TeKnowBase
A Tool for enrichment of textbooks using discussion forums

Student Name: Amani Kongara

IIIT-D-MTech-CS-DE-12-040

December, 2014

Indraprastha Institute of Information Technology
New Delhi

Thesis Committee
Dr. Srikanta Bedathur (Chair)
Dr. Maya Ramanath
Dr. Debajyoti Bera

Submitted in partial fulfillment of the requirements
for the Degree of M.Tech. in Computer Science,
with specialization in Data Engineering

Certificate

This is to certify that the thesis titled **"*TeKnowBase*, A tool for enrichment of textbooks using discussion forums"** submitted by **Kongara Amani** for the partial fulfillment of the requirements for the degree of *Master of Technology* in *Computer Science & Engineering* is a record of the bonafide work carried out by her / him under my / our guidance and supervision in the Security and Privacy group at Indraprastha Institute of Information Technology, Delhi. This work has not been submitted anywhere else for the reward of any other degree.

Dr. Srikanta Bedathur
Indraprastha Institute of Information Technology, New Delhi

Abstract

Several knowledge resources are available both online and off line in learning technical topics. Textbooks act as a basic reference with their general organization into sections where each section is dedicated in explaining a single topic. Other online resources like Wikipedia articles and its topic hierarchy help the users in structured learning of a specific technical topic by providing them with the details on advanced applications. Various discussion forums aid the users in clarifying the doubts on real world implementation details of the technical topics. For an effective learning of technical topics, all these features are to be curated. By making use of online knowledge resources,through *TeKnowBase*, we present our early explorations in trying to bridge the gaps in textbooks by providing more details on the technical topic to be learnt. To extract the discussions on the details of real world implementation and advanced applications of the topics, we make use the data in StackOverflow, a discussion forum on computer programming. Extraction of relevant discussions on a specific topic is performed using query expansion with various keywords. Two approaches are used in the query expansion, one using the keywords from Wikipedia category hierarchy and the other using the keywords describing context of a topic in textbook. A database of topics is built from the index terms and their parent topics of textbooks. Using these parent topics as keywords, we expand our search in Stackoverflow for extracting more relevant discussions on the selected topic. Keywords from the above mentioned resources helps in refining the search and extraction of relevant discussions by setting the context from the textbook to learn a topic and by using the category hierarchy of Wikipedia. The results obtained from the expanded query search are evaluated manually. Both the techniques showed an improvement over the normal keyword search in extracting relevant discussions when queried using the search framework of Lucene and evaluated using the graded evaluation measure of DCG@k.

Acknowledgments

First and foremost, I offer my sincerest gratitude to my supervisor, Dr. Srikanta Bedathur, who has supported me throughout my thesis with his patience and knowledge. I am highly obliged for the motivation and enthusiasm provided by him throughout the research and writing of this thesis. I will always be grateful to him throughout my life for helping me start my career in Information Retrieval.

Next, I express my sincere thanks and gratitude towards Dr. Maya Ramanath for giving me an opportunity to work with her. Her encouragement and insightful comments helped me a lot in improving my approach towards the problem.

I would like to thank all my friends for their invaluable help and moral support.

Contents

List of Figures

List of Tables

Keywords: Technical Topic Summarization, Co-occurring Topics, Wikipedia, Category Hierarchy, String Similarity, Jaccard coefficient, Relevance, Query expansion

Chapter 1

Introduction

1.1 Motivation

Learning about technical topics is a continuous process and is often difficult for both students, as well as experienced researchers. The difficulty comes, in part, from the lack of pre-requisite knowledge as well as context to study a given subject. For example, learning about *query plans* in the context of *database management systems* cannot occur in a vacuum. To set the context, first, the user has to know about *database system, SQL, the query processing pipeline and query optimization*. Most importantly, he has to have good knowledge about *relational algebra operators* before one can hope to understand *query plans*. Textbooks offer a systematic treatment of a particular topic, but are not really the solution when one wants a quick overview of a topic [1], or when the topic to be learnt needs multiple pre-requisites.

Learning how people assess the value of central concepts of *Computer Science* [2] depending on the disciplinary and pedagogical knowledge on the concept helps in improving the quality of teaching. This may also help in teachers training program for effective teaching of concepts in various domains. Several frameworks were also been constructed for the design and development of the lessons to meet the specific goals and knowledge structures [3].

Domain specific online discussion forums serve as a platform to discuss about various issues faced in learning a specific topic. Several problems like the real world implementation details of a topic, application aspect of the topic, potential bugs in implementation the topic, doubts in proper understanding of the topic, its uses, advantages-disadvantages in using the topic and so on faced by many users are solved by the co-users of the forum. This helps in group learning of various topics and help in overall improvement in the knowledge over any topic. Most of the times, these online discussion forums help in improving the education in CS [4].

In learning the topics in a specific domain, textbooks themselves may not be a good option in having a complete understanding of the concepts behind the topic. Various real world implementation and application details are needed for the clear understanding of the topics. Textbooks lack in providing the full fledged details on a topic due to the space constraint. So, various parameters like the prior knowledge of the user about the topic, the extent to which the user may want to learn the topic, etc are to be considered for aiding the user in learning the topic. Online discussion forums provides all the issues faced by various users in learning several topics. So, the knowledge in the discussion forums can be used as a resource to help the users in learning the topic. This can be done by providing the users with relevant discussions on the topic to be learnt.

1.2 Problem definition

In this project, we aim to provide users with a system that assists learning of new topics in the technical domain by linking textbook topics with the discussion forums.

In general many users discuss the issues on application details and implementation in the domain specific discussion forums. Textbooks lack in providing such details because of the space constraint. We made use of the knowledge in discussion forums in bridging the gap in drawbacks of textbooks by making use of context provided by them in learning a technical topic.

StackOverflow is one of such domain specific discussion forum where users post their issues in leaning various technical topics. We make use of those discussions of implementation and application details in well understanding of a specific topic. We try to provide a summary to the user on the technical topic to be learnt by extracting relevant discussions from StackOverflow data.

The system *TeKnowBase* recommends the relevant discussions from StackOverflow through the knowledge base derived from textbooks and Wikipedia. We propose two techniques in extracting the useful discussions of a technical topic: One is by making use of the given context of a topic in the textbook and the other is through the Wikipedia category hierarchy of a topic. Both the methods showed an improvement in extracting the relevant discussions on a topic based on the context, over the discussions having just the occurrence of the topic. We evaluated the relevance of the results manually and then calculated the effectiveness of each technique using the DCG score at top-k results for each topic.

1.3 Designed Framework

We propose a framework in providing the users with a set of useful discussions on technical topic to be learnt. Various steps in designing the framework of *TeKnowBase* are described briefly as follows.

Data Preprocessing

The data from textbooks is preprocessed for the preparation of dataset of topics. A database of topics is made from the BoBI(Back of Book Index) and the ToC(Table of Contents) of textbooks. The chapter titles and the index terms are to be cleaned and made consistent to be stored into a database. Other resources used in building *TeKnowBase* are Wikipedia and StackOverflow, an online discussion forum. Data from both the datasets is preprocessed prior to their usage in the framework.

Category and page titles under the chosen areas are captured from the Wikipedia category hierarchy. This makes a technical topic database from Wikipedia which is further used in the designed framework. Discussions from the resource of StackOverflow are extracted based on the PostIds. The data of StackOverflow is preprocessed using the search framework of Lucene. More details of the preprocessing and data set preparation are in chapter 4.

Designed Framework

The framework designed contains three phases in extraction of relevant discussions on a given technical topic: Extraction Phase, Query Expansion Phase and Retrieval Phase respectively as a pipeline. Query Expansion phase is performed using two techniques of expanding a query in retrieving more relevant discussions from StackOverflow. This is further discussed in section 3.

Given a technical topic, the parent topics(chapters) are extracted from the textbook in the Extraction Phase of the framework. The corresponding Wikipedia category hierarchy is also retrieved from the prepared database. The parent topics extracted from textbook are used in describing the context of the technical topic with a set of keywords extracted from the parent topics. These keywords are used in the Query Expansion Phase. The Wikipedia category hierarchy is also used in the Query Expansion Phase. The queries thus formed are passed to the Lucene search and the relevant discussions are retrieved in the Retrieval Phase of the framework.

The discussions retrieved using the two techniques in Query Expansion Phase are evaluated manually by defining the relevance levels ranging from 0-4(0 being the least relevant). The scores for each discussion retrieved are given by various users and an average Discounted Cumulative Gain is calculated for top-k results. The results obtained using the query expansion using Wikipedia categories as keywords showed 25% increase in the average DCG scores. The results obtained using the keywords describing the context showed 18% increase in the average DCG scores.

Chapter 2

Related Work

In providing a high quality education, textbooks are considered to be the vital input which drives student learning. Despite few drawbacks of textbooks in providing the detailed information on the concepts, they are well known for their structured content. The study on content present in textbooks identifies the properties of good textbooks as focus of each section in explaining each concept, unity and sequentiality of the concepts [5]. It also reveals the advantages of the structured information provided in the textbooks.

With the considerable importance of textbooks in effective learning of concepts, research has been made on overcoming the drawbacks of textbooks. Bridging the gap between the disadvantage of lacking the detailed information and making use of the well organized information present in the textbooks was identified early. There has been research on the finding the deficient areas in the textbooks and enrichment of textbooks in such areas. [1] discusses the enrichment of textbooks through data mining. It recognizes the lack of detailed information on various topics in textbooks due to space constraint and tries to find such areas in textbooks. They proposed a method of enriching textbooks by finding authoritative material from the web for the deficient areas and augment them with the links to suitable web content in such areas. Key concepts from textbooks were extracted using parts of speech tagging and the corresponding Wikipedia page links are provided for each of the key concepts. This helps in detailed learning of the sub concepts in textbooks.

The problem of textbooks being largely text-oriented and the lack of proper visual material were addressed by [6]. Visualization of the concepts helps in better understanding of the topics. For this appropriate images are to be identified for the corresponding concepts. Web is used as a source in extracting relevant image for the concept. Linking encyclopedic information to educational materials was performed in improving both the quality of knowledge and the time needed to obtain the knowledge. Online images were mined using two image mining algorithms and relevant images were added to the corresponding sections in the textbooks.

Other way of helping the users can be providing them with the videos on the concepts to be learnt. This is a way of visualizing the concepts where videos were used in efficient understanding of the topics described in the textbooks. Videos relevant for each section are identified and are to be ensured that at least one concept is being discussed in it. [7] uses videos from the web, generates candidate videos for each section of the textbooks and augments these sections with the best possible videos based on the concept phrases present in them.

While images, videos help in better understanding of a topic, the real world implementations are less discussed in neither of them. Application details of the topic in various other areas which lack in textbooks are not much discussed in any of the above mentioned ways of enrichment. While those details helps in knowing the real-world applications of concepts, they can also be used in the effective learning of the topic. Doubts being posted on those areas can be found in domain specific online discussion forums. One way of enriching the textbooks and thereby aiding the user in effective learning of the concepts can be by providing more relevant discussions on the real-world issues in learning the topic.

Online discussion forums have become a popular knowledge source for sharing information or solving problems. They have become a new way of teaching. [8] studies about the involvement of a teacher in posting the discussions in an online forum and the responses to the posts. It discusses about the patterns in the posts and responses and there by judging the quality of learning. [9] shows how to model online knowledge sharing processes and identify patterns related to effective knowledge sharing. It shows that various mining techniques and business process modeling can be used in analyzing online knowledge sharing activities.

Various methods of query expansions were used in improving the effectiveness of the search. Some of them are automatic query expansion, using external resources like dictionaries, logs, snippets, search result documents and so on. [11] discusses the query expansion using external evidence, especially the hints obtained from external web search engines to expand the original query. Semantic enrichment of queries is also performed by various available resources online. [12] uses Wikipedia and DBpedia as the external knowledge base for the semantic expansion of the query.

We present our basic trials of assisting the user in learning the technical topics present in the textbooks with the discussions on the real world applications and implementation details of the topic by mining an online discussion forum.

Chapter 3

Architecture of *TeKnowBase* System

A pipeline approach has been used in recommending the relevant StackOverflow discussions. The framework of the system is shown in the figure 3.1. It contains three phases, extraction phase, query expansion phase and retrieval phase.

TeKnowBase takes a technical topic as input and retrieves relevant discussions from StackOverflow. The input is fed to the extraction phase of *TeKnowBase* where the parent topics of the queried topic from textbook chapter hierarchy are extracted. The Wikipedia category or the article to which the queried topic belongs is extracted along with the articles titles under the category. The StackOverflow discussions which contain the queried topic are extracted in the extraction phase of the pipeline.

The query expansion phase tries to extract the more relevant StackOverflow discussions. It makes use of the Wikipedia category hierarchy of the queried topic. The presence of the sibling topics for the queried topic along with itself increases the scope of a StackOverflow discussion to be relevant in learning about the topic. This phase also makes use of the words which are present in the textbook around the queried topic within in a window. These words describe the context of the queried topic to learn and therefore help in extracting those discussions of StackOverflow which discuss the queried topic within a context.

The final phase of retrieval searches with the expanded query to retrieve more relevant discussions. The discussions retrieved in this phase contains the posts with context derived from textbook for the queried topic. Discussions containing the Wikipedia category hierarchy for the given topic are also retrieved.

The architecture of the *TeKnowBase* system is shown in the 3.1. We discuss each module in detail as follows.

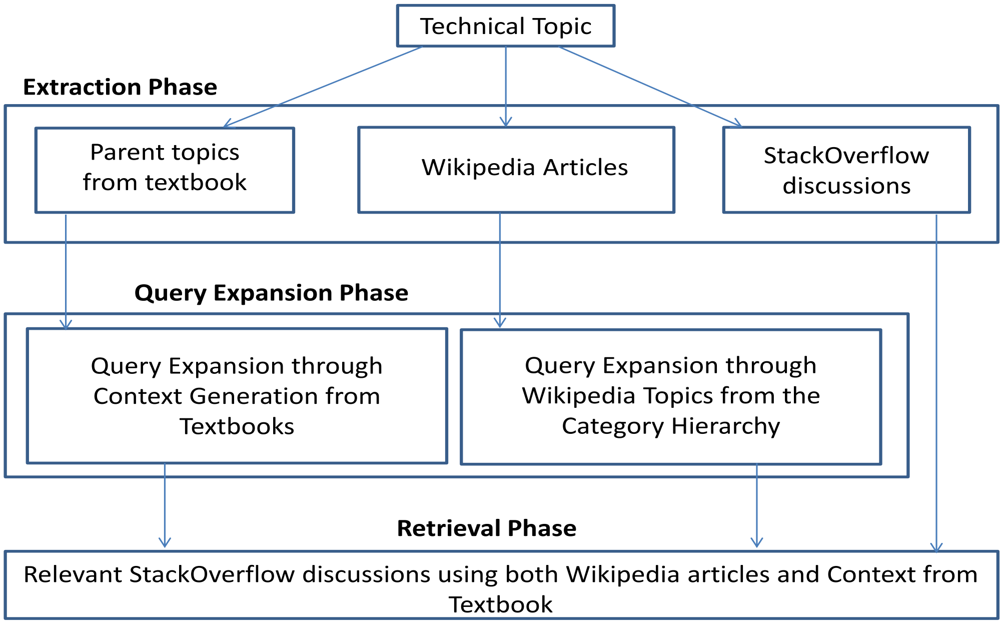
Technical Topic
Extraction Phase
Parent topics from textbook
Wikipedia Articles
StackOverflow discussions
Query Expansion Phase
Query Expansion through Context Generation from Textbooks
Query Expansion through Wikipedia Topics from the Category Hierarchy
Retrieval Phase
Relevant StackOverflow discussions using both Wikipedia articles and Context from Textbook

Figure 3.1: Framework of *TeKnowBase*

3.1 Extraction Phase

In this phase, the related topics for the queried technical topic are extracted from the three resources used in this project. Textbooks provide a context to learn a technical topic. For example, in the area of "information retrieval", we learn the topic of "hashing" in the context of "index compression". The topic "hashing" is defined under the chapter of "index compression" in this textbook. As mentioned in the section 3, a database capturing the topic hierarchy of the textbook is used in extracting the parent topics of the queried topic. The set of parent topics extracted helps in defining the context in which the queried technical topic is to be learnt.

The titles of chapters or sub chapters, under which the queried technical topic falls, are extracted from the database of textbook topic hierarchy. The depth up to which the parent topics are to be extracted is chosen to be two. So the parent and grand parent of the queried topic are retrieved from the database. These are used in the further phases of the architecture for providing the users with relevant online discussions.

We extracted the title of Wikipedia category or the article for the queried topic if present. This is used in retrieving more relevant StackOverflow discussions on the given topic in the further stages of the pipeline.

In the process of providing the users with real world application details on a topic to be learnt, we extract the discussions on StackOverflow. The discussions on the StackOverflow over the queried topic are also extracted in this phase using Lucene, a high-performance, full-featured

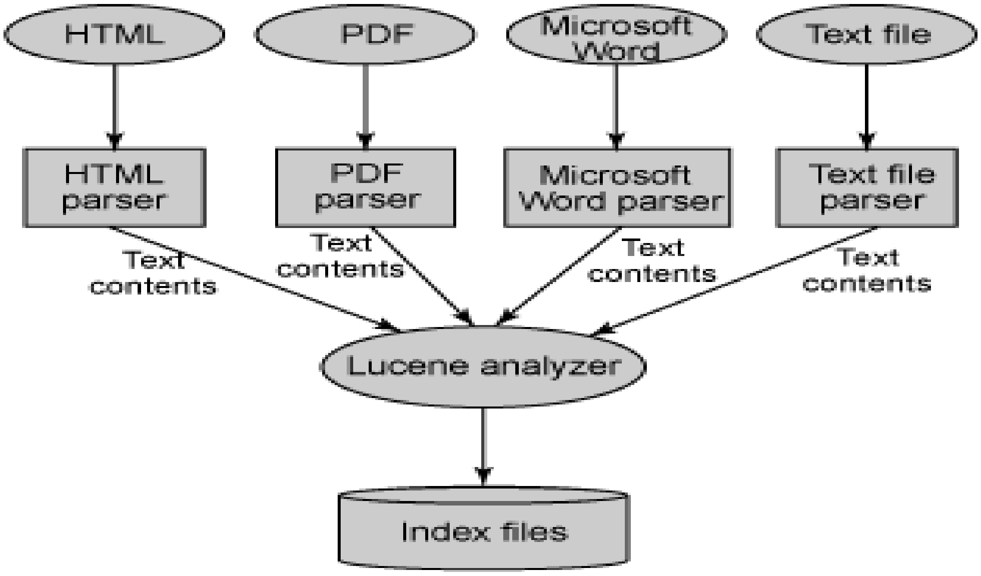

Figure 3.2: Indexing in Lucene

text engine library in Java. The indexing, search and retrieval of discussions using Lucene is briefly described below. 3.2 shows the indexing of Lucene.

3.1.1 Indexing using Lucene

Indexing is the process of breaking down the data in documents into chunks and store the chunks as a name/value pair. The whole process involves various steps which are described as follows.

Parsing the documents:

An analyzer class in Lucene does the job of parsing the documents for making the data in them to indexable tokens or keywords. The type of analyzer can be specified among various types as shown below.

Analyzer	**Description**
StandardAnalyzer	A sophisticated general-purpose analyzer.
WhitespaceAnalyzer	A very simple analyzer that just separates tokens using white space.
StopAnalyzer	Removes common English words that are not usually useful for indexing.
SnowballAnalyzer	An interesting experimental analyzer that works on word roots (a search on rain should a

Table 3.1: Lucene Analyzers

Adding a Document/object to Index:

Each token obtained from parsing the documents is stored with a list of document ids. These documents ids are those which contain the corresponding token.

3.1.2 Text search using Lucene Index:

After indexing, a query parser is used to parse the user query string and passed on to index searcher as the input. Based on the query and the prebuilt Lucene index, index searcher identifies the matching documents and returns them as an results. These results are thus retrieved in the order of relevance.

3.2 Query Expansion Phase

In this phase, we try to refine our search of relevant discussions over StackOverflow for the queried topic. The results obtained for the direct phrasal query search of the technical topic contain a wide range of discussions on the topic. They many even contain faraway applications of the topic which may lead to topic drifts. For example, a discussion of the application of penalty pricing by Bayesian networks may not be useful for the users who doesn't know Bayesian networks. Such posts may not be of assistance for the naive users in learning the topic. Posts with huge topic drifts are less useful for the users who want to know the implementation aspect of the topic. To overcome this, we use the traditional query expansion technique by adding extra terms to the initial query. The following are the two methods used in query expansion [13].

3.3 Query expansion using Wikipedia Category Hierarchy

Wikipedia has a well classified topic hierarchy of articles on various fields. It covers several domains by dividing the entities into categories, sub-categories and pages. The domain of technical topics is very well organized into categories and sub-categories with almost every technical topic as a separate page categorized under one more categories or sub-categories. These categories under which a technical topic falls restrict the areas to which a topic belongs and there by helps in reducing the topic drifts to some extent. [14] used Wikipedia articles in searching relevant books. We use Wikipedia article titles to expand the queries [15].

Wikipedia category hierarchy of the queried topic is used in retrieving the relevant discussions from StackOverflow. We expand the phrasal query of just the technical topic by adding few more topics to it. These topics are chosen to be the Wikipedia article titles which are sibling pages to the queried topic under the same categories to which it belongs. For example, if the topic to be learnt is smoothing, it is appended with the topics of Data analysis, image processing, signal processing, statistical charts and diagrams and time series analysis. All these topics are the categories to which smoothing belong. These topics help in narrowing the search to these specific areas and there by reduces the topic drifts to a large extent. For example adding these terms to the query may not retrieve the posts which discuss font smoothing which are retrieved when smoothing alone is passed as query. This can be explained as follows.

<"a b">is expanded as <"a b" "c d" "e" "f g">where "a b" is the topic to be learnt and "c d", "e", "f g" are the categories to which "a b" belong or the pages under "a b", if "a b" itself is a category.

3.4 Query expansion through contextual terms from Textbooks:

Textbooks provide the context in learning a technical. The text around a technical topic in a textbook describes the context in which the technical topic is used. For example, the topic encoding itself may refer to various areas like communication, storage etc . But learning encoding in the context of indexing and compression narrows down the context and helps the user in learning. [16] uses probabilistic way of concept based query expansion. We make use of the context provided by textbooks in extracting more relevant posts from StackOverflow.

In this method of query expansion, contextual words are added to the query phrase. We discuss the process of extracting the contextual words here.
Unlike [16] the parent topics extracted in the first phase of the framework are used in this phase of query expansion. The text with in a window around the queried topic under the corresponding chapters of parent topics is extracted. The size of window is set to be 50 characters before and after the occurrence of the topic in the text. From the extracted text, n-grams of size 3 are generated and are mapped with the Wikipedia article titles in the database of technical topics. The mapping of terms is clearly discussed in the section 3.4.1

The Wikipedia topics thus extracted within the text of window size are considered as the contextual words to the technical topic to be learnt. These contextual words are appended as the phrases to the topic to be learnt.

3.4.1 Similarity Computation

In this section, we discuss on the mapping of n-grams generated in the above phase to the Wikipedia topics. We use string similarity in mapping of the terms which is described as follows.

Two strings are similar if they have characters in common. They are a lot more similar if they have co-occurring characters in common. Incorporating co-occurrence information helps more in finding the similarity of the strings. N-grams capture the co-occurrence. So, we form all the n-grams of all n less than the length of each string. Now, each string can be viewed as a set of n-grams of all lengths below n. We then calculate the intersection of the two sets divided by its union.

We use a well-known string similarity metric called Jaccard coefficient to compute the similarity between two topics. The topics are considered to be mapped only if the calculated coefficient is greater than 0.8. Jaccard coefficient of two strings is defined as the division between the number of characters that are common to all divided by the number of characters as shown below.

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}, \quad 0 \leq J(A, B) \leq 1 \tag{3.1}$$

Where A and B are the sets of n-gram of characters from the two strings to be compared, respectively.
Calculation of Jaccard coefficient is shown with the help of an example as follows.

Consider the calculation of similarity between the two strings "*computer science*" and "*areas of computer science*".
The string computer science is made into a set of *n-gram* of characters in it of all *n* less than or equal to the length of the string.
Now, the string "*computer science*" is represented as a set of n-grams of characters of it as { *co, om, mp, pu, ut, te, er, r , s, sc, ci, ie, en, nc, ce* }.
And the string "*areas of computer science*" is represented as { *ar, re, ea, as, s , o, of, f , c, co, om, mp, pu, ut, te, er, r , s, sc, ci, ie, en, nc, ce*}.

Intersection of the two sets A ∩ B is 15 and the union A ∪ B is 24. Jaccard Coefficient of the two strings is calculated by dividing the intersection by union of the two sets. So the Jaccard coefficient of "*computer science*" and "*areas of computer science*" is 0.625. Thus the similarity score of the above two strings is 0.625. If the calculated score is beyond the threshold, the two strings are considered to be mapped.

3.5 Retrieval Phase

In this phase, *TeKnowBase* passes the expanded query to Lucene for search. The expanded query from both the techniques, using Wikipedia category hierarchy and contextual terms from textbook is passed to the Lucene search as a phrasal query to retrieve more relevant discussions for a queried topic. The list of posts extracted from StackOverflow by Lucene search for an easy learning of a technical topic are presented to the users in the retrieval phase.

Chapter 4

Data Pre-processing

In this chapter we discuss the pre-processing of data from the resources used in the project. The data in the chapters of textbooks are to be pre-processed before the extraction of technical topics out of it. Section 3.4 discusses the pre-processing of textbook data. The Wikipedia article page titles obtained from the category links sql dump of Wikipedia are to be pre-processed for further accessing. Section 3.2 discusses the pre-processing of the tiles of Wikipedia pages. The noisy text of StackOverflow should go through the pre-processing phase for the discussion threads to be useful for further experiments. We discuss the pre-processing of StackOverflow data in section 3.3.

4.1 Pre-processing the data from textbooks:

HTML version of a textbook is taken and the HTML pages of contents and index terms are parsed to extract the technical topics out of the textbook. Jsoup parser is used to parse the HTML pages of the textbook. Jsoup is a Java library for working with real-world HTML. It provides a very convenient API for extracting and manipulating data, using the best of DOM, CSS, and jquery-like methods. Jsoup implements the WHATWG HTML specification, and parses HTML to the same DOM as modern browsers do. It is designed to deal with all varieties of HTML found in the wild; from pristine and validating, to invalid tag-soup; Jsoup will create a sensible parse tree.

The names given to html pages vary in formats for different textbooks. They contain special characters in them. The textbook we have chosen has hyphens instead of spaces in the names given to html pages. These names are used to store the textbook as a hierarchy of topics in the database. So, the names of html pages are to be pre-processed before storing into the database. The names are pre-processed by removing the hyphens and special characters before storing into the database. Pre-processing the names given to html pages is done to remove the special characters, maintaining the consistency and for the ease of further processing. The chapter names are listed in <ul><li> tags. These tags are to be parsed to extract the titles of chapters and are pre-processed before storing into the database. The sub-chapters names are also extracted by parsing the <ul><li> tags further. The index terms which are treated as the technical terms are listed with the hyper link to the corresponding chapter. These links are parsed to extract the name of the html document and is pre-processed by removing the hyphens and the special characters.

4.2 Pre-processing the titles of Wikipedia articles:

Category hierarchy of few chosen Wikipedia categories is to be built for the technical topic database . The titles of sub-categories and the pages under each category are stored in a database with their corresponding parent categories. The sql dump of category links is used in preparing this. The titles of sub-categories and pages are not in a consistent format. Sql dump of category links of Wikipedia contain the category titles with underscore in spaces and change of word in uppercase where the page titles are completely in upper case. Some of the titles also contain special characters. These are removed in the pre-processing phase for ease of querying. The underscores in each title are removed and the titles are stored in lowercase.

4.3 Pre-processing of StackOverflow data:

StackOverflow is an online discussion forum which features questions and answers on a wide range of topics in computer programming. It has the discussions on the real world applications of various topics in various fields. Several implementation details and advanced applications of technical topics are being discussed by various users. These discussions provide a rich knowledge on learning a technical topic.

The posts in StackOverflow and the corresponding discussions on them are extracted as a single thread and saved as separate html files. Each html file contains the whole discussion on a post made by the user. This corpus of StackOverflow discussions are indexed using Lucene.

4.3.1 Lucene

Lucene is a powerful full-text search library written in Java. Lucene is used to provide full-text indexing across the discussions on StackOverflow. Some of the features provided by Lucene through its API are as follows.

- Scalable, High-Performance Indexing
- Ranked searching with best results returning first.
- many powerful query types: phrase queries, wild card queries, proximity queries, range queries and more

- pluggable ranking models, including *the vector space model* and *Okapi BM25*

Supporting full-text search using Lucene requires two steps:

1. Creating a lucene index on the documents and/or database objects and
2. parsing the user query and looking up the pre-built index to answer the query.

Text search using Lucene requires two steps. They are as follows.

Indexing

We pump the data into the Index, then do searches on it to get results out. Document objects are stored in the Index, and so the data is to be converted into Document objects and store them to the Index. For this, we read in each data file (or Web document, database tuple or whatever), instantiate a Document for it, break down the data into chunks and store the chunks in the Document as Field objects (a name/value pair).

Searching

The query passed is parsed and the documents containing the keywords in the query are searched in the index. The documents containing the query terms are ranked based on a ranking model and are retrieved according to their scores.

Chapter 5

Dataset Preparation

In this project, domain specific data from textbooks, Wikipedia and StackOverflow, an online discussion forum is used. We discuss about the preparation of datasets in this chapter. The section 5.1 describes the formation of database out of the topic hierarchy made from textbooks. In section 5.2 we describe the formation of technical topic database out of the article page titles of Wikipedia. section 5.3 describes the dataset of StackOverflow

5.1 Database of topics from textbooks

In a textbook, the text under each chapter and sub-chapter can be viewed as a collection of technical terms. Generally, each of these technical terms is listed at the end of the textbook as BoBI. [17] uses BoBI and ToCs as searching and browsing tools in e-book environment. We use the BoBI and ToC of textbooks to build a hierarchy of technical topics. We make the title of the textbook at the higher level, chapter names at the next level, the sub-chapters under each chapter at further level and the terms in BoBI as the lowest level. Each index page term may fall under one or more sub-chapters. We capture all the sub-chapters under which an index term falls. Now, the whole textbook can be viewed as a hierarchy of technical topics.

Textbooks are available online either in PDF or HTML format. In the HTML version of a textbook, all the items in the list which are listed under the <ul><li> tags are captured. The technical terms are chosen to be the value of href tag in the items of the list. Since the value in the href tag is can be given to more than one technical topic, value of the href tag is not unique. Therefore names of the HTML pages of the textbook are considered for storing in the database. For the sub-chapters, the items listed in the sub-list of <ul><li> tags are further captured to maintain the hierarchy. The sub-lists of the sub-lists are captured until the end of the hierarchy. For the textbook we have chosen, the maximum levels of hierarchy has turned to be 6 including the root. The index terms in the index page of the textbook are recorded separately and the hyper-links given to each term are used to maintain the hierarchy. The index terms are stored

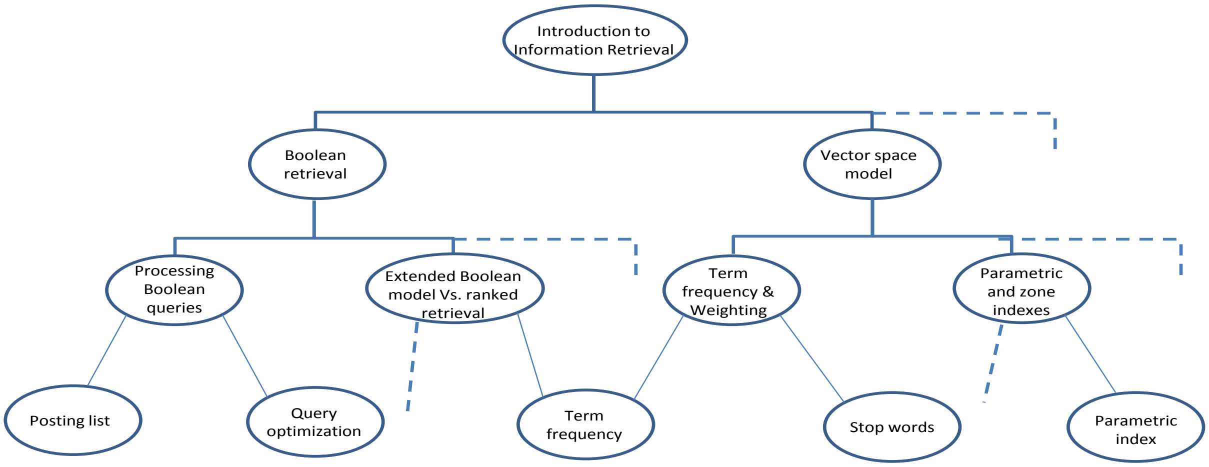
Introduction to Information Retrieval
Boolean retrieval
Vector space model
Processing Boolean queries
Extended Boolean model Vs. ranked retrieval
Term frequency & Weighting
Parametric and zone indexes
Posting list
Query optimization
Term frequency
Stop words
Parametric index

Figure 5.1: Textbook as a hierarchy of topics

in the database with the corresponding chapters the terms fall under.

If the textbook is available as a PDF format, the text in the book is parsed for the chapter names and the sections under the chapters. The page numbers of each index term are noted and the corresponding page is parsed for the chapter title. Thus the database of topics for each chapter and sub-chapter is prepared.

Textbook hierarchy of technical topics is shown in the figure 5.1.

5.2 Database of topics from Wikipedia

Wikipedia provides a well organized and structured way of categorization of various topics. It has a wide coverage of topics over technical domain on a large scale. The sibling topics of a topic under the category to which a topic belong may serve as a pivot in driving the search for more relevant discussions on the topic. For example, if the technical topic to be learnt is “Database Normalization”, the discussions on database normalization with loss-less dependency will be more relevant than the discussions on database normalization alone. Wikipedia category hierarchy is used in solving many problems such as Entity Disambiguation [18], Entity ranking [19] etc. We use the category hierarchy of Wikipedia as one of the techniques in retrieving the relevant discussions on a topic from StackOverflow data.

All the subcategories and article page titles of few selected categories are stored along with the title type. This forms a huge complex graph of category titles and article page titles. Since a

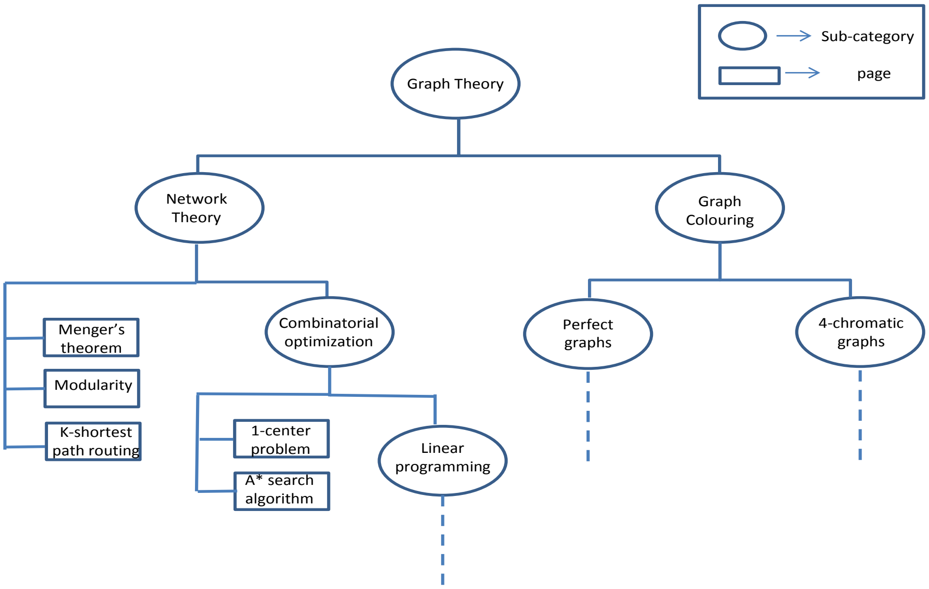

Figure 5.2: Wikipedia as a hierarchy of topics

page may fall under one or more categories which may further come under different categories. For example, the topic "Normalization" falls under the categories of "Data Modeling" and "Statistical Ratios". This makes Wikipedia to be viewed as a directed acyclic graph.

Even though the APIs of Wikipedia like MediaWiki helps in extracting the category hierarchy of Wikipedia, the setup of the API is quite demanding. So, the SQL dumps of Wikipedia are used in setting up the database with the topic hierarchy of selected categories.

The category hierarchy of Wikipedia categories is shown in the figure 5.2

5.3 Data of StackOverflow

StackOverflow is a discussion forum which features questions and answers on a wide range of topics in computer programming. Users discuss the issues in understanding of several topics [20] while implementing them or their applications. It can be viewed as an information market where people transfer goods of questions and answers and get scores as reputation in return. It has the discussions on the real world applications of various topics in a wide range of fields [21]. Several implementation details and advanced applications of technical topics are being discussed by various users tightly focused on a specific problem [22]. These discussions provide a rich knowledge on learning a technical topic by providing a reference to the problems already faced in implementing or the understanding of a specific topic by other users.

Chapter 6

Experiments

In this section we discuss the implementation details of the framework designed for *TeKnowBase*.

6.1 Dataset

The textbook of "Introduction to Information Retrieval" is used to evaluate the framework and the techniques used in it. Hierarchy of topics is prepared from the chapters of the given textbook. The index terms in the index page of the textbook are considered as the technical topics to be learnt in the context of its parent topics. The title of the textbook is considered to be at level one followed by the chapter titles at level two, further followed by the sub chapters in the later levels.

We have randomly chosen 5 index terms form the textbook as queries for assessing the techniques used in query expansion.

The categories selected for making the sub hierarchy of technical topics from Wikipedia are as follows.

- Areas of Computer Science
- Information science
- Graph Theory
- Theory of Computation
- Probability and statistics
- Mathematics

The sub hierarchy of the above mentioned areas is captured from Wikipedia category hierarchy which includes the set of pages and further sub categories. This is performed by a join operation over the category data dump of Wikipedia.

The discussions made over StackOverflow are available for download in the form of XML dump. The Posts.xml file which is 5.2GB (zipped) and 26GB(Unzipped) is downloaded and the posts are extracted as the discussion threads.

The posts in StackOverflow and the corresponding discussions on them are extracted and saved as HTML files. Each html file contains the whole discussion on a post with it. There are around 70 lakh posts in the 26GB posts.xml file. Lucene(lucene.apache.org) is used in indexing the posts of StackOverflow. The default ranking model of Lucene, *vector space model* is used in ranking the results to the query. The queries to Lucene are passed as phrasal queries and the default conjunction operator of Lucene is used between the phrasal terms in the queries.

6.2 Phases in the architecture

In the extraction phase, parent topics from textbook are extracted for the given technical topic. From the technical topic database of Wikipedia, the parent categories and the pages under the parent category of the queried topic are extracted. These titles include the sibling articles, parent categories, child categories and so on. The top 10 discussions over the queried topic are extracted from the StackOverflow data.

In the query expansion through Wikipedia category hierarchy, for each queried topic, the article titles in its hierarchy are extracted from the captured Wikipedia sub hierarchy. The actual query is expanded by appending each topic extracted from Wikipedia category hierarchy for chosen 5 queries.

In the expansion phase through contextual topic from textbook, the text within the window of 50 characters before and after the occurrence of the topic in a parent topic in the textbook is extracted. The n-grams of size three are generated out of the text and each n-gram is mapped with the technical topic in the technical topic database of Wikipedia. The terms which are mapped are considered to be the contextual words for the query. The query is expanded by appending the contextual words as phrases to the actual phrase of the topic for the chosen 5 queries.

In the retrieval phase, the expanded query from both the techniques is now passed to the Lucene search library as a phrasal query to retrieve the StackOverflow discussions on the topic. Top 10 results among those retrieved from StackOverflow are considered for further evaluation.

6.3 Defining the Relevance Levels

The discussions over StackOverflow contains various levels of detailed information. Some discussions assumes prior knowledge on the concept being discussed. Some may contain an indirect way of providing the details on the concept. Considering all these attributes of a discussion, evaluating them with binary relevance may not be good measure in effectiveness of the tool. Graded relevance is chosen to be the measure in evaluating since discussions can be rated over a scale depending on the content present in it. We define each relevance level based on the attributes a technical topic can possess and the amount of relevant information being discussed in the post.

The scale of graded relevance level is chosen to be standard 0-4. Guidelines are formulated for each level of relevance considering the attributes a technical topic can possess. These guidelines help the judges in evaluating the results upon their relevance. We divide various attributes possessed by a technical topic such as definition, applications, uses, advantages, disadvantages, implementation details, advanced application details, examples etc into different levels of relevance. The categorization of attributes of a technical topic into various relevance levels is shown below.

- Highly relevant: All the posts which discuss the definition, examples on the topic, the advantages, disadvantages of it, differences between two similar kind of topics, implementation details of the topic, etc.
- Moderately relevant: The posts which discuss bugs on the implementation details, differences between the implementations, applications of the topic, etc are considered to be in level 2.
- Less relevant: Posts which contain the advanced application details are considered to be in level 1.
- Not relevant: Posts which do not contain any useful information on the topic are non-relevant.

6.4 Evaluation and Results

Top 10 results of randomly chosen 5 queries by Lucene are captured from the three techniques; one is the direct phrasal search of the topic in the extraction phase, second is from the expanded query with Wikipedia topics and third is from the expanded query with contextual words in the query expansion phase. These results are to be manually evaluated for the relevance each result is carrying with it. The prior knowledge of a person on a particular topic may make him assign relevance level to a post on the topic rather than judging whether it is relevant or not. So, graded relevance is preferred over Boolean relevance [23]. Relevance levels are defined and assigned as discussed in section 6.3.

The definitions of the relevance levels are given to the judges in evaluating the results to each query. Each result is evaluated by three judges and the ratings to each post are recorded. Top 10 results for each query in each technique are rated by 3 judges on their level of relevance. The popular Discounted Cumulative Gain(DCG) based evaluation techniques are used to evaluate the results with graded relevance [24]. DCG assumes that highly relevant discussions are more useful then marginally relevant ones. It also assumes that the lower the ranked position of a relevant discussion, the less it is useful to refer. We calculate DCG@10 to measure the effectiveness of each technique by calculating the DCG@10 score for each set of rating [25]. There by, DCG@10 is calculated for each of the 3 manually given scores for each query as shown below. The discounted cumulative gain at rank k is calculated by the following formula.

$$DCG = r_1 + \frac{r_2}{\log 2} + \frac{r_3}{\log 3} + ... + \frac{r_k}{\log k}$$

Where $r_1, r_2, ...r_k$ are the ratings of the documents in ranked order.

This is performed on the results from each technique and are averaged upon the 3 ratings for the final score of DCG.The DCG@10 scores for the 5 queries on three techniques are shown in the following table 6.1.

Query	**Query term as a phrase**	**Query expanded with Wikipedia topics**	**Query expanded with contextual words**
B-Tree	8.56255	8.13810	11.91662
Bayesian Networks	6.65876	11.01707	12.12214
Mutual Information	10.63158	6.39435	6.74630
Natural Language Processing	7.70055	13.03972	4.29873
Un-supervised Learning	5.90629	11.055311	11.70948
Average DCG for each technique	**7.89195**	**9.92891**	**9.35866**

Table 6.1: Results: Average DCG scores for each technique

Normalizing DCG at rank n by the DCG value at rank n of an ideal ranking system gives Normalized Discounted Cumulative Gain(NDCG) score. The ideal ranking would return the discussions with highest relevance, then the next highest relevance level, etc. NDCG@10 can also be used in evaluating the system when the results of an ideal ranking system is known. Since the results of an ideal system is not known, we use the DCG@10 as the evaluation measure for the effectiveness of the three techniques.

The DCG@10 values shows the effectiveness of both the techniques of query expansion in retrieving more relevant discussions on a particular topic. However, the contextual words based query expansion showed higher improvement over Wikipedia topic based query expansion.

However, we evaluated our techniques using the normalized discounted cumulative gain by making assumptions on the relevance level of the discussions retrieved by the ideal ranking system. An ideal system would rank the discussions with highest relevance level first and the second highest relevance level next and so on. Since the relevance level of all the posts on StackOverflow data is unknown, we assume the relevance level of the ideal ranking system to be one of

Query	**Query term as a phrase**	**Query expanded with Wikipedia topics**	**Query expanded with Contextual words**
B-Tree	0.747810285	0.713270874	0.913730433
Bayesian Network	0.671412293	0.876924568	0.901558394
Mutual Information	0.833802802	0.726013224	0.650144364
Natural Language Processing	0.816811225	0.959503297	0.697770594
Un-supervised Learning	0.591230071	0.97391095	0.790089911
Average NDCG@10 for each technique	**0.732213335**	**0.849924583**	**0.790658739**

Table 6.2: Average NDCG@10 for each technique using the Ideal NDCG as the sorted relevance levels of results

the following.

- An ideal ranking system would rank the results in the sorted order of relevance with the highest relevance level first among the results obtaining by one of the techniques used [26].
- We have at least 10 posts with relevance level 3, and an ideal system would rank these 10 discussions as top-10 results for each technique [25].

With these assumptions, we calculated the NDCG@10 scores of each query for all the techniques. We used both the assumptions on the ideal ranking system in calculating the NDCG scores.

Average NDCG@10 scores of each technique with assumption 1 are listed in the following table6.2. With this assumption, the technique of query expansion using contextual words showed an improvement over the the technique with just the query term as a phrase. And the technique of query expansion using Wikipedia topics showed an improvement over both of the other techniques.

Using the assumption 2, average NDCG@10 scores of each technique are calculated and listed in the following table 6.3. The results are same as the results using assumption 1. The technique of query expansion using contextual words showed an improvement over the the technique with just the query term as a phrase. And the technique of query expansion using Wikipedia topics showed an improvement over both of the other techniques.

So, using DCG@10 as the evaluation measure, query expansion using contextual words showed the higher performance than query expansion with Wikipedia topics. Where as in NDCG@10 as the evaluation measure with both the assumptions, query expansion using Wikipedia topics showed higher performance over the technique of query expansion using contextual words.

Query	**Query term as a phrase**	**Query expanded with Wikipedia topics**	**Query expanded with Contextual words**
B-Tree	0.543189162	0.516262933	0.755964142
Bayesian Network	0.422417305	0.698898415	0.769001647
Mutual Information	0.674443583	0.405643663	0.427970685
Natural Language Processing	0.396014925	0.827211021	0.27270232
Un-supervised Learning	0.374682312	0.701324161	0.736457845
Average NDCG@10 for each Technique	**0.482149457**	**0.629868039**	**0.611143683**

Table 6.3: Average NDCG@10 for each technique using the ideal NDCG as the documents with 3 as relevance level

Chapter 7

Conclusion and Future Work

Through *TeKnowBase*, we have presented our early explorations in bridging the gap between the drawbacks of textbooks by making use of the online resources and the advantages of textbooks. We provide a summary on the technical topics to be learnt by extracting the relevant online discussions on the topic. In this project, we proposed two approaches in retrieving the most relevant discussions on a technical topic from StackOverflow based on the context to learn the topic. Both the proposed approaches of extracting relevant discussions from StackOverflow by expansion of query through Wikipedia topics and contextual terms are proved to be effective over the retrieval of the discussions containing just the topic.

Some important observations made in the project are that *TeKnowBase* can actually propose new articles that can be made in Wikipedia. It can also suggest various links that are to be made in a Wikipedia article between two technical topics, through the contextual words extracted out of the text around the topic in the textbooks. *TeKnowBase*, when extended to the other domains can contribute in overall enrichment of Wikipedia.

Bibliography

[1] Rakesh Agrawal, Sreenivas Gollapudi, Krishnaram Kenthapadi, Nitish Srivastava, and Raja Velu. Enriching textbooks through data mining. In *Proceedings of the First ACM Symposium on Computing for Development*, ACM DEV '10, pages 19:1–19:9, New York, NY, USA, 2010. ACM.

[2] Peter Hubwieser and Andreas Zendler. How teachers in different educational systems value central concepts of computer science. In *Proceedings of the 7th Workshop in Primary and Secondary Computing Education*, WiPSCE '12, pages 62–69, New York, NY, USA, 2012. ACM.

[3] Ira Diethelm, Peter Hubwieser, and Robert Klaus. Students, teachers and phenomena: Educational reconstruction for computer science education. In *Proceedings of the 12th Koli Calling International Conference on Computing Education Research*, Koli Calling '12, pages 164–173, New York, NY, USA, 2012. ACM.

[4] Radu P. Mihail, Beth Rubin, and Judy Goldsmith. Online discussions: Improving education in cs? In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education*, SIGCSE '14, pages 409–414, New York, NY, USA, 2014. ACM.

[5] Rakesh Agrawal, Sreenivas Gollapudi, Anitha Kannan, and Krishnaram Kenthapadi. Textbooks for developing regions. In *Proceedings of the First Workshop on Information and Knowledge Management for Developing Region*, IKM4DR '12, pages 1–2, New York, NY, USA, 2012. ACM.

[6] Rakesh Agrawal, Sreenivas Gollapudi, Anitha Kannan, and Krishnaram Kenthapadi. Enriching textbooks with images. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, pages 1847–1856, New York, NY, USA, 2011. ACM.

[7] Rakesh Agrawal, Maria Christoforaki, Sreenivas Gollapudi, Anitha Kannan, Krishnaram Kenthapadi, and Adith Swaminathan. Mining videos from the web for electronic textbooks. In CynthiaVera Glodeanu, Mehdi Kaytoue, and Christian Sacarea, editors, *Formal Concept Analysis*, volume 8478 of *Lecture Notes in Computer Science*, pages 219–234. Springer International Publishing, 2014.

[8] Margaret Mazzolini and Sarah Maddison. When to jump in: The role of the instructor in online discussion forums. *Comput. Educ.*, 49(2):193–213, September 2007.

[9] G. Alan Wang, Harry Jiannan Wang, Jiexun Li, and Weiguo Fan. Mining knowledge sharing processes in online discussion forums. In *Proceedings of the 2014 47th Hawaii International Conference on System Sciences*, HICSS '14, pages 3898–3907, Washington, DC, USA, 2014. IEEE Computer Society.

[10] Z. Yin, M. Shokouhi, and N. Craswell. Query expansion using external evidence. In *Proceedings of the 31st European Conference on Information Retrieval*, page to appear, Toulouse, France, 2009.

[11] Zhijun Yin, Milad Shokouhi, and Nick Craswell. Query expansion using external evidence. In *Proceedings of the 31th European Conference on IR Research on Advances in Information Retrieval*, ECIR '09, pages 362–374, Berlin, Heidelberg, 2009. Springer-Verlag.

[12] Nitish Aggarwal and Paul Buitelaar. Query expansion using wikipedia and dbpedia. 2012.

[13] Donna Harman. Relevance feedback revisited. In *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '92, pages 1–10, New York, NY, USA, 1992. ACM.

[14] Marijn Koolen, Gabriella Kazai, and Nick Craswell. Wikipedia pages as entry points for book search. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, WSDM '09, pages 44–53, New York, NY, USA, 2009. ACM.

[15] Yinghao Li, Wing Pong Robert Luk, Kei Shiu Edward Ho, and Fu Lai Korris Chung. Improving weak ad-hoc queries using wikipedia asexternal corpus. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, pages 797–798, New York, NY, USA, 2007. ACM.

[16] Yonggang Qiu and Hans-Peter Frei. Concept based query expansion. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '93, pages 160–169, New York, NY, USA, 1993. ACM.

[17] Noorhidawati Abdullah and Forbes Gibb. Using a task-based approach in evaluating the usability of bobis in an e-book environment. In *Proceedings of the IR Research, 30th European Conference on Advances in Information Retrieval*, ECIR'08, pages 246–257, Berlin, Heidelberg, 2008. Springer-Verlag.

[18] Saurabh S. Kataria, Krishnan S. Kumar, Rajeev R. Rastogi, Prithviraj Sen, and Srinivasan H. Sengamedu. Entity disambiguation with hierarchical topic models. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pages 1037–1045, New York, NY, USA, 2011. ACM.

[19] Rianne Kaptein, Pavel Serdyukov, Arjen De Vries, and Jaap Kamps. Entity ranking using wikipedia as a pivot. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, pages 69–78, New York, NY, USA, 2010. ACM.

[20] Dennis Schenk and Mircea Lungu. Geo-locating the knowledge transfer in stackoverflow. In *Proceedings of the 2013 International Workshop on Social Software Engineering*, SSE 2013, pages 21–24, New York, NY, USA, 2013. ACM.

[21] Bogdan Vasilescu, Alexander Serebrenik, Prem Devanbu, and Vladimir Filkov. How social q&#38;a sites are changing knowledge sharing in open source software communities. In *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work &#38; Social Computing*, CSCW '14, pages 342–354, New York, NY, USA, 2014. ACM.

[22] Patrick Morrison and Emerson Murphy-Hill. Is programming knowledge related to age? an exploration of stack overflow. In *Proceedings of the 10th Working Conference on Mining Software Repositories*, MSR '13, pages 69–72, Piscataway, NJ, USA, 2013. IEEE Press.

[23] Jaana Kekäläinen. Binary and graded relevance in ir evaluations-comparison of the effects on ranking of ir systems. *Inf. Process. Manage.*, 41(5):1019–1033, September 2005.

[24] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, October 2002.

[25] Yining Wang, Liwei Wang, Yuanzhi Li, Di He, Tie-Yan Liu, and Wei Chen. A theoretical analysis of ndcg type ranking measures. *CoRR*, abs/1304.6480, 2013.

[26] Wikipedia. Ideal NDCG with sorted relevance levels. `http://en.wikipedia.org/wiki/Discounted_cumulative_gain#Normalized_DCG`.