of the ACM*, 40(3):66–72, 1997.

[24] B. Liu and Z. Yuan. Incorporating social networks and user opinions for collaborative recommendation: local trust network based method. In *Proceedings of the Workshop on Context-Aware Movie Recommendation*, pages 53–56. ACM, 2010.

[25] Jiawei Yao, Jiajun Yao, Rui Yang, and Zhenyu Chen. Product recommendation based on search keywords. In *Web Information Systems and Applications Conference (WISA), 2012 Ninth*, pages 67–70. IEEE, 2012.

[26] Stuart E Middleton, David De Roure, and Nigel R Shadbolt. Ontology-based recommender systems. *Handbook on ontologies*, pages 779–796, 2009.

[27] Ido Guy, Naama Zwerdling, Inbal Ronen, David Carmel, and Erel Uziel. Social media recommendation based on people and tags. In *Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 194–201. ACM, 2010.

[28] Tommaso Di Noia, Roberto Mirizzi, Vito Claudio Ostuni, and Davide Romito. Exploiting the web of data in model-based recommender systems. In *Proceedings of the sixth ACM conference on Recommender systems*, pages 253–256. ACM, 2012.

[29] Jonathan L Herlocker, Joseph A Konstan, Loren G Terveen, and John T Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1):5–53, 2004.

[30] Shouvick Mukherjee, Jayesh Vrajlal Bhayani, Jagdish Chand, and Ravi Narasimhan Raj. Keyword recommendation for internet search engines, March 4 2004. US Patent App. 10/794,006.

[31] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670. ACM, 2010.

[32] Ja-Hwung Su, Hsin-Ho Yeh, Philip S Yu, and Vincent S Tseng. Music recommendation using content and context information mining. *Intelligent Systems, IEEE*, 25(1):16–26, 2010.

[33] Z. Gantner, S. Rendle, and L. Schmidt-Thieme. Factorization models for context-/time-aware movie recommendations. In *Proceedings of the Workshop on Context-Aware Movie Recommendation*, pages 14–19. ACM, 2010.

[34] Suthathip Maneewongvatana. A recommendation model for personalized book lists. In *Communications and Information Technologies (ISCIT), 2010 International Symposium on*, pages 389–394. IEEE, 2010.

[35] I. Guy, N. Zwerdling, D. Carmel, I. Ronen, E. Uziel, S. Yogev, and S. Ofek-Koifman. Personalized recommendation of social software items based on social relations. In *Proceedings of the third ACM conference on Recommender systems*, pages 53–60. ACM, 2009.

[36] R. Wetzker, W. Umbrath, and A. Said. A hybrid approach to item recommendation in folksonomies. In *Proceedings of the WSDM'09 Workshop on Exploiting Semantic Annotations in Information Retrieval*, pages 25–29. ACM, 2009.

[37] Chumki Basu, WW Cohen, H Hirsh, and Craig Nevill-Manning. Technical paper recommendation: A study in combining multiple information sources. *arXiv preprint arXiv:1106.0248*, 2011.

[38] P. Winoto, T.Y. Tang, and G.I. McCalla. Contexts in a paper recommendation system with collaborative filtering. *The International Review of Research in Open and Distance Learning*, 13(5):56–75, 2012.

[39] C. Nascimento, A.H.F. Laender, A.S. da Silva, and M.A. Gonçalves. A source independent framework for research paper recommendation. In *Proceedings of the 11th annual international ACM/IEEE joint conference on Digital libraries*, pages 297–306. ACM, 2011.

[40] Qi He, Daniel Kifer, Jian Pei, Prasenjit Mitra, and C Lee Giles. Citation recommendation without author supervision. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 755–764. ACM, 2011.

[41] Qi He, Jian Pei, Daniel Kifer, Prasenjit Mitra, and Lee Giles. Context-aware citation recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 421–430. ACM, 2010.

[42] J. He, J.Y. Nie, Y. Lu, and W. Zhao. Position-aligned translation model for citation recommendation. In *String Processing and Information Retrieval*, pages 251–263. Springer, 2012.

[43] M. Lipczak. Tag recommendation for folksonomies oriented towards individual users. *ECML PKDD Discovery Challenge*, 84, 2008.

[44] M. Lipczak, Y. Hu, Y. Kollet, and E. Milios. Tag sources for recommendation in collaborative tagging systems. In *ECML/PKDD Discovery Challenge Workshop (DC09)*, 2009.

[45] R.L. Cilibrasi and P.M.B. Vitanyi. The google similarity distance. *IEEE Transactions on Knowledge and Data Engineering*, 19(3):370–383, 2007.

[46] Michael Ley and Patrick Reuther. Maintaining an online bibliographical database: The problem of data quality. In *EGC*, pages 5–10, 2006.

[47] Shmuel Nitzan and Ariel Rubinstein. A further characterization of borda ranking method. *Public Choice*, 36(1):153–158, 1981.

[48] M Khabsa, CL Giles, and Ren Zhang. The number of scholarly documents on the public web. *PLoS ONE*, 9(5):e93949, 2014.

[49] James A Lamb. What is wrong with full text searches, 2008.

[50] Ian H Witten, Gordon W Paynter, Eibe Frank, Carl Gutwin, and Craig G Nevill-Manning. Kea: Practical automatic keyphrase extraction. In *Proceedings of the fourth ACM conference on Digital libraries*, pages 254–255. ACM, 1999.

[51] Peter D Turney. Learning algorithms for keyphrase extraction. *Information Retrieval*, 2(4):303–336, 2000.

[52] Glenda Browne. Automatic indexing. *Online Currents, the AusSI Newsletter*, 20(6):4–9, 1996.

[53] Evgeniy Gabrilovich and Shaul Markovitch. Overcoming the brittleness bottleneck using wikipedia: Enhancing text categorization with encyclopedic knowledge. In *AAAI*, volume 6, pages 1301–1306, 2006.

[54] Levent Ertöz, Michael Steinbach, and Vipin Kumar. Finding topics in collections of documents: A shared nearest neighbor approach. *Clustering and Information Retrieval*, 11:83–103, 2003.

[55] Eibe Frank, Gordon W Paynter, Ian H Witten, Carl Gutwin, and Craig G Nevill-Manning. Domain-specific keyphrase extraction. 1999.

[56] Chris Clifton, Robert Cooley, and Jason Rennie. Topcat: data mining for topic identification in a text corpus. *Knowledge and Data Engineering, IEEE Transactions on*, 16(8):949–964, 2004.

[57] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47, 2002.

[58] Thorsten Joachims. *Text categorization with support vector machines: Learning with many relevant features*. Springer, 1998.

[59] Dawn Lawrie, W Bruce Croft, and Arnold Rosenberg. Finding topic words for hierarchical summarization. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 349–357. ACM, 2001.

[60] Maria Fernanda Moura and Solange Oliveira Rezende. Choosing a hierarchical cluster labelling method for a specific domain document collection. *New Trends in Artificial Intelligence*, pages 812–823, 2007.

[61] Chin-Yew Lin. Knowledge-based automatic topic identification. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, pages 308–310. Association for Computational Linguistics, 1995.

[62] Sabrina Tiun, Rosni Abdullah, and Tang Enya Kong. Automatic topic identification using ontology hierarchy. In *Computational Linguistics and Intelligent Text Processing*, pages 444–453. Springer, 2001.

[63] Oren Zamir and Oren Etzioni. Web document clustering: A feasibility demonstration. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 46–54. ACM, 1998.

[64] Kazi Saidul Hasan and Vincent Ng. Conundrums in unsupervised keyphrase extraction: making sense of the state-of-the-art. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 365–373. Association for Computational Linguistics, 2010.

[65] Mostafa M Hassan, Fakhri Karray, and Mohamed S Kamel. Automatic document topic identification using wikipedia hierarchical ontology. In *Information Science, Signal Processing and their Applications (ISSPA), 2012 11th International Conference on*, pages 237–242. IEEE, 2012.

[66] Sonal Jain and Jyoti Pareek. Automatic topic (s) identification from learning material: An ontological approach. In *Computer Engineering and Applications (ICCEA), 2010 Second International Conference on*, volume 2, pages 358–362. IEEE, 2010.

[67] Johannes Hoffart, Fabian M Suchanek, Klaus Berberich, and Gerhard Weikum. Yago2: a spatially and temporally enhanced knowledge base from wikipedia. *Artificial Intelligence*, 194:28–61, 2013.

[68] Ho Chung Wu, Robert Wing Pong Luk, Kam Fai Wong, and Kui Lam Kwok. Interpreting tf-idf term weights as making relevance decisions. *ACM Transactions on Information Systems (TOIS)*, 26(3):13, 2008.

[69] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

[70] Susan T Dumais, G Furnas, T Landauer, S Deerwester, S Deerwester, et al. Latent semantic indexing. In *Proceedings of the Text Retrieval Conference*, 1995.

[71] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. 1999.

[72] DBLP. Faceted dblp.

[73] Hsinchun Chen, Wingyan Chung, Jennifer Jie Xu, Gang Wang, Yi Qin, and Michael Chau. Crime data mining: a general framework and some examples. *Computer*, 37(4):50–56, 2004.

[74] Nathan Eagle and Alex Pentland. Reality mining: sensing complex social systems. *Personal and ubiquitous computing*, 10(4):255–268, 2006.

[75] Diane J Cook and Lawrence B Holder. *Mining graph data*. Wiley-Interscience, 2006.

[76] Bamshad Mobasher, Robert Cooley, and Jaideep Srivastava. Automatic personalization based on web usage mining. *Communications of the ACM*, 43(8):142–151, 2000.

[77] Ron Kohavi and Foster Provost. *Applications of data mining to electronic commerce*. Springer, 2001.

[78] Aaron M Cohen and William R Hersh. A survey of current work in biomedical text mining. *Briefings in bioinformatics*, 6(1):57–71, 2005.

[79] Auroop R Ganguly and Karsten Steinhaeuser. Data mining for climate change and impacts. In *Data Mining Workshops, 2008. ICDMW'08. IEEE International Conference on*, pages 385–394. IEEE, 2008.

[80] Michael Steinbach, Pang-Ning Tan, Vipin Kumar, C Potter, S Klooster, and A Torregrosa. Data mining for the discovery of ocean climate indices. In *Proc of the Fifth Workshop on Scientific Data Mining*, 2002.

[81] Lars Backstrom, Paolo Boldi, Marco Rosa, Johan Ugander, and Sebastiano Vigna. Four degrees of separation. In *Proceedings of the 3rd Annual ACM Web Science Conference*, pages 33–42. ACM, 2012.

[82] Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. Mining multi-label data. In *Data mining and knowledge discovery handbook*, pages 667–685. Springer, 2010.

[83] Julian McAuley, Jure Leskovec, and Dan Jurafsky. Learning attitudes and attributes from multi-aspect reviews. In *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, pages 1020–1025. IEEE, 2012.

[84] José Kahan and Marja-Ritta Koivunen. Annotea: an open rdf infrastructure for shared web annotations. In *Proceedings of the 10th international conference on World Wide Web*, pages 623–632. ACM, 2001.

[85] Ka-Ping Yee. Critlink: Advanced hyperlinks enable public annotation on the web. *University of California, Berkeley*, 2002.

[86] Koen Holtman. The futplex system. In *Proceedings of the ERCIM workshop on CSCW and the Web*, page 3, 1996.

[87] M Rscheisen and C Mogensen. Commentor: Scalable architecture for shared www annotations as a platform for value-added providers. Technical report, Stanford University Technical Report, 1994.

[88] Siegfried Handschuh, Steffen Staab, and Alexander Maedche. Cream: creating relational metadata with a component-based, ontology-driven annotation framework. In *Proceedings of the 1st international conference on Knowledge capture*, pages 76–83. ACM, 2001.

[89] Maria Vargas-Vera, Enrico Motta, John Domingue, Mattia Lanzoni, Arthur Stutt, and Fabio Ciravegna. Mnm: Ontology driven semi-automatic and automatic support for semantic markup. In *Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web*, pages 379–391. Springer, 2002.

[90] Siegfried Handschuh, Steffen Staab, and Fabio Ciravegna. S-cream:semi-automatic creation of metadata. In *Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web*, pages 358–372. Springer, 2002.

[91] Borislav Popov, Atanas Kiryakov, Angel Kirilov, Dimitar Manov, Damyan Ognyanoff, and Miroslav Goranov. Kim–semantic annotation platform. In *The Semantic Web-ISWC 2003*, pages 834–849. Springer, 2003.

[92] Stephen Dill, Nadav Eiron, David Gibson, Daniel Gruhl, R Guha, Anant Jhingran, Tapas Kanungo, Sridhar Rajagopalan, Andrew Tomkins, John A Tomlin, et al. Semtag and seeker: Bootstrapping the semantic web via automated semantic annotation. In *Proceedings of the 12th international conference on World Wide Web*, pages 178–186. ACM, 2003.

[93] T Hubbard, Daniel Barker, Ewan Birney, Graham Cameron, Yuan Chen, L Clark, T Cox, J Cuff, Val Curwen, Thomas Down, et al. The ensembl genome database project. *Nucleic acids research*, 30(1):38–41, 2002.

[94] Susan Tweedie, Michael Ashburner, Kathleen Falls, Paul Leyland, Peter McQuilton, Steven Marygold, Gillian Millburn, David Osumi-Sutherland, Andrew Schroeder, Ruth Seal, et al. Flybase: enhancing drosophila gene ontology annotations. *Nucleic acids research*, 37(suppl 1):D555–D559, 2009.

[95] Todd W Harris, Nansheng Chen, Fiona Cunningham, Marcela Tello-Ruiz, Igor Antoshechkin, Carol Bastiani, Tamberlyn Bieri, Darin Blasiar, Keith Bradnam, Juancarlos Chan, et al. Wormbase: a multi-species resource for nematode biology and genomics. *Nucleic acids research*, 32(suppl 1):D411–D417, 2004.

[96] Daniel L Rubin, Pattanasak Mongkolwat, Vladimir Kleper, Kaustubh Supekar, and David S Channin. Medical imaging on the semantic web: Annotation and image markup. In *AAAI Spring Symposium Series, Semantic Scientific Knowledge Integration*, 2008.

[97] Joan Beaudoin. Folksonomies: Flickr image tagging: Patterns made visible. *Bulletin of the American Society for Information Science and Technology*, 34(1):26–29, 2007.

[98] Qiaozhu Mei, Dong Xin, Hong Cheng, Jiawei Han, and ChengXiang Zhai. Generating semantic annotations for frequent patterns with context analysis. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 337–346. ACM, 2006.

[99] Ziqi Zhang, Sam Chapman, and Fabio Ciravegna. A methodology towards effective and efficient manual document annotation: addressing annotator discrepancy and annotation quality. In *Knowledge Engineering and Management by the Masses*, pages 301–315. Springer, 2010.

[100] Paul Heymann, Georgia Koutrika, and Hector Garcia-Molina. Can social bookmarking improve web search? In *Proceedings of the International Conference on Web search and web data mining*, pages 195–206. ACM, 2008.

[101] Daniel Ramage, Paul Heymann, Christopher D Manning, and Hector Garcia-Molina. Clustering the tagged web. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, pages 54–63. ACM, 2009.

[102] Zheng Liu and Hua Yan. Annotating flickr photos by manifold-ranking based tag ranking and tag expanding. In *Information Computing and Applications*, pages 221–228. Springer, 2010.

[103] Peng Li, Bin Wang, Wei Jin, and Yachao Cui. User-related tag expansion for web document clustering. In *Advances in Information Retrieval*, pages 19–31. Springer, 2011.

[104] Irene Cramer, Tonio Wandmacher, and Ulli Waltinger. Exploring resources for lexical chaining: A comparison of automated semantic relatedness measures and human judgments. In *Modeling, Learning, and Processing of Text Technological Data Structures*, pages 377–396. Springer, 2012.

[105] Peter Skomoroch. Some datasets available on the web. url http://www.datawrangling.com/some-datasets-available-on-the-web, 2008.

[106] Mukund Deshpande and George Karypis. Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems (TOIS)*, 22(1):143–177, 2004.

[107] D Seung and L Lee. Algorithms for non-negative matrix factorization. *Advances in Neural Information Processing Systems*, 13:556–562, 2001.

[108] Chih-Jen Lin. Projected gradient methods for nonnegative matrix factorization. *Neural computation*, 19(10):2756–2779, 2007.

[109] Finn Årup Nielsen, Lars Kai Hansen, and Daniela Balslev. Mining for associations between text and brain activation in a functional neuroimaging database. *Neuroinformatics*, 2(4):369–379, 2004.

[110] Morten Mørup, Lars Kai Hansen, Christoph S Herrmann, Josef Parnas, and Sidse M Arnfred. Parallel factor analysis as an exploratory tool for wavelet transformed event-related eeg. *NeuroImage*, 29(3):938–947, 2006.

[111] Lu Zhiqiang, Shao Werimin, and Yu Zhenhua. Measuring semantic similarity between words using wikipedia. In *International Conference on Web Information Systems and Mining, 2009. WISM 2009.*, pages 251–255. IEEE, 2009.

[112] Murray Campbell, Alexander Haubold, Ming Liu, Apostol Natsev, John R Smith, Jelena Tesic, Lexing Xie, Rong Yan, and Jun Yang. Ibm research trecvid-2007 video retrieval system. In *TRECVID*, 2007.

[113] Guo-Jun Qi, Xian-Sheng Hua, Yong Rui, Jinhui Tang, Tao Mei, and Hong-Jiang Zhang. Correlative multi-label video annotation. In *Proceedings of the 15th International Conference on Multimedia*, pages 17–26. ACM, 2007.

[114] Lei Wu, Steven CH Hoi, Rong Jin, Jianke Zhu, and Nenghai Yu. Distance metric learning from uncertain side information with application to automated photo tagging. In *Proceedings of the 17th ACM International Conference on Multimedia*, pages 135–144. ACM, 2009.

[115] Yanan Liu, Fei Wu, Yueting Zhuang, and Jun Xiao. Active post-refined multimodality video semantic concept detection with tensor representation. In *Proceedings of the 16th ACM International Conference on Multimedia*, pages 91–100. ACM, 2008.

[116] Fei Wu, Yahong Han, Qi Tian, and Yueting Zhuang. Multi-label boosting for image annotation by structural grouping sparsity. In *Proceedings of the International Conference on Multimedia*, pages 15–24. ACM, 2010.

[117] Shaoting Zhang, Junzhou Huang, Yuchi Huang, Yang Yu, Hongsheng Li, and Dimitris N Metaxas. Automatic image annotation using group sparsity. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010*, pages 3312–3319. IEEE, 2010.

[118] Jinhui Tang, Xian-Sheng Hua, Guo-Jun Qi, Yan Song, and Xiuqing Wu. Video annotation based on kernel linear neighborhood propagation. *IEEE Transactions on Multimedia*, 10(4):620–628, 2008.

[119] Xun Yuan, Xian-Sheng Hua, Meng Wang, and Xiu-Qing Wu. Manifold-ranking based video concept detection on large database and feature pool. In *Proceedings of the 14th annual ACM International Conference on Multimedia*, pages 623–626. ACM, 2006.

[120] Oksana Yakhnenko and Vasant Honavar. Annotating images and image objects using a hierarchical dirichlet process model. In *Proceedings of the 9th International Workshop on Multimedia Data Mining: held in conjunction with the ACM SIGKDD 2008*, pages 1–7. ACM, 2008.

[121] Tomer Hertz, Aharon Bar-Hillel, and Daphna Weinshall. Learning distance functions for image retrieval. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 2, pages II–570. IEEE, 2004.

[122] Matthieu Guillaumin, Thomas Mensink, Jakob Verbeek, and Cordelia Schmid. Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation. In *IEEE 12th International Conference on Computer Vision, 2009*, pages 309–316. IEEE, 2009.

[123] Andrew Gilbert and Richard Bowden. A picture is worth a thousand tags: automatic web based image tag expansion. In *Computer Vision–ACCV 2012*, pages 447–460. Springer, 2013.

[124] Börkur Sigurbjörnsson and Roelof Van Zwol. Flickr tag recommendation based on collective knowledge. In *Proceedings of the 17th International Conference on World Wide Web*, pages 327–336. ACM, 2008.

[125] Xirong Li, Cees GM Snoek, and Marcel Worring. Learning social tag relevance by neighbor voting. *IEEE Transactions on Multimedia*, 11(7):1310–1322, 2009.

[126] Sare Gul Sevil, Onur Kucuktunc, Pinar Duygulu, and Fazli Can. Automatic tag expansion using visual similarity for photo sharing websites. *Multimedia Tools and Applications*, 49(1):81–99, 2010.

[127] Yang Yang, Zi Huang, Heng Tao Shen, and Xiaofang Zhou. Mining multi-tag association for image tagging. *World Wide Web*, 14(2):133–156, 2011.

[128] Dr J. Mark Tippett. The importance of data, 1999-2012.

[129] Ayush Singhal, Ravindra Kasturi, Vidyashankar Sivakumar, and Jaideep Srivastava. Leveraging web intelligence for finding interesting research datasets. In *Web Intelligence*

(WI) and Intelligent Agent Technologies (IAT), 2013 IEEE/WIC/ACM International Joint Conferences on, volume 1, pages 321–328. IEEE, 2013.

[130] J Pederson. Query understanding at bing. *Invited talk, SIGIR*, 2010.

[131] Daniel E Rose and Danny Levinson. Understanding user goals in web search. In *Proceedings of the 13th international conference on World Wide Web*, pages 13–19. ACM, 2004.

[132] Stefan Siersdorfer, Jose San Pedro, and Mark Sanderson. Automatic video tagging using content redundancy. In *Proceedings of the 32nd International ACM SIGIR conference on Research and development in information retrieval*, pages 395–402. ACM, 2009.

[133] Alan Emtage and Peter Deutsch. Archie: An electronic directory service for the internet. In *Proceedings of the Winter 1992 USENIX Conference*, pages 93–110, 1992.

[134] Farhad Anklesaria, Mark McCahill, Paul Lindner, David Johnson, Daniel Torrey, and B Albert. The internet gopher protocol (a distributed document search and retrieval protocol). 1993.

[135] Saeid Asadi and Hamid R Jamali. Shifts in search engine development: A review of past, present and future trends in research on search engines. *Webology*, 1(2), 2004.

[136] Gus Venditto. Search engine showdown-alta vista, excite, infoseek, lycos, open text, web-crawler, www worm. they all claim to be the best, so we put them to the test. *Internet World*, 7(5):78–87, 1996.

[137] Bruno Martins and Mário J Silva. Spelling correction for search engine queries. In *Advances in Natural Language Processing*, pages 372–383. Springer, 2004.

[138] Wessel Kraaij and Renée Pohlmann. Viewing stemming as recall enhancement. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 40–48. ACM, 1996.

[139] W Cavnar. Using an n-gram-based document representation with a vector processing retrieval model. *NIST SPECIAL PUBLICATION SP*, pages 269–269, 1995.

[140] Thanh Tran, Philipp Cimiano, Sebastian Rudolph, and Rudi Studer. Ontology-based interpretation of keywords for semantic search. In *The Semantic Web*, pages 523–536. Springer, 2007.

[141] Sougata Mukherjea, Kyoji Hirata, and Yoshinori Hara. Towards a multimedia world-wide web information retrieval engine. *Computer networks and ISDN systems*, 29(8):1181–1191, 1997.

[142] Ayush Singhal and Jaideep Srivastava. Data extract: Mining context from the web for dataset extraction. *International Journal of Machine Learning and Computing*, 3(2):219–223, 2013.

[143] Karen Sparck Jones, Steve Walker, and Stephen E. Robertson. A probabilistic model of information retrieval: development and comparative experiments: Part 1. *Information Processing & Management*, 36(6):779–808, 2000.

[144] W Bruce Croft, Donald Metzler, and Trevor Strohman. *Search engines: Information retrieval in practice*. Addison-Wesley Reading, 2010.