IMPROVING EDUCATIONAL SEARCH AND QUESTION ANSWERING

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF
MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

By
Tolga Yılmaz
June 2016

IMPROVING EDUCATIONAL SEARCH AND QUESTION ANSWERING
By Tolga Yılmaz
June 2016

We certify that we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Özgür Ulusoy(Advisor)

Fazlı Can

Rıfat Özcan

Approved for the Graduate School of Engineering and Science:

Levent Onural
Director of the Graduate School

ABSTRACT

IMPROVING EDUCATIONAL SEARCH AND QUESTION ANSWERING

Tolga Yılmaz
M.S. in Computer Engineering
Advisor: Özgür Ulusoy
June 2016

Students use general web search engines (GSEs) as their primary source of research while trying to find answers to school related questions. Although GSEs are highly relevant for the general population, they may return results that are out of education context. Another rising trend; social community question answering websites (CQ&A) are the secondary choice for students who try to get answers from other peers online. We focus on discovering possible improvements on educational search by leveraging both of the two information sources.

The first part of our work involves Q&A websites. In order to gain contextual and behavioral insights, we extract the content of a commonly used educational Q&A website with a scraper we implement. We analyze the content in terms of user behavior and try to understand to what extent the educational Q&A differs from the general purpose Q&A.

In the second part, we implement a classifier for educational questions. This classifier is built by an ensemble method that employs several regular learning algorithms and retrieval based ones that utilize external resources. We also build a query expander to facilitate classification. We further improve the classification using search engine results.

In the third part, in order to find out whether search engine ranking can be improved in the education domain using the classification model, we collect and label a set of query results retrieved from a GSE. We propose five ad-hoc methods to improve search ranking based on the idea that the query-document category relation is an indicator of relevance. We evaluate these methods on various query sets and show that some of the methods significantly improve the rankings in the education domain.

In the last part, we focus on educational spell checking. In educational search systems, it is common for users to make spelling mistakes. Actual query logs of two commercial search engines in the education domain are analyzed in terms of spelling mistakes using 5 well-known spell correction software that are not education specific and lack the terms that are used in the education field. It is shown that by extending the spell-check dictionary of one of them, even with a small-sized education oriented word-list, one can improve the precision, recall and F1 values of a spell-checker.

Keywords: Education, Classification, Search Engine Ranking, Spell Checkers, Social Q&A.

ÖZET

EĞİTSEL ARAMA VE SORU CEVAPLANDIRMANIN GELİŞTİRİLMESİ

Tolga Yılmaz
Bilgisayar Mühendisliği, Yüksek Lisans
Tez Danışmanı: Özgür Ulusoy
Haziran 2016

Öğrenciler, okulla ilgili soruları için ilk tercih olarak arama motorlarını kullanırlar. Arama motorları, her ne kadar genel popülasyon için oldukça kullanışlı olsa da eğitim kurgusunun dışında yanıtlar getirebilir. Bir başka eğilim, sosyal ağ soru-cevap web siteleri ise emsallerinden cevaplar almak isteyen öğrenciler için ikinci bir seçenek olarak karşımıza çıkmaktadır. Çalışmamızda, bu iki bilgi kaynağının birbirlerinden faydalanılarak geliştirilmesi üzerinde durulmuştur.

Çalışmamızın ilk kısmı soru-cevap web siteleri ile ilgilidir. Eğitsel soru-cevap web siteleri üzerinde bağlamsal ve davranışsal anlayışa sahip olmak için bir soru-cevap web sitesinin içeriği toplanmıştır. Bu içerik, kullanıcı davranışları ve eğitsel soru-cevap sitelerinin genel soru-cevap sitelerinden ne derece farklı olduğunu anlamak açısından analiz edilmiştir.

İkinci kısımda, eğitsel sorular için bir sınıflandırıcı geliştirilmiştir. Bu sınıflandırıcı makine öğrenmesi tabanlı bir kaç algoritma ile dış kaynaklar üzerinde oluşturulmuş bir kaç arama tabanlı sınıflandırıcıdan oluşan bir "*ensemble*" sınıflandırıcıdır. Ayrıca, sınıflandırmayı güçlendirmek için bir sorgu genişletme yöntemi geliştirilip kullanılmıştır. Oluşan bu sınıflandırıcı, son olarak arama motoru sonuç sayfaları da kullanılarak daha da geliştirilmiştir.

Üçüncü kısımda, eğitsel arama motoru sıralamasının sınıflandırma modeli kullanılarak geliştirilebilirliğini test etmek için, bir arama motorundan alınan sonuç sayfaları toplanıp etiketlenmiştir. Sorgu-doküman sınıf ilişkisinin ilgi düzeyi ile alakalı olduğu varsayımından yola çıkarak, arama motoru sıralamasını geliştirmek üzere beş yöntem kullanılmıştır. Bu yöntemler, çeşitli sorgu setleri üzerinde uygulanıp eğitsel sıralama bağlamında kayda değer gelişme olduğu görülmüştür.

Son olarak, eğitsel yazım denetimi üzerinde durulmuştur. Eğitsel arama sistemlerinde, kullanıcıların yazım hataları yapması sık rastlanan bir durumdur. İlk olarak iki ticari arama motorunun sorgu kayıtları, eğitsel amaçlı tasarlanmamış ve eğitsel kelimeleri içermeyen fakat genel olarak iyi bilinen beş sorgu denetimi ve düzeltmesi yazılımı kullanılarak denetlenmiştir. Bu yazılımlardan bir tanesinin sözlüğünün, küçük boyutlu bir eğitsel kelime listesi ile bile desteklendiğinde *"precision"*, *"recall"* ve F1 değerlerinin gelişme gösterdiği görülmüştür.

Anahtar sözcükler: Eğitim, Sınıflandırma, Arama Motoru Sıralaması, Yazım Denetleyiciler, Sosyal Soru-Cevap.

Acknowledgement

Foremost, I would like to express my sincere gratitude to my advisor Prof. Dr. Özgür Ulusoy for his guidance and patience throughout my M.S. study. This thesis was made possible with his continuous help.

I would also like to thank Asst. Prof. Dr. İsmail Sengör Altıngövde and Asst. Prof. Dr. Rıfat Özcan for their comments, insights and corrections in our meetings.

I would like to thank Prof. Dr. Fazlı Can for kindly agreeing to be in my jury.

I would like to mention that this work was supported by TÜBİTAK under the grant number 113E065, between 2013 and 2015.

Finally, I would like to express my feelings to my family for being the very definition of support. Through their love and patience, I have come this far.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Students choose the Internet as their primary resource for research when it comes to finishing a homework or learning a new subject in their curriculum whether the subject at hand is a Mathematics problem or an open-ended Social Sciences project. Web search engines provide the ability to scan a variety of web pages and provide a list of possible candidates for the student to choose from. It is up to the student then, to extract the information available in the returned web pages. However, most of the available web search engines are designed for general purpose and may not serve the students' needs that are very specific to the education context. For example, the web pages returned to the queries can be from other topics due to a diversification process employed by the search engine or just because textual similarity. Consider, for instance, a student performs a query on a general search engine (GSE): "How do mirages occur?" ("Serap nasıl oluşur" in Turkish) hoping to get answers in a scientific context. Instead, the GSE returns a website talking about a celebrity person named Mirage (Serap in Turkish).

Other than being out of context, the results returned by a GSE can be for users from other levels. For instance, let's say a K-12 student performs a query on a GSE: "What are the three organs of the state?" and the GSE returns a highly technical text for law school students, which would, of course, be harder

to use for the student. Another problem with GSEs is that the answers to the questions are from various parts of the Internet which may not be appropriate at all in the education context. For example, a car parts lovers chat website or forum may have a section that contains the answers to the questions above. The answers to the questions are also highly duplicated among many websites and the student may spend more time to find a well-written original answer.

There are efforts to create more education-focused search engines such as ISEEK [1] which has a database containing a list of editor-reviewed websites the user can search on. The content is labeled under a taxonomy according to level, category, topic and date in order to further allow the users to narrow their search. It is also a very hard task to continue adding more documents since human editing consumes a considerable amount of time and the knowledge in the education field is too large to be handled by the mere human review. Intute [2] was another example of a similar effort. It has been shut down since 2011. There are of course academic search engines such as Google Scholar, Microsoft Academic Search, CiteSeerx, and CiteULike, but these are for the users with higher education.

In Turkey, as part of providing interactive online course content, SEBIT [3] company provides paid services through online Vitamin Eğitim [4] platform for registered students which include a search tool built on their course material. On the other hand, Eğitim.com [5], a subsidiary of SEBIT, provides free of charge search service for education related web pages. Usta et al. analyzed the search log of Vitamin in [6] and provided a Learning to Rank method in [7] for improving Vitamin search engine ranking based on various features such as content, title, description, dwell time, and clicks.

The Turkish government has also been investing into e-education. As part of the Fatih project [8], Turkish Ministry of Education created the Education Information Technology Network [9] in order to create reliable e-material for education to be targeted at students, teachers and parents. They incorporated the previously stated Eğitim.com as a search service in one of their subdomains

[10]. This way, the service has been officially recommended for use to 17.5 million students [11] enrolled in the formal education (K-12) in the country.

Student behavior, besides web search, includes asking for help in online communities. These websites in the last decade evolved from forums with continued discussion to question and answer (Q&A) websites which are more focused on the quality of questions and answers. Although forums are still popular as discussion platforms, it is easier to extract information from structured community Q&A websites. Additionally, while search engines list relevant web pages, one needs to scan these pages and extract the answer from them. In Q&A websites, there is direct access to answers. Many-topic Q&A websites dominate the World Wide Web today such as Quora [12], Yahoo! Answers [13] and StackExchange [14]. There are many small scale homework sites and also subsections under popular websites such as Yahoo! Answers Education & Reference Category [15].

Turkish student community also populates educational Q&A websites. Brainly.co [16] operates an educational Q&A platform in 35 countries including Poland, Russia, Brazil, the U.S.A., Spain, France and Turkey. They report 60 million monthly unique users across their websites. Turkish one is named as EOdev.com [17] and it claims to have over 4.5 Million questions.

Our work focuses on these two online main information sources of students and their combination, in order to improve the topicality of search engines as we state this as one of the shortages of GSEs. In this respect, we go to the source of educational questions and analyze an educational Q&A website in order to show the differences between this type and general purpose Q&A websites, if any.

Secondly, we implement and evaluate a classifier specific for educational questions collected from the Q&A website. We employ various data sources that we either directly obtained or compiled. These include online course material, textbooks, a large educational term collection we compiled and search engine results and manually labeled questions. We build retrieval based classifiers on top of the course material, the textbooks and the term collection. We also use various machine learning classifiers such as Naive Bayes, its variations, and Support Vector

Machines. We first combine the retrieval and machine learning based classifiers using an ensemble method. We improve the classification with a query expansion technique. Then we employ a voting method using search engine results to further improve the accuracy of the ensemble.

Using this classifier, we show that by employing simple techniques we can re-rank search results to increase relevance. We first generate a relevance estimation based on classification and re-rank results based on this estimation. The first class of our techniques includes the identification of result pages that are possibly from other subject and demoting them. Other class of our techniques uses a combination of the initial relevance and the classification based relevance.

Finally, we implement and evaluate an education-aware spell checker. We show that by enhancing its dictionary with educational words, it is possible to build a spell checker better than state-of-the-art but context-unaware spell checkers. We test and evaluate our techniques on real-world query logs.

The thesis is structured as follows. In Chapter 2, we give a brief review of the literature related to our work. Chapter 3 includes the analysis of an educational Q&A website. Chapter 4 presents our educational question classification method. Chapter 5 explains the use case of our classifier in ranking results of educational queries. In Chapter 6, we give the implementation details of the educational spell checker. Chapter 7 concludes the thesis and gives future research directions.

Chapter 2

Related Work

In this chapter, we briefly discuss the literature related to Question Answering (Q&A) systems, educational query classification, and spell checking systems.

2.1 Social Question Answering

A Social Q&A website is defined as the place where questions in natural language form, rather than in the form of keywords (i.e., in the form of search engine queries), are asked by regular users to find answers within the community. These websites have gained world-wide popularity as a new way to access information for people who seek answers to their questions. Yahoo! Answers website was visited by 46 million single users in May 2015 [13] and StackOverflow.com, an online Q&A community for programming questions, has over 4 million users [18].

According to Gazan [19], research on Social Q&A can be divided into two categories as question quality/classification and answer quality/classification. Example works include research on finding better users in the community [20, 21], identifying good answers [22], finding the intent behind the questions [23], and classifying questions based on type [24] (i.e., is it opinion asking or fact seeking) or based on subject [25].

Wang et al. [26] analyze the user behavior and question topics using user, user-topic and related question graphs on Quora. Mamykina et al. [27] explore the content of StackOverflow and list the details such as the percentage of answered questions, the first and accepted answer times for questions, changing answer time, distribution of answerers and askers, user activities, classification of users based on answer types.

Barua et al. [28] try to find the most common subjects of the questions using LDA analysis. They also try to find out whether questions trigger one another, and whether the user attention changes over time or how an interest in a topic changes over time. Correa et al. [29] predict which questions will be closed, i.e., correctly answered. Another work on StackOverflow by Movshovitz-Attias et al. [30] analyze the record of a user to predict her future reputation. Harper et al. [22] work on different types of Q&A websites and try to evaluate and compare their answer quality. The findings include that better answers come with more links and length. Question subject and type (factual or not) also affect answer quality. Expert identification papers are also common. Yang et al. [20] analyze the questions and answers to identify two types of experts: focused and highly active. Finally, Liu et al. [31] try to predict the popularity of questions based on their current characteristics and the askers who own them.

In the education field, Gosh et al. [32] focus on K-12 forum sites as opposed to Q&A sites and try to find what attracts users to contribute. They also point out the differences between forums and Q&A sites under the assumption that forums are assisted by actual instructors. Mao [33] focuses on the attitudes of high school students for social media. Their data show that for social media to be an effective learning environment, extraction of current social media habits of students, adoption of these into such learning environments, and interacting with students are necessary. Our work tries to identify student patterns in educational Q&A and help to find ways to improve the overall experience in future work.

2.2 Using External Resources for Question Answering

There have been efforts combining information retrieval techniques such as search to the question answering problem. In order to integrate search engines to social networks, Hecht et al. [34] developed a system that produces algorithmic answers to Facebook status questions. According to Evans et al. [35], there are three types of question asking: questions targeting another user, public asking and searching through a knowledge base; and the best way for finding good answers is to combine them. In [34], search engines are brought into the social network. Another system on this subject is Googles Confucius system [36] which was brought live in 2009 and is currently inactive. The aim of the system was to link Google to a Q&A website, and the system had parts such as question labeling, question recommendation, and NLP-based answer generation.

Komiya et al. [37] worked on a question answering scheme that performs query expansion and candidate answer evaluation by using the vocabulary of a Q&A website. Traditionally, question answering can be split into four parts: question analysis, relevant document access, candidate answer generation, and evaluation. A similar work to Komiyas was done by Mori et al. [38].

In the educational domain, Gurevych et al. [39] describe how to use social media for educational question answering. They use a classifier for the subjectivity of questions. They also present a Q&A system that fetches answers from social media content.

2.3 Text and Query Classification

A summary of machine learning techniques in text classification is given by Sebastiani [40]. Starting from Naive Bayes Classification to Support Vector Machines (SVMs), many methods have been proven to be useful for many text classification

tasks. A considerable amount of work has also been conducted on query classification. Shen et al. [41] describe their ACM KDDCUP 2005 winning method. They build a synonym-based classifier and an SVM classifier and use two ensemble strategies based on these two classifiers using bag-of-words model. Gabrilovich et al. [42] present a real-time method for classifying queries using the web as a resource. They first develop a document classifier for query results based on a centroid-based algorithm and the bag-of-words model. Then, they use a voting algorithm that decides on the category of the query by incorporating the classification information of query results obtained by using the document classifier. Cao et al. [43] present a new, context and intent-aware query classification method. They use conditional random field (CRF) models and incorporate similar queries and click information as features in addition to query terms. Agrawal et al. [44] discuss transforming the query classification problem into an information retrieval task.

2.4 Question Classification

Question classification is an important part of question answering systems and lately, CQ&A websites and there is a considerable amount of work in the last decade. Question classification generally serves as an intermediate step towards achieving better results in other systems such as obtaining better answers by channeling questions to answerers better or simply for generating them better in the case of automatic answering systems.

Earliest question classification systems used hand written rules [45]. Although these rules could work quite fast, the problem was their creation. It would be hard by mere human intelligence to cover every possible aspect of question-category relationships and write them for growing number of categories. These rules needed to be rewritten for every new dataset. As a result, later works employ machine learning methods. For example, Zhang and Lee [46] use machine learning techniques to create more robust classification techniques. They try multiple learning

algorithms that take n-grams as features and report SVM to be the most accurate. Li and Roth [47] use semantic and syntactic features with SnoW algorithm to classify questions. Metzler and Roth [48] similarly work on semantic and syntactic features and on different data sets composed of fact-based questions. They show question classification using machine learning, SVM in their case, is robust compared to rule-based classifiers which require an immense amount of effort.

Huang et al. [49] introduce headwords and their hypernyms as an important feature of question classification. Most of the later works include this feature as well [50, 51, 52]. Most of the recent features and a broad survey are given by Loni [53].

There are works in classifying educational questions as well. For instance, Vlasák [54] classifies educational web pages according to educational subjects in Czech using SVM and n-grams. Li et al. [55] classify questions in an epistemic game that tries to teach various concepts to students with instructor involvement. Sangodiah et al. [56] gives a short review of the existing work on educational question classification featuring Bloom's taxonomy [57] which is sorting educational objectives into hierarchical levels of complexity and mastery. Research on this field includes work that employs algorithms such as SVM [58], rule-based classifiers [59] and neural networks [60].

Figueroa and Neumann's work [61] bases itself on question-like search queries from Yahoo! Answers. They try to motivate the connection between search engines and CQ&A like we do. They utilize many features and external data sources. In these ways, their paper shares common ground with ours. In Table 2.1 we show some of the works mentioned above with their features and algorithms and the method we use in this work. Our work, compared to the most of the other work, utilizes external resources such as books, term collections and search engine result pages. We also use an ensemble classifier rather than using a single algorithm. We base our work on the education context whereas most question classification research is based on general categories.

Table 2.1: The List of Works on Question Classification by Features and Algorithms

Author	Features	Algorithms	Year
Hermjakob [45]	N/A	Parse Tree	2001
Li and Roth [47]	words, POS tags, chunks, named entities, head chunks, semantically related words	SnoW	2002
Zhang and Lee [46]	n-grams	NB, Winnow, Decision Tree, SVM	2003
Metzler and Roth [48]	Bag-of-words, n-grams, POS tags, question word, noun phrases, question length, long distance k-grams,head word, hypernyms of the head word using (WordNet)	SVM	2004
Li and Roth [62]	Bag-of-words, POS, Head, Named Entities, WordNet, Class-specific related words, Distributional similarity of words	Winnow	2006
Huang et al. [49]	Wh word, Headword, WordNet, Direct-Indirect Hypernyms, Unigrams, Wordshapes	SVM, Maximum Entropy	2008
Silva et al.[50]	Unigrams, Headwords	Rule Based, SVM	2011
Loni [51]	Unigrams, Bigrams, Headwords, Headrules, Word shapes, Wh words, Hypernyms, Query expansion, Question Category	NN, SVM, LSA	2011
Mishra et al.[52]	Unigrams, Wordshapes, headword, question patterns, POS tags, hypernyms, question category	KNN, Naive Bayes, SVM	2013
Li et al. [55]	POS Tags, Wh Words, Categorical Wordlists	Decision Tree	2014
Vlasák [54]	Bag-of-words	KNN, Naive Bayes, SVM	2015
Figueroa and Neumann[61]	Bag-of-words, Latent-topic models, Acronyms, String analysis, String distances, POS tags, Yago2s, Wikipedia, Yahoo Categories	Maximum Entropy, Winnow	2016
Our work	Unigrams, Bigrams, Question length, Wh words, word shapes, POS tags, Tagged Unigram, Word dependencies, Object and Subject, Object and Subject Phrases, Hypernyms, Hyponyms and other semantic relations, Query Expansion, Search Engine Results	Ensemble of SVM (Linear RB), Complement Naive Bayes, Naive Bayes, Discriminative Naive Bayes, Hyper Pipes and search based classifiers built on multiple data sources	

2.5 Spell Checkers

A comprehensive but fairly old survey of spelling detection and correction is presented by Kukich in [63]. We also revisit and elaborate the most common techniques in Chapter 6. In this section, we focus on the context-awareness of spell checkers.

Context-aware spelling correction is generally taken as a word sense disambiguation mechanism based on the context of the input. For instance, a word may be perfectly typed but in the context of its sentence, it is mistyped. Consider; *It is to cold in the winter*. Here, *to* should have been *too* based on the content. This is an example of a spelling mistake caused by a valid word that is in the dictionary. The studies on this topic include techniques such as Bayesian classifiers [64], Winnow classifiers [65, 66], statistics based [67, 68] and Latent Semantic Analysis [69]. Features of this task are generally n-grams that help identify the context. Then, corrections are suggested based on that context. Based on training sets, the probabilities of words being consecutive to each other are calculated and words that do not conform are selected as candidates for correction.

There is another type of context awareness issue where a perfectly valid word is identified as a mistake. This is due to the misidentified word not being in the dictionary of the spell checker. Context awareness of a spell checker in this sense is to be aware of the particular subject on which it operates. The vocabulary of today's people changes rapidly with new technologies, trends and spell checkers may fall behind this development. There is a U.S. patent on constantly updating the spell checker dictionaries with user inputs [70]. Our work is also more related to this type of awareness rather than disambiguation. In this sense, we try to create an education aware spell checker to overcome the problem of general spell checkers being education unaware.

Chapter 3

An Analysis of an Educational Q&A Website

3.1 Introduction

Community Q&A websites have evolved from forums with continued discussion to knowledge bases that focus on the quality of questions and answers. Although forums are still popular as discussion platforms, it is easier to extract information from structured community Q&A websites. Platforms such as Quora, Yahoo! Answers and StackExchange have gained worldwide popularity over the last decade, attracting more users every day.

K-12 students also engage in Q&A communities seeking help for their assignments. Brainly.co operates an educational Q&A platform in 35 countries including Poland, Russia, Brazil, USA, France and Turkey [16]. They report 60 million monthly unique users. Turkish one, EOdev.com, claims to have over 4.5 million questions [17].

In the social environment of community Q&A, students can find moderated content as opposed to scanning through the results of web search where they

can come across inappropriate or wrongly leveled content. They can also see good answers promoted by other students. Q&A websites, by nature, motivate asking better questions thus teaching students to form more precise questions. By answering the questions of others, they also exercise learning by teaching and even altruism although there are other motives such as increasing reputation or receiving badges among the community.

There is a lot of research analyzing both general purpose Q&A websites [26] and domain specific ones such as programming [27], health [71, 72] and construction [72]. However, to the best of our knowledge, an analysis of a Q&A website which targets K-12 students does not exist. In order to understand how it differs from other Q&A websites, we analyze the content of an educational Q&A website.

3.2 Data Set

We have collected the publicly accessible data of `http://msxlabs.com/okul` which serves as a free to use Q&A website to support students in the Turkish K-12 system. The website has an interface similar to StackOverflow. Figure 3.1 shows a snapshot of the user interface at the time we collected the website data.

In the website, users are able to ask questions, answer them, comment on questions and answers. Updating is also possible. Users are able to upvote and downvote an answer based on its content. Highly voted answers are displayed higher in the stack. The asker is also able to select one of the answers as the accepted answer which is displayed with a "tick" mark. An example screenshot covering basic Q&A capability is given in Figure 3.2 and basic statistics about the data we collected from the website is given in Table 3.1.

Figure 3.1: Homepage of the Educational Q&A website

Table 3.1: Overview of Msxlabs, an Educational Q&A Website

Questions	10,717
Answers	19,691
Question comments	846
Answer comments	3,623
Registered users	2,044
Registered users with some activity	963
User who are not registered but did some activity	1,141

3.3 Analysis

We analyze different aspects of our dataset and compare our findings with the literature in the following paragraphs.

Figure 3.2: Inside a Question

3.3.1 Activity

We define user activity as one of the following: asking a question, answering a question, commenting on a question, commenting on an answer and updating. In Figure 3.3, the Y axis is the number of users doing X activities. The average number of activities for a user is 12.4 and median is 1. Indeed, in [27] the same skewness is observed. We did not show asking, answering and commenting separately because they were quite similar to this.

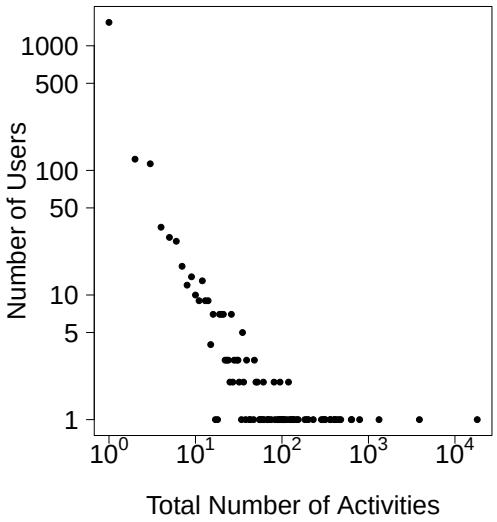


Figure 3.3: Activity Quantity by Users

3.3.2 Activity Period

The activity period of a user is defined as the average time between her activities. Note that to calculate this, a user must have issued at least 2 activities. In Figure 3.4a, we see the activity level of 691 users. In minutes, the median time between users' first and last activities is 3 days whereas the median activity period is 7.7 hours. Figure 3.4b shows that users are most active earlier in their memberships, meaning they lose interest over some time. In Figure 3.4b, we calculate Pearson correlation as 0.53. However, due to skewness, calculating Spearman correlation yields 0.92, which means that there is a strong correlation between membership duration and activity period.

3.3.3 Popularity

A signal to point popularity is the view count of questions as Figure 3.5a shows. The distribution is similar to that of StackOverflow (mean 255, median 35). In the website, 94% of all the questions are answered which is similar to the rate of 92.6% obtained in StackOverflow [27] and 88.2% in Yahoo! Answers [24]. Answer rate is an indicator to bring future users to Q&A websites. The number

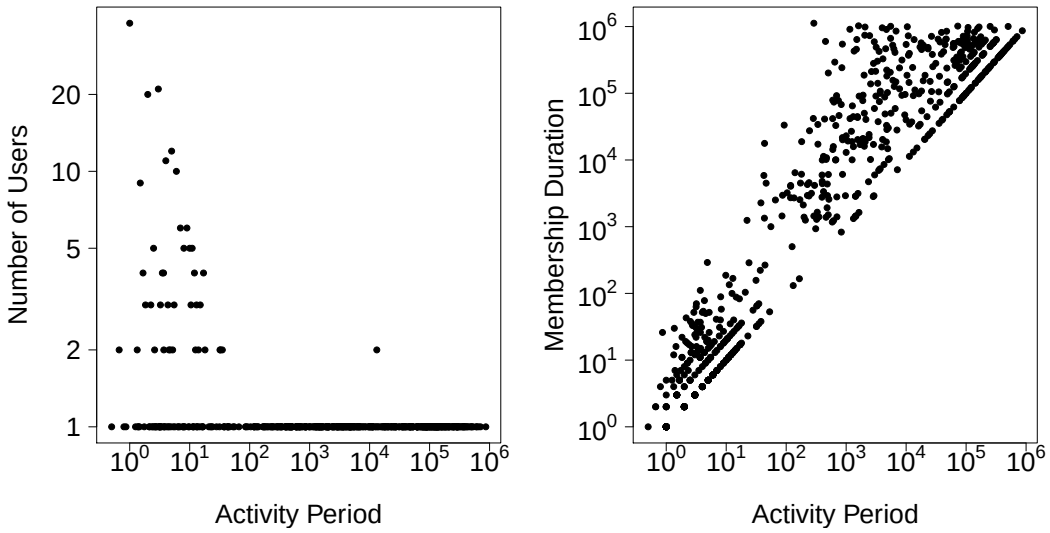


Figure 3.4: Activity Time by Users (minutes, log-scale)

of times a question is answered is an important factor identifying popular and important questions. In Figure 3.5b, the mean and median number of answers per question is 1.95 and 1, respectively. This pattern also exists in Yahoo! Answers [31]. Figure 3.5c shows that the number of cumulative votes (upvotes-downvotes) for answers is symmetrically aligned at x = 0 with fewer answers having more positive or negative outlooks. The mean cumulative vote for accepted answers is 0.25 whereas it is 0.03 for non-accepted answers. Finally, 38% of questions have accepted answers.

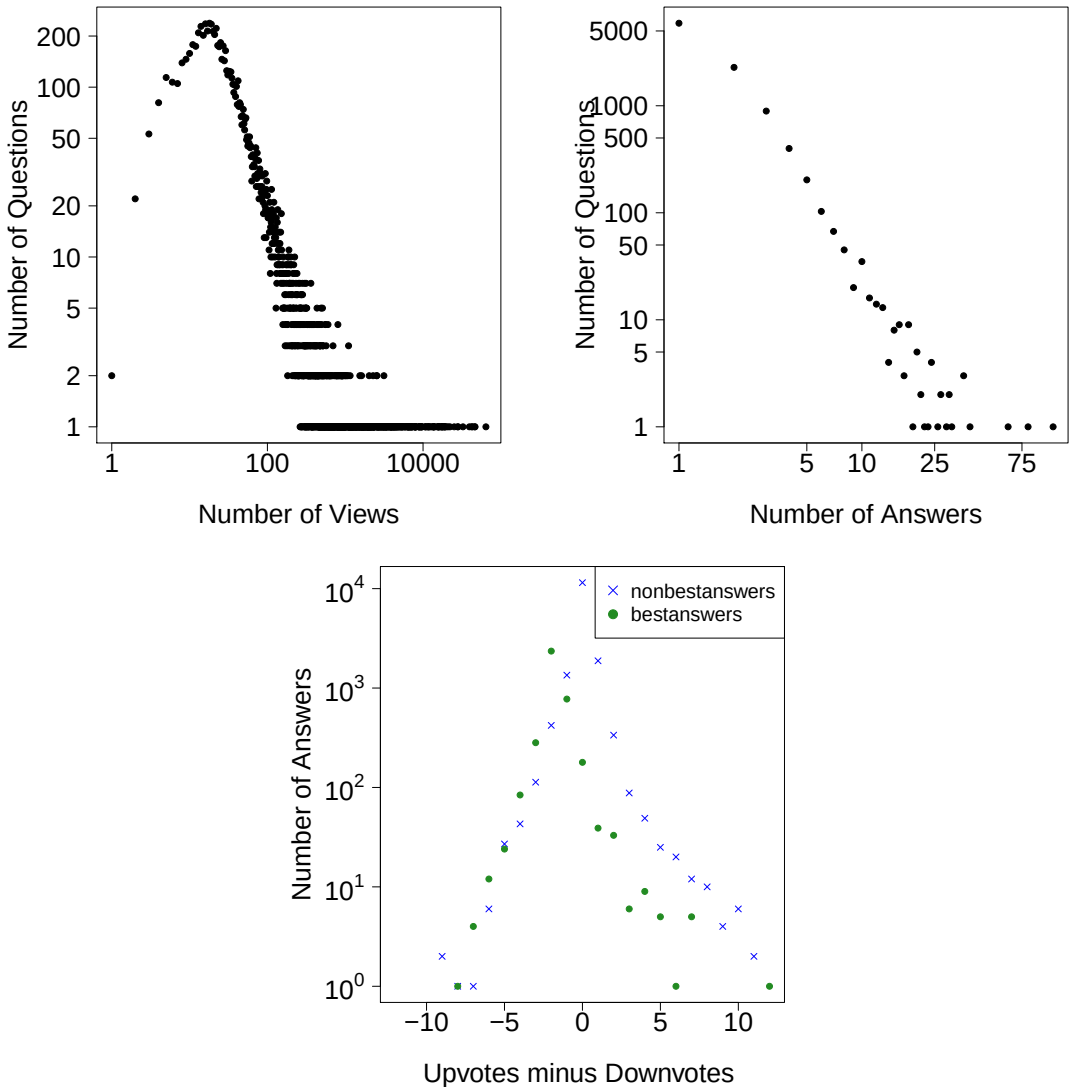


Figure 3.5: Various Popularity Indicators

3.3.4 When are questions answered?

The distribution of the time difference between the publish and earliest answer times is given in Figure 3.6a in minutes, following a power law. The mean here is 34268 minutes but the median is 225 minutes (i.e., 3.75 hours). Figure 3.6b shows the accepted answer times for questions. The median accepted answer time is 94 minutes. Figure 3.6c shows that fewer questions receive answers later, i.e., the answers are received closer to the question publish time.

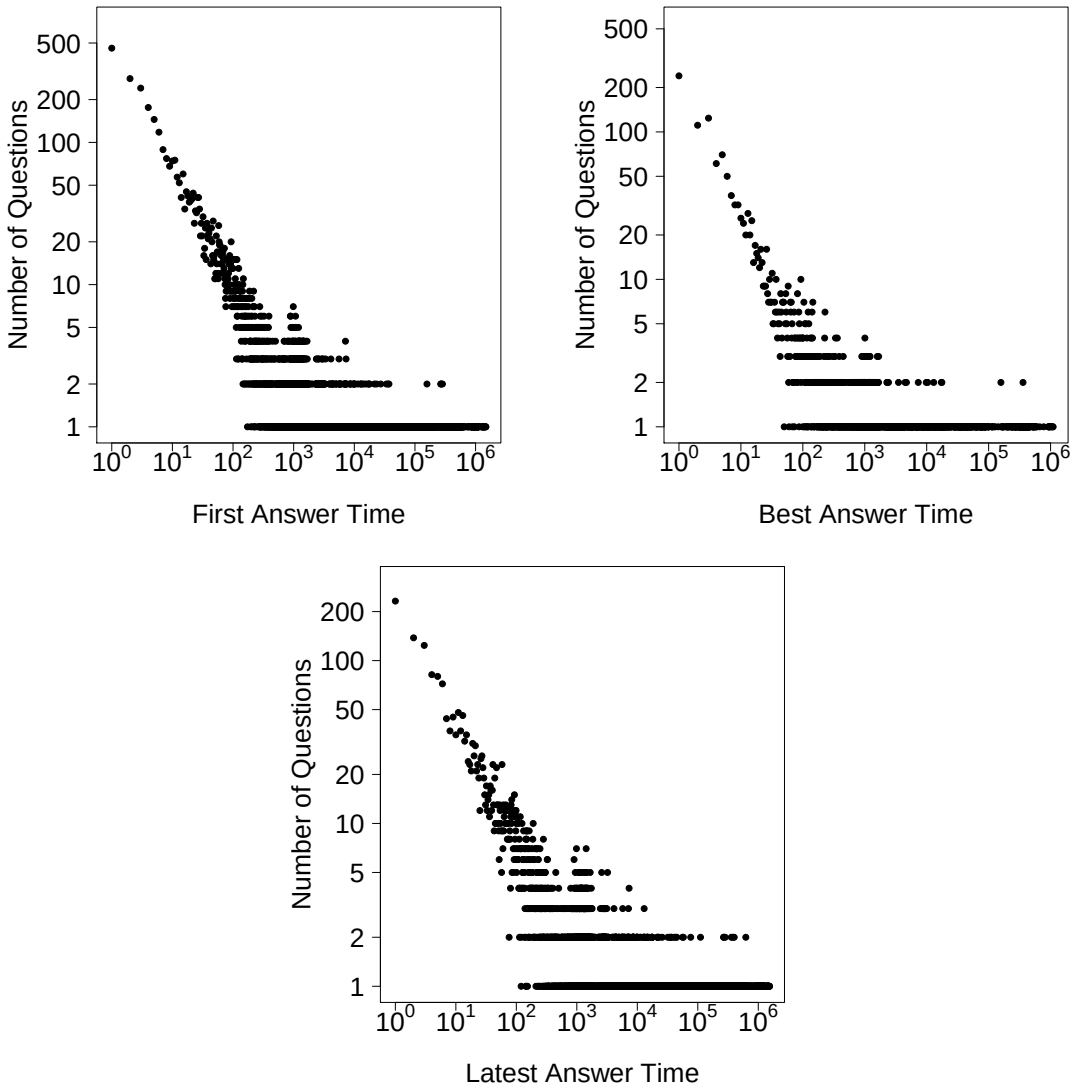


Figure 3.6: First, Accepted and Latest Answer Times (minutes)

We also analyze the daily and weekly patterns of question answering in Figure 3.7. We were expecting more activity during the weekends but observed a similar pattern again. In Yahoo! Answers and other general answering services, user activity is also reported lower during the weekends in [73] and [74], respectively. Hourly activity, on the other hand, seems to be very low between 0 and 6 a.m., confirming the expected student behavior. This is different compared to other Q&A websites where there is low but steady contribution during that time.

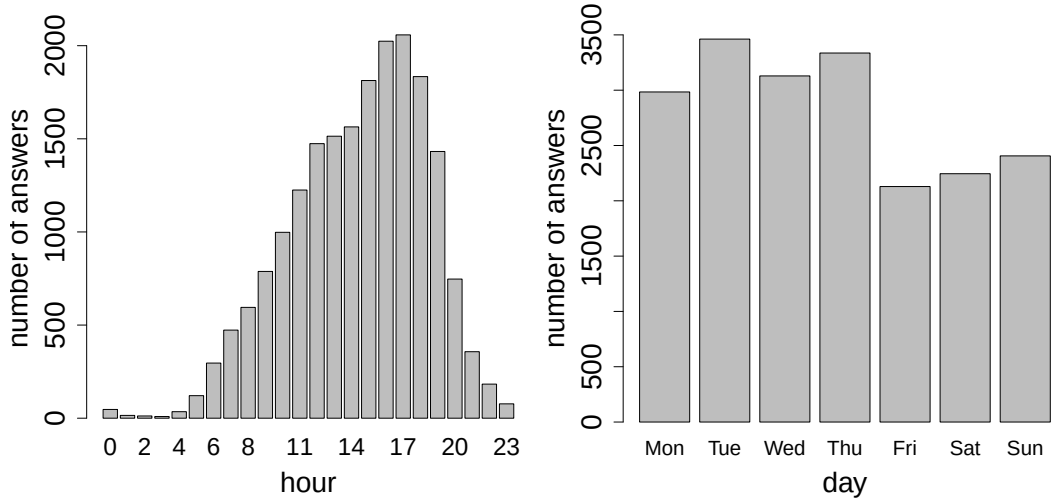


Figure 3.7: Hourly and Weekly Answering Patterns

Table 3.2: Q&A Statistics for each Subject

	Soc. Sci.	Misc.	Religion	Turkish	English	Math.	Sci.	Rep. Hist.
Median first answer time (all)*	16.7 hrs	14 hrs	9.3 hrs	11.4 hrs	1.2 hrs	7 hrs	9.5 hrs	17 hrs
Median first answer time	8.9 hrs	1 hrs	2.6 hrs	1.9 hrs	1 min	4.4 hrs	5.4 hrs	5.4 hrs
Median accepted answer time	16.5 hrs	2 hrs	5.8 hrs	3.6 hrs	1 min	10.4 hrs	15.5 hrs	13.8 hrs
Avg., Median views	484, 43	128, 31	192, 36	236, 42	144, 30	313, 45	272, 36	230, 45
Avg., Median answers	2, 1	1.8, 1	1.5, 1	1.8, 1	1.9, 2	2.4, 1	1.9, 1	1.9, 1
Questions with accepted answers	25.4%	29%	32.4%	37%	48%	38.8%	35%	31.8%
Answered questions	88.9%	91.3%	93.5%	91.3%	91.8%	94.8%	92.1%	88%

(*:contains the questions w/o accepted answers)

3.3.5 Do Subjects Matter?

We labeled 5000 randomly selected questions based on their educational subjects, resulting in: Social Sciences (29%), Religion (4%), Turkish (10%), English (3%), Mathematics (11%), Science (23%), Republic History (4%) and Miscellaneous (16%). In order to see how different subjects affect the Q&A environment, we have calculated the previously stated general statistics for each subject (see Table 3.2). All subjects are similar considering the number of answers and the rate of answered questions. However, they are quite different in other metrics especially median first answer times and accepted answer times. English seems like an outlier with its 1 min accepted answer time. This may be due to the extreme skewness in the answer times of English questions.

We observe that English and Mathematics have the highest ratio of accepted answers with 48% and 38.8%, respectively. Social Science has the least such ratio with 25.4%. This may be because the objectivity of questions is very high with English and Mathematics according to our observations. For instance, English questions are basic translation questions (e.g., "*What is the translation of ... ?*") and Mathematics questions are basic problems (e.g., "*if two cars move at 30 km/h, what is the ... ?*") with one correct answer most of the time. On the other hand, Social Science questions tend to be more verbal and opinion asking (e.g., *"What is the role of social media in ...?"*). Is it the case that it takes more time for verbal and opinion asking questions to receive answers? In order to quantify this, we randomly selected and labeled 100 questions as factoid/non-factoid resulting in 73% - 27%, respectively. Using the Mann-Whitney test, we calculated that non-factoid questions receive answers later than factoid ones with $p < 0.07$.

3.3.6 User Interest

We want to know whether students put more interest on some subjects than others, i.e., do they behave like straight A students with the know-it-all attitude. To examine this, we calculate how diverse their answers are in terms of their subjects. We use user-subject entropy in Equation 3.1 as a diversity measure (simplifying from [75]). $P(x_{ij})$ represents the probability of user i to answer in subject j. Here, lower scores mean less diversity.

$$H(x_i) = -\sum_{j=0}^{j=7} P(x_{ij}) log P(x_{ij}) \tag{3.1}$$

We show entropy distribution of students in Figure 3.8. We do not see too much skewness, meaning there is a good amount of users who are diverse but there is considerable amount of users who are quite focused at the same time. In Yahoo! Answers, they behave similarly [75].

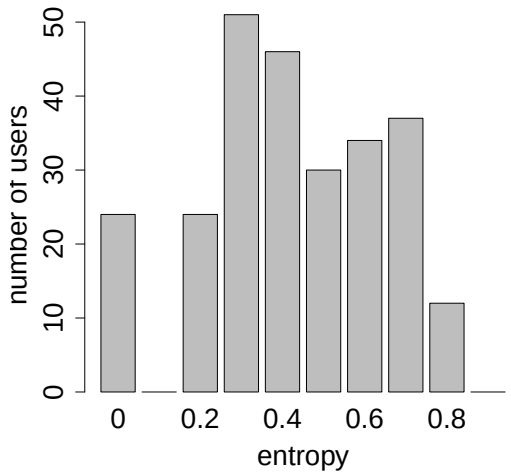


Figure 3.8: User-Subject Entropy

3.3.7 How do users answer and comment?

Good answerers are valuable in Q&A websites. When we analyze how answering takes place in their activities, we see that more active users tend to have more answers with y = 0.5 and above in Figure 3.9a. We calculate Spearman correlation as 0.53 ($p < .000001$) which indicates that there is a positive correlation between the answering ratio of a user and overall activity. Too much commenting and not answering is undesirable for the community. We plot the commenting ratio of users in Figure 3.9b which is almost the opposite of Figure 3.9a. Here we calculate Spearman correlation as -0.752 ($p < .000001$), which means that there is a negative correlation between the commenting ratio of a user and overall activity. Because, in this Q&A site comments are not moderated, most of the comments are in the form of "Thank you", "How could not I think of that before" type text and do not prevail extra information. There is a new version of the website where they do not allow commenting. Many Q&A communities such as StackOverflow, have disabled comments until a user reaches a reputation level by answering questions.

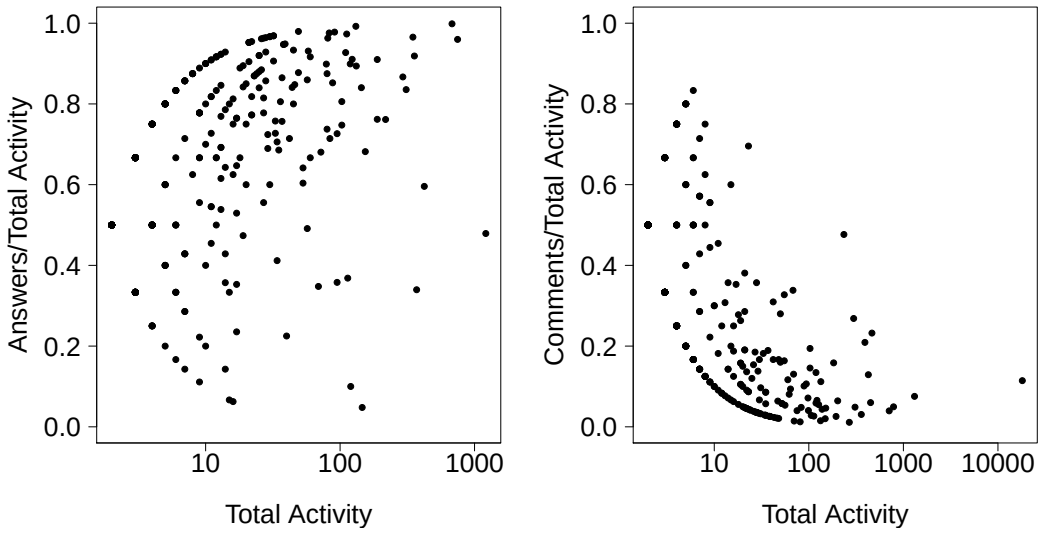


Figure 3.9: Answering and Commenting Behavior

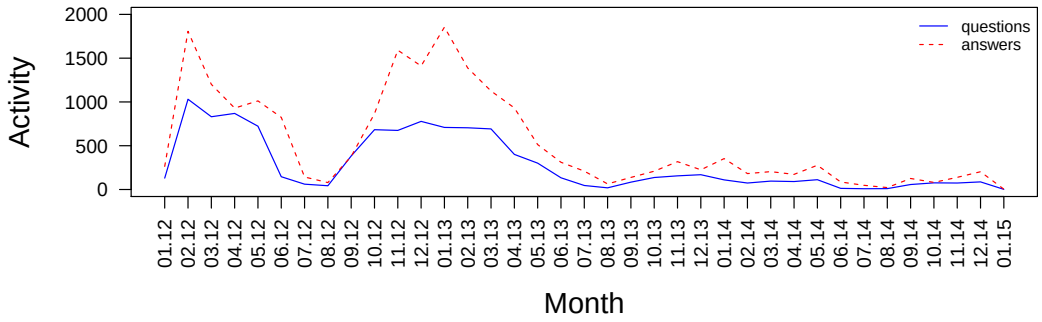


Figure 3.10: Website Activity over 3 years

3.3.8 Educational Seasonality

Students tend to study harder towards the end of semesters and they rest during the holiday seasons. This means that unlike general purpose Q&A websites, an educational website may go through seasonal fluctuations in terms of the traffic they receive. Figure 3.10 shows that there exists such a pattern. We observe local maximums during exam periods such as January and May and local minimums during summer season. On the other hand, the website seemingly has lost user interest over the last half of its lifetime.

3.4 Conclusion

We analyzed the entire data of an educational Q&A website. We investigated various aspects of user activity including asking questions, answering questions, commenting. We observed that an educational Q&A website, in general, follows a similar pattern to that of a general Q&A website, regarding user activity and how it changes over time. We also identified some of the differences between general and educational Q&A websites. We observed that educational Q&A websites have temporal peaks and dips in terms of overall activity. We associated this with school semesters and exam periods. We also showed that users follow different patterns towards questions from different categories.

Initiatives thinking of running such systems need to evaluate what students need, how they need it and when. Such systems must be prepared to overcome the ups and downs of user contribution levels. If they want to be successful and keep their user base, they need to present new motivations. Therefore, analyses of existing such initiatives should be helpful. For example, why the subject website experienced a catastrophic user churn and was unable to recover from it as it did the previous year can be researched from the education perspective. Another research direction would be to interview a group of students about their motivation contributing to these websites and compare these to that of adults in general Q&A.

Chapter 4

Classification of Educational Questions

4.1 Introduction

Although web search may provide useful information to answer a question posed in the form of keywords, it is the user's job to select the correct resources and come up with an answer to her question. Question answering (Q&A) systems try to automate this process for more complex, natural language questions. Research on this topic has led to development of these systems to answer factual questions such as "What is the currency of Turkey?", but the difficulty of questions has no limit. The harder the question, the less powerful the automatic Q&A systems become. Some questions are not easily answered with web searches as well. This has led to the forming of many online services to answer user questions. According to Harper et al. [22], these services evolved to basically three forms such as Digital reference services like "Ask a Librarian","Ask-an-expert services" and Community Q&A websites. Using the power of social elements, community or social Q&A websites have become one of the best addresses for the job. In fact, Gazan [19] says that they are the best ones in terms of answer quality. Some well-known examples of these websites include StackOverflow.com, Yahoo! Answers and Quora. These

websites maintain such atmosphere that user reputation systems have been built into to further motivate user participation.

As expected, social Q&A communities are also very preferable among students enrolled in primary schools to high schools. A Turkish social Q&A website, EOdev.com (translates to E-Homework.com), has over 4.5 million questions where students from different ages ask questions in various topics including Mathematics, Science, English, Social Sciences etc. and find answers written by willing friends. Although there is no proof, it may be preferable for students to receive direct answers to their homework questions, instead of going through a web search process and inferring answer on their own. In fact, according to our dataset, 72% of student questions, when searched on Bing, returned the perfect document (i.e., highest relevance) in the top 10 results and 42% in the top document. Although the communities in other websites such as StackOverflow may not tolerate asking questions without doing a basic search on search engines, educational online communities tend to overlook this behavior. Our work concentrates on this type of educational social Q&A websites and especially the questions.

In Q&A, understanding what the question is about is crucial to answering it. Therefore, classification of questions in the sense of finding its category (e.g., Mathematics, Science, History, ...) is important as it may improve answer quality. For instance, question answering systems, by categorizing questions, can improve bringing more relevant answer candidates matching the category. It may also be used to improve the direction of categorized questions to the experts of these categories. It can also aid the content preparation of highly popular question categories. Educational questions, in this sense, when categorized, can be forwarded to the teachers who are experienced in those categories. It can also serve as an indication of how students struggle with each subject. For instance, the high number of Social Sciences questions may indicate that the subject is less understood compared to other subjects.

In this part of our thesis, we present a new question classifier specifically designed for educational subjects. We first provide an implementation of an ensemble classification method that utilizes several supervised machine learning algorithms and unsupervised algorithms using external resources. We then present how the ensemble method has been enhanced using a general purpose search engine. Section 4.2 provides the list of datasets we collected and used. Section 4.3 describes the problem setting and the ensemble method. It also presents how classification can be improved using search engine result pages (SERPs). Section 4.4 presents the experimental results. Section 4.5 suggests future research directions and concludes this chapter.

4.2 Data Sets

In this work, we utilize the content of a Turkish educational Q&A website, textbooks that were taken from Turkish Ministry of Education website [76], educational objects from Vitamin education service [4], and query results obtained using Bing API [77] in response to educational questions. Below, every data set is explained in detail.

4.2.1 Educational Social Q&A Website Data

We have collected the entire Q&A data from msxlabs [78]. This website has an interface very similar to StackOverflow; there are questions, comments, and answers. Functions are also similar; there are up votes, down votes and selection of best answers. An analysis of the website data is provided in Chapter 3. The reasons behind choosing this website as a resource include the continuous moderation of the content, its content being in coordination with the current curriculum of the education system, its active user base and its structural similarity to well-known Q&A websites.

4.2.2 Textbooks

Textbooks for students are great resources to find answers to questions. We have downloaded the pdf versions of the books from meb.gov.tr with the subjects listed above for every grade in Turkish Education System. Then, we have extracted the content of these pdf textbooks and merged the books from different grades with the same subject, leaving each document representing a corpus for a subject. In the end, there are 7 large documents that cover the subjects: Turkish, English, Mathematics, Republic History, Science and Social Sciences.

4.2.3 Online Course Content

Vitamin web service [4] is a paid educational tool in Turkey. In their system, there are learning objects representing tiny bits of information in different subjects. A learning object consists of a title (i.e., an educational topic), a description and a path (i.e., subject such as Math, Science, etc., grade information such as 8th grade). For us, they serve as mappings from topics to subjects. We have 6264 unique objects like this. Although not extensive, this content was prepared by the experts of the company and is, therefore, reliable after some preprocessing.

4.2.4 Educational Term Collection

We crawled a list of term-definition pairs from [79] where terms are educational entities such as "mitochondria : an organelle...". Our compiled list contains about 33K terms. For each subject, the numbers of terms are: Mathematics(6,71), Science (7,071), Miscellaneous (14,870), Social Sciences (6,974), Turkish (1,552), Religion (1,430), Republic History (1,170).

4.2.5 Bing Query Results

We have also used the Bing API to collect top 50 query results for a randomly selected set of 5000 questions from the msxlabs data. Considering that some queries returned less than 50 results, we have approximately 240,000 query results.

4.3 Educational Question Classification

We approach the question classification first as a text classification problem. We follow the bag-of-words model so that terms are considered as features. Later, we introduce other features special to question classification which can be seen in the next subsection.

We first implement an ensemble classifier exploiting annotated educational questions, course textbooks, educational term collection and online course content resources (see Figure 4.1). We also implement a query expander using the educational term collection. In the next stage, we incorporate search engine result pages into our classifier to further improve our accuracy. The method is basically as follows:

1. Pre-classify questions and their query results with the ensemble question classifier.

2. Reclassify questions with a weighted majority voting scheme based on pre-classified questions and their query results.

4.3.1 Features

The features of question classification are divided into three categories: lexical, syntactic and semantic. In our work, we mostly benefit from the survey in [53].

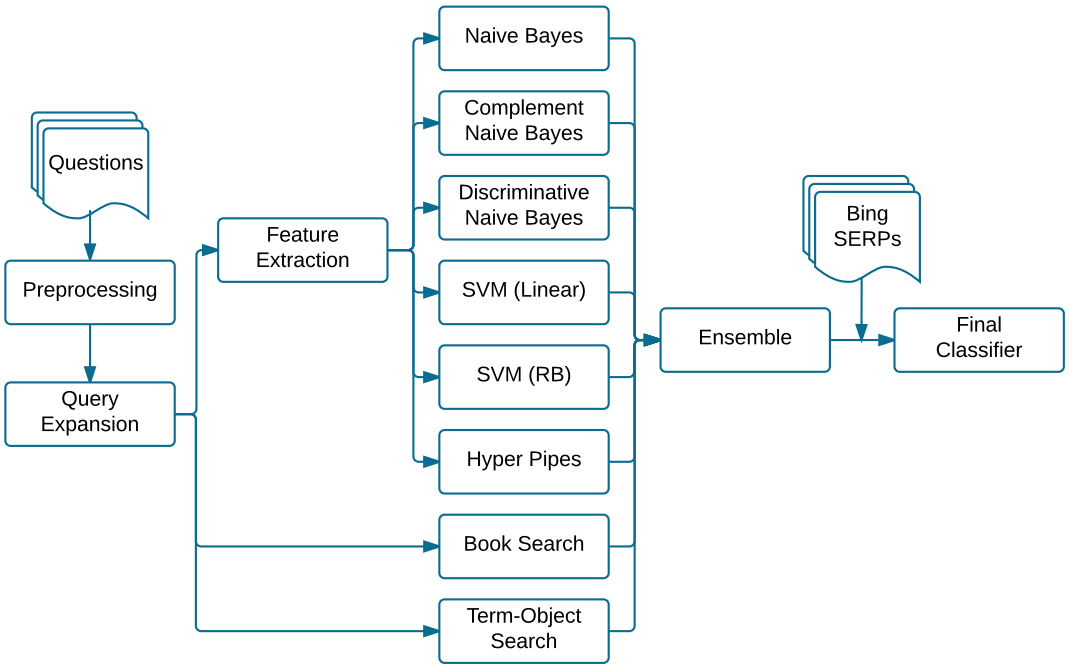


Figure 4.1: Overview of the Educational Question Classifier

The features we implement are presented in Table 4.1. In this section, we give the details for each of these features.

4.3.1.1 Lexical Features

Lexical features are simply the words in a question rather than the grammatical structure of it.

Unigrams, Bigrams, and Trigrams The result set of the tokenization or the sequencing a sentence is called the n-grams of that sentence. In information retrieval terminology, using these tokens as features is named as the bag-of-words model. Unigrams, bigrams, and trigrams are the most widely used types of n-grams.

Table 4.1: Features and Abbreviations

Type	Feature	Abbreviation
Lexical	Unigrams	U
	Bigrams	B
	Trigrams	T
	Wh words	WH
	Word shapes	WS
	Question length	QL
Syntactic	POSTags	P
	Fine Grained POSTags	FP
	Tagged Unigrams	TU
	Object and Subject	OS
	Object and Subject Phrases	OSP
	Word Dependencies	WD
Semantic	Synonyms	S
	Antonyms	A
	Hypernyms	HR
	Hyponyms	HO
	Side Concepts	SC
	Associated Words	AW
	Similar to	ST
	Used for	UF
	By goal	BG

Wh-words Wh-words, or question words describe what is wanted about the subject of the question and therefore may be useful in determining the category of the question. In English, *Who, Where, What, Which, When, How, Why* are the most commonly used wh-words. For Turkish, we determined the following words as question words: *Ne, Nere (Nerede,Nereye), Nasıl, Neden (Niçin), Kim, Hangi, Ne Zaman, Kaç*. Since a question can include one or more of these words, we represent this feature as <featurename, occurence>. Consider the question "2016 Yaz Olimpiyatları nerede ve ne zaman yapılacak?" (When and where will the 2016 Olympics take place?). Its extraction is performed as follows:

{(Ne, 0), (Nere, 1), (Nasıl, 0), (Neden, 0), (Kim,0), (Hangi,0), (Ne Zaman,1), (Kaç,0)}

Word Shapes A question can include words in many forms such as *lowercase, uppercase, all digits, mixed* and *other.* For the same question, these features are extracted as:

{(All digit, 1), (Lowercase, 5), (Uppercase, 2), (Mixed, 0), (Other,0)}

Question Length The length of the question in terms of words may be an indicator of its category. It might be the case that longer questions are from a certain category.

4.3.1.2 Syntactic Features

Syntactic features represent the grammatical structure of a question. We start building this type of features using the ITU NLP API [80]. The extraction of syntactic features depends on the construction of the dependency tree of a given question. Consider the question: *Saf maddelerin ayırt edici özellikleri nelerdir?* (What are the distinctive properties of pure substances?). ITU API generates the syntactic information in Table 4.2 using CONLL [81] format. Using the binary dependency information, one can generate a dependency tree as in Figure 4.2.

Table 4.2: Dependency information of "Saf maddelerin ayırt edici özellikleri nelerdir?" question

Id	Form	Lemma	Cpostag	Postag	Feats	Head	Deprel
1	Saf	saf	Adj	Adj	_	2	MODIFIER
2	maddelerin	madde	Noun	Noun	A3pl—Pnon—Gen	8	SUBJECT
3	ayırt	ayırt	Noun	Noun	A3sg—Pnon—Nom	4	MWE
4	_	et	Verb	Verb	Pos	5	DERIV
5	edici	_	Adj	Agt	_	6	MODIFIER
6	özellikleri	özellik	Noun	Noun	A3pl—Pnon—Acc	8	OBJECT
7	_	ne	Pron	Ques	A3pl—Pnon—Nom	8	DERIV
8	nelerdir	_	Verb	Zero	Pres—A3sg—Cop	0	PREDICATE
9	?	?	Punc	Punc	_	8	PUNCTUATION

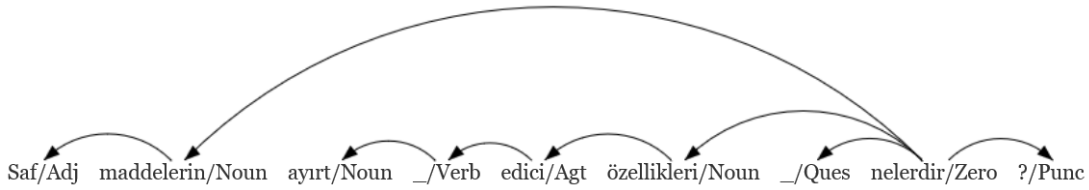


Figure 4.2: An Example Dependency Tree

Part of Speech Tags Part of speech (POS) tags are the most widely used syntactic features. We represent this feature as bag-of-postags (i.e., the words converted to corresponding postags). An example based on the question in Table 4.2 is given below. We also consider the coarse and fine grained POS tags.

{"Adj Noun Noun Verb Adj Noun Pron Verb Punc"}

Tagged Unigrams This feature combines the unigrams with their corresponding pos tags. An example based on the question in Table 4.2 is given below.

{"saf_Adj madde_Noun ayırt_Noun et_Verb -_Adj özellik_Noun ne_Pron -_Verb ?_Punc"}

Object and Subject Determining the object and the subject of a question is different than determining its POS tags. However, they are quite helpful determining its context. The object and subject can be extracted using the dependency structure of a sentence. Analyzing languages with dependency grammar is useful with free word order languages such as Turkish. Dependency parsing has also been studied for Turkish [82]. It is also available using the ITU API. Using the CONLL format we easily extract the object and subject. For 29% and 73% of the questions, we are able to extract the object and the subject, respectively. Only 15% has both the object and the subject. The following is an example of the object and subject information based on the question in Table 4.2.:

{"madde özellik"}

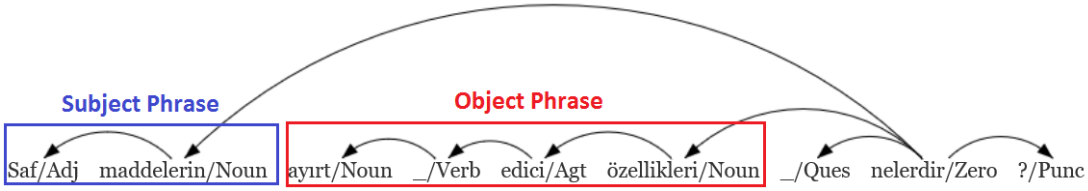


Figure 4.3: Subject and Object Phrases in a Dependency Tree

Object and Subject Phrases After finding the object and subject of a question, we want to expand them to extract more information. The object and the subject are single words but they may be a part of a phrase. If we extract the subject of the question "What do the green plants do to generate energy" as "plant", we miss the phrase "green plant", which may be more informative since it is exactly what the question is about. To find the object and subject phrases, we first find the object and subject and get their dependents recursively. Considering the dependency information in Table 4.2, *madde* is the *SUBJECT*. It has one dependent, which is *saf*. Similarly, *özellik* is named as *OBJECT*. It also has a dependent at the 5th token whose dependent is the 4th whose dependent is the 3rd. By combining these, the object and subject phrases are found as "saf madde" and "ayırt et özellik", respectively. These are much better in capturing the essence of the question. We also show the corresponding covered information in Figure 4.3.

{"saf madde ayırt et özellik"}

Word Dependencies By using the dependency tree we constructed, we extract one to one dependencies and use them as bag-of-dependencies, i.e., they are depth-1 parent-child relations.

{"saf-madde madde-ne ayırt-et et-özellik özellik-ne ?-ne"}

4.3.1.3 Semantic Features

Semantic features provide extra information based on the context of the question. In general, words have different semantic relations to other words, entities in the world. If a classifier is unsure about a word, it can deduce what it is about if it knows another word that relates to that word. However, semantic relations are mostly defined at the word level. Therefore, we extract semantic features based on the objects and subjects of questions. In order to extract semantic features, generally WordNets are used. In this work, we use the semantic relation data set from [83, 84]. It contains 127,203 relations and is general purpose not educational. The data set was constructed using 98,107 words from TDK (Turkish Language Association) dictionary [85] and 160,049 from Vikisozluk [86]. It has "word $\leftarrow$ relation $\rightarrow$ word" format. Based on the data set, the following features are extracted and used as bag-of-words. We also list the ratio of questions that we were able to extract for each feature.

Synonyms Synonyms are other ways of saying words (e.g., *autumn synonym fall*). (Extraction: 65 % of questions).

Antonyms Antonyms are the semantic opposites of words (e.g., *autumn antonym spring*). (Extraction: 1.6 % of questions)

Hypernyms A hypernym of a word is defined as a generalization of it (e.g., *autumn hypernym season*). (Extraction: 27 % of questions)

Hyponyms A hyponym of a word is a specification of it (e.g., *autumn hyponym october*). (Extraction: 1.6 % of questions)

Side Concept While hypernyms and hyponyms result from a parent-child relationship, side concept can be viewed as a sibling relation (e.g., *autumn side_concept summer*).(Extraction: 1.1 % of questions)

Associated Words Associated words denote words that do not have a family relationship but are somehow related (e.g., *observation word_assoc observation satellite*).(Extraction: 6.6 % of questions)

Similar to This relation captures words that are lexically similar (e.g., *observation similar_to observer*). (Extraction: 1 % of questions)

Used For In this relation, the target concept denotes what the source concept is used for (e.g., *eraser used_for erasing*). (Extraction: 0.7 % of questions)

By Goal This relation is quite similar to the *Used for* relation but the target concept is not a direct consequence of the source concept (e.g., *assembly by_goal talking*). (Extraction: 0.7 % of questions)

4.3.2 Ensemble Method

In our experiments, we have used Weka [87] which is an open source data mining software written in Java and equipped with many machine learning algorithms. We tried many classifiers from it. We observed that some of them were able to classify some class of questions better than others. This has led us to think that by combining these classifiers, we may be able to get better accuracy. We chose simple majority voting because it gave us better accuracy. It should be noted that by majority we actually mean first-past-the-post rule, i.e., the subject with the highest number of votes directly wins. Equation 4.1 gives the rule where q represents the question, C_i the classifier i, n is the number of classifiers. Equation

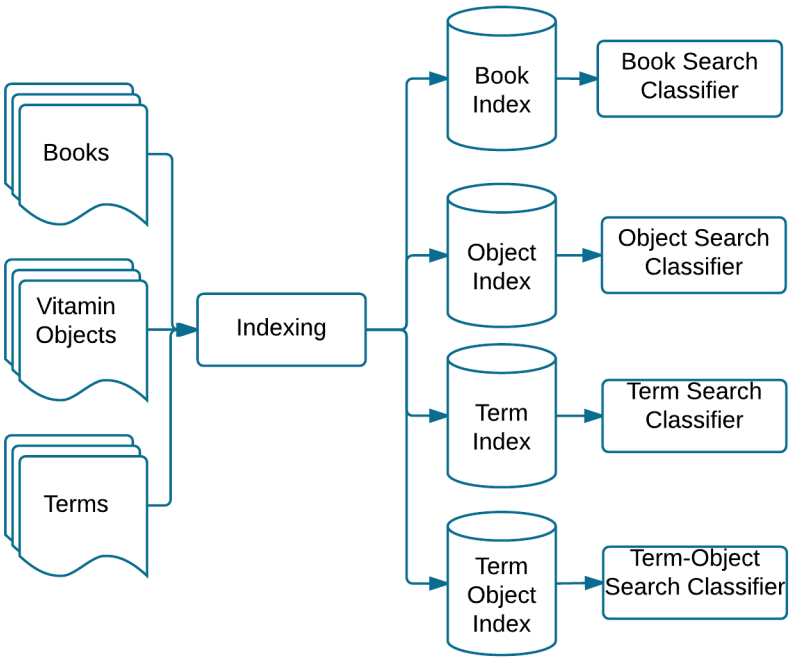


Figure 4.4: Overview of the Search Based Classifiers

4.2 shows how the probability of assigning q to each class is calculated where c_i is the class i and $S = \{c_1, c_2, ..., c_8\}$

$$Classify(q) = mode\{C_1(q), C_2(q), ..., C_n(q)\} \tag{4.1}$$

$$P(c_i|q) = \frac{\sum_{C_j \in C} P(c_i|C_j, q)}{\sum_{c_i \in S} \sum_{C_j \in C} P(c_i|C_j, q)} \tag{4.2}$$

Preprocessing We applied traditional information retrieval techniques in the preprocessing step of the training and test sets. A stemming tool [88] is applied to reduce the number of features. We also remove the Turkish stop words [89] and do lowercasing, and a lot of noise, punctuation removal.

Search Based Classifiers These classifiers utilize the resources we have and do not require training. They are also very fast. An overview of search classifiers is given in Figure 4.4.

Book Search Classifier We have implemented a searcher on the book set using Lucene. First, we indexed the books. If we search a question in these books/documents, since each book/document represents a subject, the first matching document for this question becomes the classified subject. Here, Lucene's default similarity is used and a threshold t=0.025 is applied to further improve the accuracy. Documents passing the threshold are taken into account.

Object Search Classifier We retrieve the subject information such as "Science" from the path column of a learning object given in the format (title, description, path). We treat each object as if it is a document and we want to search through these objects. Thus, we index them (i.e. their title + description) along with their subjects. If a question is asked, we search through our learning object index and return the most similar objects (i.e. documents) in our case. Since the index returns the instances based on a ranking, doing a weighted/graded voting may be better than the equal-weight majority voting. We settle with the following voting function based on our observations. Here, r represents an object returned as a result and i is a class.

$$\gamma(r,i) = \begin{cases} 1 & : class(r) = i \\ 0 & : class(r) \neq i \end{cases} \tag{4.3}$$

$$Classify(Q) = \arg\max_{i} \sum_{j=1}^{k} (\gamma(r_j, i)/(j+1)) \tag{4.4}$$

For the value of k, the number of results to retrieve, we decided on empirical value 40. If we had simple majority voting, k=5 would be our choice. But we decided on the weighted voting because it performs better based on our observations. Furthermore, we use BM25 here without any threshold.

Term Search Classifier We indexed the terms in the educational glossary where a document is represented as <term, definition, class>. Using the same idea in the object search, we implement a classifier.

Term-Object Search Classifier We combined the data from two data sources of educational glossary and Vitamin learning objects and build a search classifier that searches in this combined index.

Machine Learning Classifiers We have used the following classifiers: Naive Bayes Multinomial Updateable (NB), Complement Naive Bayes (CNB), Discriminative Naive Bayes (DMNB), SVM (Linear, Radial Basis(RB)), and HyperPipes (HP). For all these classifiers, we have tried to optimize their specific parameters also including normalization and the number of words to store.

4.3.3 Query Expansion

We implement a query expansion (QE) mechanism to enhance the query class for classification. The idea is to expand the question with terms from the same subject in order to increase the accuracy of the training and testing. Rather than using traditional query expansion mechanisms such as Rocchio's or finding the headword of the query then expanding it, we use a simpler approach taking advantage of the structure of our term list. Using the term list we compiled, we create an index in the form <title, definition, class>. Taking the question as a document, we search through our index finding the most similar documents. Since the title itself intuitively represents the most important word in that document, we simply take the title as our expansion term. However, further improvements are necessary. We calculate the score of each document first and use an empirical TFIDF threshold 0.8. Documents passing the threshold are taken into account. Furthermore, before expanding a query we ensure that the expansion term is not a rare term. We assume that terms that occur in a small number of documents are to appear in a small number of instances in classification as well, which makes the classification of these terms harder. For this reason, we simply check whether the terms document frequency is larger than a threshold (3 in this case). We also find out that larger number of expansion terms degrades the classification and use only one expansion term. Finally, since we know the classes of the expansion terms

as they are indexed, we do a premature classification of the input query using a voting among the returned documents and find an initial class. Documents that do not belong in this class are discarded. After all these improvements, we observe that only around 14% of queries are expanded. Below, we give some examples of questions and resulting expansion terms after applying our expansion method.

Bilgisayar ağları nelerdir?—internet
What are computer networks? — Internet

Dünyanın şekli tam olarak nasıldır?—geoit
What is the exact shape of the Earth? – Geoid

İğ ipliği oluşturma görevi hangi organele aittir—sentriyol
Which organelle is responsible for creating interconnecting fibers?–Sentriole

Hangi bilim dallarında çalışanlar MLyi sık sık kullanırlar?—biyomekanik
In which disciplines do people use ML frequently?–Biomechanics

In our query expansion experiments, we observed that we were able to provide hints, if not exact answers, to the posed questions. In fact, query expansion is highly used in automatic question answering systems [90].

4.3.4 Exploiting Search Engine Results

Search engine result pages are relevant to the question itself and it is intuitive to use SERPs for classification. In fact, if we had a query result classifier, we may have expected that the majority of results returned to the query to be from the same subject. Since building a new classifier for SERPs requires extra effort of labeling, we use our question classifier to classify the SERPs as well. Since SERPs

are in a different form than questions, we only use the bag-of-words features for classifying them and do not use the syntactic and semantic ones.

For finding out what the majority of query results returned to a query says about the subject of the query, we use a voting scheme that emphasizes the ranks and per-class probabilities of query results and initial classification of the question. In this way, top documents will have a higher impact on the voting (see Equation 4.5). Here, c_q represents the final class of the query. d_j is a query result returned to query q at rank j. $P(c|q)$ and $P(c|d_j)$ are the initial probability of query q and query result d_j being in class c, respectively. These are calculated using our ensemble method (see Equation 4.2). We also optimize the classification accuracy by selecting top 20 results.

$$c_q = \arg\max_c \left[\lambda P(c|q) + (1-\lambda) \sum_{d_j \in D} \left[\frac{P(c|d_j)}{\log_2(j+1)} \right] \right] \tag{4.5}$$

λ is a confidence parameter between 0 and 1. When it is bigger than 0.5, we put more trust on the initial classification of the query. When it is smaller than 0.5, we trust more on the knowledge of query results. In order to incorporate trust, we use the assumption that questions with fewer terms are less informative about their classes. Therefore, in classifying shorter questions, we trust the voting among query results more with smaller λ. This result in the following setup: For shorter questions (i.e., shorter than or equal to 6 tokens) $\lambda = 0.3$, otherwise $\lambda = 0.5$. The choice of "6" is not arbitrary. It is the median number of tokens of a question in our data set. Additionally, using this method we are able to calculate the final class wise probabilities $P(c_i|q)$ of queries without needing a transformation, which can be used for various purposes later.

We use the snippets of query results when classifying them. Other approaches may include aggregating all query results and making a classification on this aggregated document. Utilization of full web pages was also possible. We also tried those and chose the previously stated individual weighted voting scheme over these other approaches because those methods did not show promising results.

Table 4.3: Instance Distributions as Result of Labeling

Social Sciences	1463	English	148
Miscellaneous	795	Mathematics	526
Religion	231	Science	1125
Turkish	511	Republic History	201

4.4 Experimental Results

We have randomly selected 5000 questions from our question set and labeled them based on the classes in Table 4.3 which also shows the class distribution. The decision on this number of classes is in accordance with the external resources i.e., not all data sets shared the same class labels and our setup is a combination that made the use of the external data sets possible. In our experiments, we obtain the accuracy by 10-fold cross validation. For each fold, we train 90% of questions and classify 10% of questions and query results returned to them and then improve the classification of these pre-classified questions using the pre-classified query results by employing the voting technique described in the previous section.

As the first feature space, we first try the bag-of-words model, namely unigrams and bigrams. According to this setup, Table 4.4 gives the results of individual classifiers, our ensemble method and the SERP enhancement over it. The ensemble method contains NB, HP, CNB, DMNB, Book Search, Term-Object Search and SVM (Linear and RB). The results show that Naive Bayes and SVM (Linear) based classifiers perform well but our retrieval based classifiers perform the worst. We also see the small but positive effect of query expansion which is 0.2% over the ensemble. SERP enhancement is observed to be 4.8% over this. Query expansion and SERP enhancements add up to 5% improvement.

Next, we run experiments on a broader feature space and try to find the features that are the most beneficial. Table 4.5 shows our greedy approach reducing the feature space (see Table 4.1 for abbreviations). Our approach is similar to the sequential backward selection (SBS) algorithm where we start with all the features and eliminate those that provide the highest accuracy gain when removed at each step. Jain and Zongker discuss the advantages and limitations of this and

Table 4.4: Individual, Ensemble and SERP Enhancement Accuracies using the Bag-of-Words Model (U-B)

NB	75.9
HP	55.2
CNB	76.4
Book Search	54.0
DMNB	77.2
Object Search*	55.7
Term Search*	54
Term-Object Search	64.4
SVM (Linear)	76.8
SVM (RB)	66.8
Ensemble	78.2
Ensemble+QE	78.4
Ensemble+SERPs	82.0
Ensemble+SERPs+QE	**83.2**

(*:not included in the ensemble)

other feature selection algorithms such as sequential forward selection (SFS) in [91]. As the results suggest, in addition to unigrams and bigrams, we find tagged unigrams, object, subject, hypernyms of the object and subject to be the most beneficial. Table 4.6 gives the confusion matrix for the final classifier which is built using unigrams, bigrams, tagged unigrams, object, subject, hypernyms of object and subject, query expansion and SERP enhancement. Before reviewing class-wise confusions, we will focus on the Miscellaneous category since it is the most confused one. The primary reason is this category hardly has any integrity in itself. During the labeling process, we put questions that do not fit into any other category into this; forcing many one-time or rare questions to be associated with each other. Since we mainly use Bayes based classifiers, that makes even harder to recognize this category which is hard to be established on mere word occurrences. We could opt to further atomize this class into smaller classes with very few instances or we could completely eliminate this class by saying that some of the questions are not classifiable. However, we found many of these questions to be education related, thus kept the class.

Considering class-wise confusions, we observe that the classifier confuses three classes the most: Social Sciences, Science, and Miscellaneous. We will try to

Table 4.5: Lexical, Syntactic and Semantic Features Accuracies

Lexical*	Syntactic	Semantic	Accuracy
WH-WS-QL	P-TU-OS-OSP-WD	S-A-HR-HO-SC-AW-ST-UF-BG	77.62
WH-WS-QL	FP-TU-OS-OSP-WD	S-A-HR-HO-SC-AW-ST-UF-BG	77.32
WH-WS-QL	FP-TU-OS-OSP-WD	S-A-HR-HO-SC-AW	78.78
WH-WS-QL	FP-TU-OS-OSP-WD	HR-HO	78.8
WH-WS-QL	FP-TU-OS-OSP-WD	HR	79.04
WH-WS-QL	FP-TU-OS-OSP	HR	79.22
WH-WS-QL	FP-OS-OSP	HR	77.9
WH-WS-QL	FP-TU-OS	HR	79.24
WS-QL	FP-TU-OS	HR	79.32
QL	FP-TU-OS	HR	79.6
QL	P-TU-OS	HR	79.16
QL	P-OS	HR	78.18
QL	FP-TU-OS	HR	83.3†
-	FP-TU-OS	HR	83.34†
-	TU-OS	HR	**83.58**†

(*:U-B-QE enabled in all)
(†:SERPs enabled)

explain this using the relation between Miscellaneous and Social Sciences. The highest confusion occurs with Miscellaneous questions classified as Social Sciences. This is because Miscellaneous category contains questions similar to Social Sciences but these are too specific to be included in the Social Sciences curriculum or they are so rare that the classifier assigns them to the first class it finds a clue about. Consider the following miscellaneous question example below. Underlined words are the most important words and we see that the most important words of the miscellaneous question are contained in Social Sciences to which the classifier assigns it. The same logic applies to classifying Miscellaneous questions to Science.

Miscellaneous Q:Japonya'da Avukat olmak için ne yapmak lazım?
What should I do to become a lawyer in Japan?
Social Sciences Q.1: Avukatlık mesleğinin toplumdaki yeri ve önemi nedir?
What is the place and the importance of the law profession in the society?

Table 4.6: Confusion Matrix for the U-B-TU-OS-HR-QE-SERPS Model

		predicted							
		Turkish	Math	English	Rep. Hist.	Religion	Soc. Sci.	Science	Misc.
actual	Turkish	447	1	0	5	1	31	2	24
	Math.	2	482	0	0	0	15	3	24
	English	9	0	124	2	0	2	1	10
	Rep. Hist.	6	1	0	157	0	31	1	5
	Religion	4	0	0	0	191	17	3	16
	Soc. Sci.	19	7	0	6	2	1279	55	95
	Science	6	18	0	0	0	69	978	54
	Misc.	39	12	1	2	4	161	55	521

Social Sciences Q.2: Japonya'nın başkenti nedir?
What is the capital of Japan?

Table 4.7 shows the precision, recall and F1 measurements for each class for the first model (U-B-QE-SERPS) and the second (U-B-TU-OS-HR-QE-SERPS). English shows the optimal performance (near optimal for the second model) in precision, meaning that there are no false positives for English questions. This may be because the most of the student questions in the English subject are asking for translations of sentences from Turkish to English or homework assignments such as English summaries of known books. Mathematics shows the overall best performance with the highest F1 score. This may be because the questions on this subject are formed as mathematical problems that have direct answers (e.g., If two trains are approaching each other at A and B km/h, when are they going to meet?) and have not much in common with other subjects. It is also the best in terms of recall which denotes the rate of correct identification of the ground truth. Miscellaneous category is the least successful in terms of F1 score.

Table 4.7: Measurements for Each Class

	U-B-QE-SERPS			U-B-TU-OS-HR-QE-SERPS		
Class	Precision	Recall	F1	Precision	Recall	F1
Turkish	0.81	0.87	0.84	0.84	0.87	0.86
Mathematics	0.92	0.92	0.92	0.93	0.92	0.92
English	1.00	0.84	0.91	0.99	0.84	0.91
Republic History	0.91	0.77	0.84	0.91	0.78	0.84
Religion	0.95	0.83	0.88	0.96	0.83	0.89
Social Sciences	0.80	0.88	0.83	0.80	0.87	0.83
Science	0.88	0.86	0.87	0.89	0.87	0.88
Miscelleanous	0.72	0.65	0.68	0.70	0.66	0.67

4.5 Conclusion

As a first step towards improving the answer quality in educational Q&A sites, we addressed the classification of user questions into the subjects in the K-12 curriculum. In our work, we develop an ensemble classifier in order to classify a subset of questions we collected from an educational Q&A website. This ensemble classifier utilizes a few supervised classifiers as well as unsupervised, external resource based ones. By using SERPs over the ensemble classifier, we obtain the question classification accuracy as 83.2%. We also use other features to slightly improve the accuracy to 83.58%.

There are several directions for future work. We used a question classifier for both questions and query results. Building a classifier specific for query results can also improve question classification (similar to [30]). Furthermore, in using book search, it may be better to have fine grained documents instead of combining all textbooks of a course into a big document.

Chapter 5

Search Engine Result Page Ranking based on Classification

5.1 Introduction

General search engines are one of the most useful tools to students who are trying to find answers to their questions. However, coming from a pool of resources, results may not be as context-aware as they expect. In this work, we try to answer two basic questions: 1) Is there a relation between the relevance of query results and query-document category (i.e., educational subject such as Math, Science etc.) pair? 2) If yes, can we improve the overall relevance by using this category knowledge to make the educational search better?

We use Bing to obtain initially ranked query results to educational questions and to establish a baseline. Note that we may refer to them questions or queries interchangeably throughout the work since these are the questions that are to be issued as queries. We label these queries with their query results and find out that the answer to the first question is yes. After finding query results that may be irrelevant using the classifier described in the previous chapter, we demote them in their lists with different methods. In order to compare these methods,

we use the NDCG metric. We perform our evaluation on query sets that vary in query length and being factoid and non-factoid. We perform significance tests to answer the second question as yes.

The rest of the chapter is organized as follows: Section 5.2 describes the data set and the labeling process. Section 5.3 presents how we use the classifier to indicate query-result similarity. Section 5.4 gives the details of our methods. In Section 5.5 we evaluate our methods. Section 5.6 concludes the chapter and mentions future work.

5.2 Data Set and Labeling

We have randomly selected 150 questions from the 5000 labeled question set. Based on a graded relevance scheme, we have labeled the top-50 results obtained from Bing search engine for these questions. A screen from the labeling system we developed is given in Figure 5.1. The grades were 3, 2, 1, 0 where 3 represents a very good document and 0 represents a completely irrelevant one. Our labeling of query results consists of two parts: Labeling for relevance and labeling for the class.

The resulting relevance of query results after the labeling is given in Figure 5.2b. In Figure 5.2a, we see an expected behavior; the number of highly graded documents decreases as the result rank increases and vice versa.

We also present the relationship between the query classes and the document classes in Table 5.1. It is seen that 99% of the results that do not share the same class ($c_q \neq c_d$), are found to be irrelevant (i.e., P(irrelevant | $c_q \neq c_d$) = 0.99). This means that class difference may be an indicator of irrelevance in the education context. On the other hand, of the irrelevant pages only 39% found to be from different classes (i.e., P($c_q \neq c_d$ | irrelevant) = 0.39).

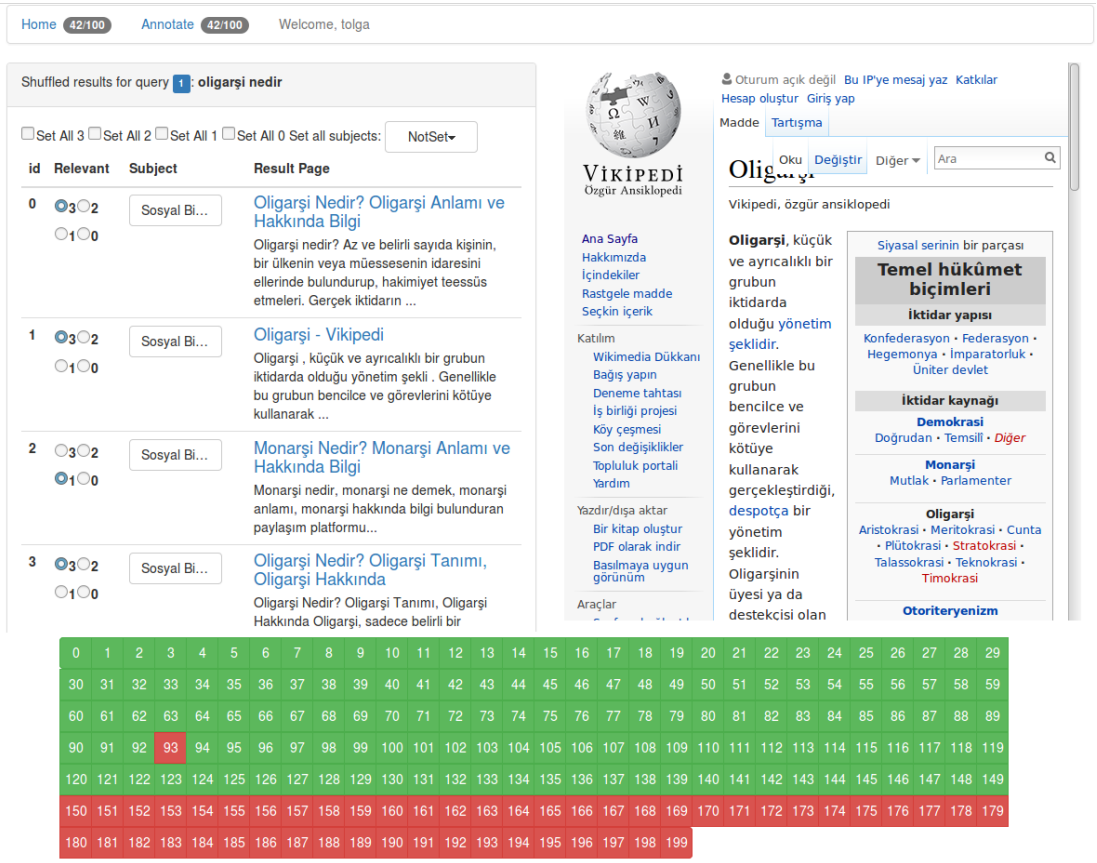
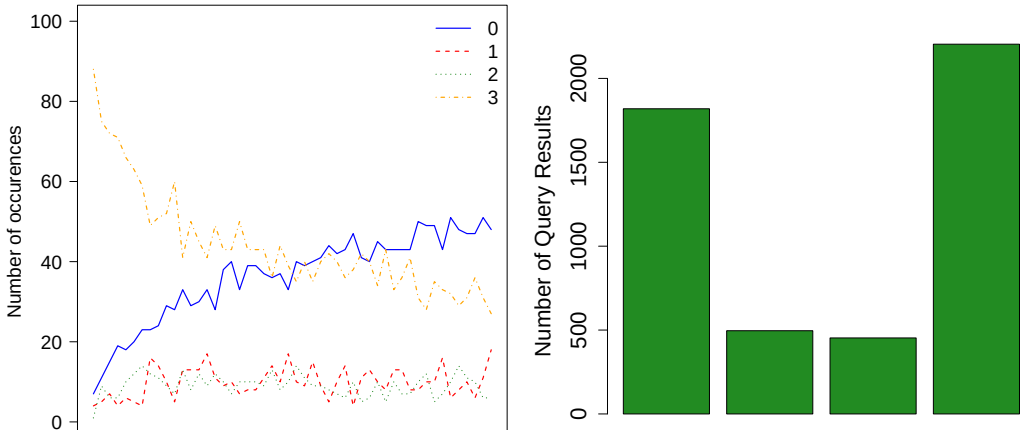


Figure 5.1: Labeling System



(a) Relevancy at each rank

(b) Relevancy overview

Figure 5.2: Relevance Judgments

Table 5.1: Class Match Contingency Matrix

	Relevant	Irrelevant
$c_q = c_d$	4232	1864
$c_q \neq c_d$	9	1192

5.3 Estimating Similarity based on Classification

Retrieval systems try to rank documents based on similarity measures that try to estimate $P(d|q)$, the probability of seeing document d given query q, in different ways. We start with the same point. If we take the relevance to be :

$$Sim(d,q) \approx P(d|q) = \frac{P(q|d)P(d)}{P(q)} \tag{5.1}$$

When we take $P(q|d)$ as $P(c_i|d)P(q|c_i)$ and do this for each c_i:

$$Sim(d,q) \approx \sum_{c_i \in C} \frac{P(c_i|d)P(q|c_i)P(d)}{P(q)} \tag{5.2}$$

When we take $P(q|c_i)$ as $P(c_i|q)P(q)/P(c_i)$ using Bayes theorem and ignore $P(d)$:

$$Sim(d,q) \approx \sum_{c_i \in C} \frac{P(c_i|d)P(c_i|q)}{P(c_i)} \tag{5.3}$$

$P(c_i|d)$ and $P(c_i|q)$ are the probabilities of classifying document d and query q in class c_i, respectively. These are already available using our classification method. $P(c_i)$ can be calculated by dividing the number of instances of c_i in the population by total number of instances: $n_{c_i} / \sum_{c_j \in C} n_{c_j}$. Or, we can further simplify by ignoring this. This reduces the summation to an inner product between probability vectors. Consider the example in Table 5.2.

In fact, by using query and document vectors as shown in the example, where each point is represented by a class probability, we can calculate other vector-based similarity measures. For the sake of this work, we focus on the initial similarity.

Table 5.2: Similarity Example

	$P(c_1)$	$P(c_2)$	$P(c_3)$	Sim(d,q)
query	0.7	0.3	0.0	-
d_1	0.3	0.5	0.2	$0.3 \times 0.7 + 0.5 \times 0.3 + 0.2 \times 0.0 = 0.36$
d_2	0.5	0.3	0.2	$0.5 \times 0.7 + 0.3 \times 0.3 + 0.2 \times 0.0 = 0.44$
d_3	0.6	0.1	0.3	$0.6 \times 0.7 + 0.1 \times 0.3 + 0.3 \times 0.0 = 0.45$

5.4 Proposed Methods

In this section, we list the ad hoc methods we use to improve the ranking for educational queries. We use the following notation:

i : the initial rank of a query result, $0 \leq i < 50$
d : the newly calculated rank of a query result, $0 \leq d < 50$

Additionally, the top ranked query result was too important that we do not touch the ranking of this document for any query.

5.4.1 Point-wise

These algorithms try to demote query results that are classified differently than their query. However, since our classifier, which is described in the previous section is not 100% accurate, we risk demoting pages that can be relevant. These algorithms work basically like this: We find a query result classified differently than the query and its similarity score (described in the previous section) is lower than a threshold, we demote this page to a rank that is calculated by the methods below.

Naive Push Down If we had a 100% accurate classifier, we would directly demote query results to the end of the list. Since we have 50 pages at most, this method pushes a query result to the end of the list.

Step Push Down This method is purely based on the current position i of the query result, aiming to provide a smooth reduction mechanism by dividing by the logarithm.

$$d = i + \left\lceil \frac{i}{\log_2 (i+2)} \right\rceil \tag{5.4}$$

Confusion Push Down We calculate the probability of confusing one class of queries with others using the confusion matrix from Chapter 5 (dividing pairwise confusion by total confusion). With this method, demotion amount is calculated to be larger when transitivity between the class of the query c_q and the class of the query result c_i is smaller. $g(.)$ is a transformation function that scales the probability values which are between 0 and 1 to the desired range. For instance, questions with types "Social Sciences" and "Miscellaneous" can be confused with each other more often than any other pair. This indicates a higher error rate of classification for such "Social Sciences-Miscellaneous" pairs and motivates us to demote such query results more cautiously.

$$d = i + \left\lceil \frac{g(P(c_i|c_q))}{\log_2 (i+2)} \right\rceil \tag{5.5}$$

Hybrid Push Down This method tries to combine Step and Confusion methods. λ is a weighting factor. We take it as 0.5.

$$d = i + \left\lceil \frac{\lambda i + (1-\lambda) g(P(c_i|c_q))}{\log_2 (i+2)} \right\rceil \tag{5.6}$$

5.4.2 List-wise

Rather than thinking about a pointwise demotion technique, we can calculate a generic score for all results that use both the initial ranking and the classification knowledge.

Linear Combination of Initial Relevance and Classification based Relevance If we had the original ranking scores of query results, we could use them.

Since we do not, estimating the initial relevance can be done in different ways. One could take the relevance of a query result at rank i as $1/\log_2 i + 1$ or inspired by the DCG metric; $1/(i+1)$. However, since we want a slower decay we use the relevance function in the form $-a \log(i+1) + b$; incorporation of both this initial relevance estimation and the classification knowledge is achieved in the following equation by using λ as a tuning parameter between 0 and 1. In our experiments, we set it as 0.7. a and b are taken as 0.4 and 2.8, respectively. These values give us a smooth function between 2.8 and 1.2 for i=0 to i=49. This way, we simulate the decreasing relevance with increasing rank.

$$rel(q, d_i) = \lambda rel_{init}(i) + (1 - \lambda) Sim(q, d_i) \tag{5.7}$$

5.5 Evaluation

In this section, we provide the evaluation results of our methods that try to improve educational query result ranking based on classification.

5.5.1 Methodology and Metric

Using the set of query result with relevance judgments, we would be able to evaluate performance of the proposed methods. In information retrieval, for binary judgments many evaluation metrics are used including Precision, Recall, MAP, and Precision@k. Since we employ graded relevance, we use the highly common Normalized Cumulative Discounted Gain (NDCG) to evaluate our methods at various result ranks. NDCG is the normalized version of Discounted Cumulative Gain (DCG) which gives more importance to higher ranked documents, where normalization IDCG is the ideal version of ranking; sorted by relevance values.

$$DCG@k = \sum_{i=1}^{k} \frac{2^{rel_i} - 1}{\log_2 i + 1} \tag{5.8}$$

$$NDCG@k = \frac{DCG@k}{IDCG@k} \tag{5.9}$$

We choose the default ranking of Bing as our baseline. Note that this is a strong baseline that employs various signals to generate query rankings. We mostly pay attention to top 10 query results but provide significance tests for ranks up to 50.

5.5.2 Overall Performance

We run our methods on the set of 150 queries. NCDG@k results (up to @10) are given in Figure 5.3. For earlier ranks, the Naive method is above the baseline but starts to decline for not being able to compensate for FP errors (i.e., demoting relevant pages). Although different in nature, all the other methods seem to outperform the baseline. The Step method seems to work despite the fact that it only depends on the initial rank of the query results.

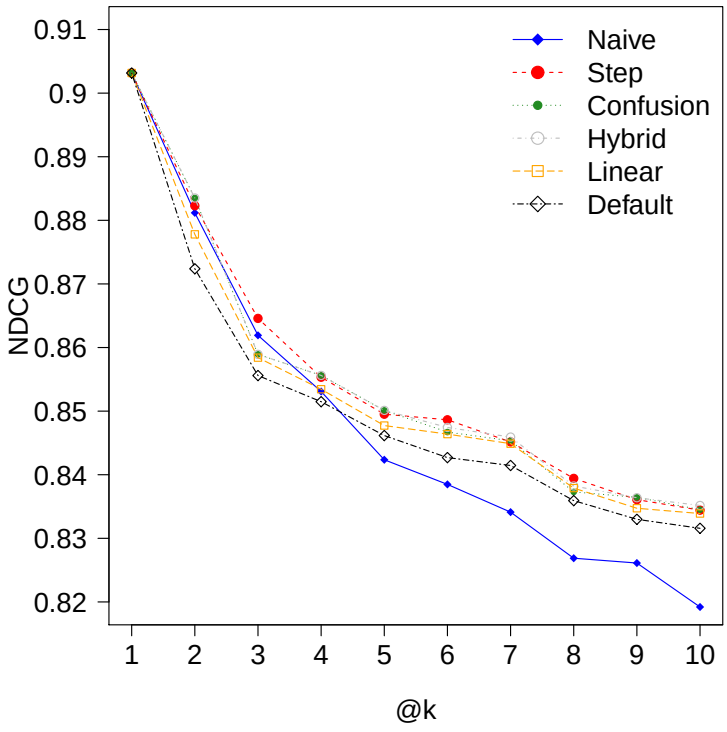


Figure 5.3: NDCG Comparison on the Whole Set

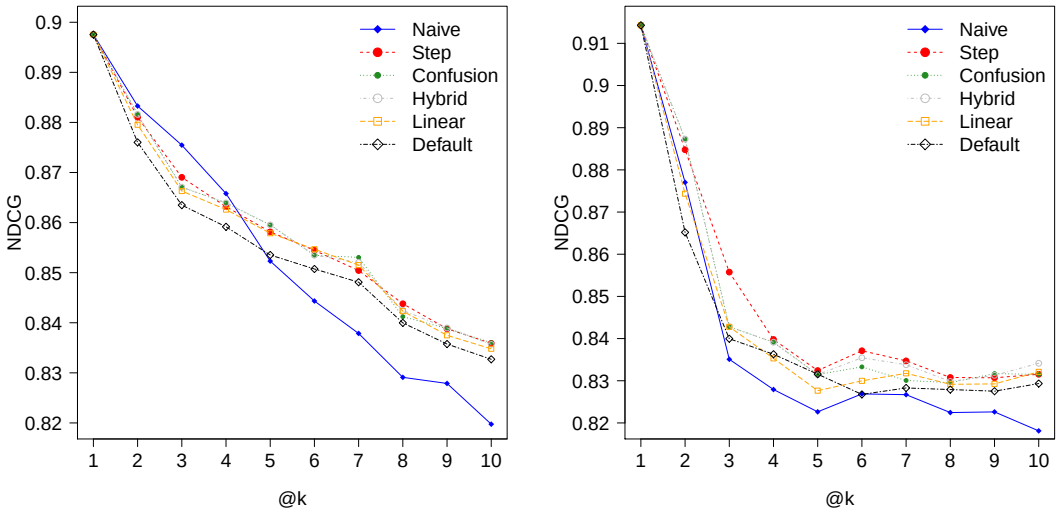
5.5.3 Query Length

In informational retrieval systems, some studies suggest that query length can be associated with retrieval performance since they take part in the smoothing processes of retrieval systems and contain more knowledge about user intent in earlier [92] and later [93] studies. We also want to briefly explore the effect of query length for our query set which consists of educational questions that are in natural language form. Our query set consists of 100 queries that have 6 or fewer tokens and 50 queries with more than 6 tokens. 6 is the median length of the queries in our entire set described in the previous chapters, meaning that our choice of 6 is not arbitrary. In Figure 5.4a we see the retrieval effectiveness of all methods for shorter queries. Although this looks similar to the effectiveness performance on the whole set in Figure 5.3, we see that the Naive method performs slightly different at earlier ranks. For long queries, in Figure 5.4b, the Naive method is not as much separated from the other methods as it is in other evaluations. We associate this to the assumption that longer queries are better classified than shorter ones or they return better results. The latter part of the assumption goes parallel only with the first 2 ranks. Compared to the on the left, NDCG values drop significantly after rank 2. Our methods seemingly perform better in longer queries with respect to the baseline, as they do for short queries.

5.5.4 Factoid vs Non-Factoid Questions

In Question Answering research, factoid and non-factoid questions attracted a wide span of attraction such as [94, 95]. Factoid questions are the ones with simple and mostly one word or phrase answers. An example is given below:

> Q: Wright kardeşler hangi tarihte uçağı uçurmaya başlamıştır?
> *When did the Wright brothers fly their first airplane?*
> A: 1903'te.
> *in 1903.*



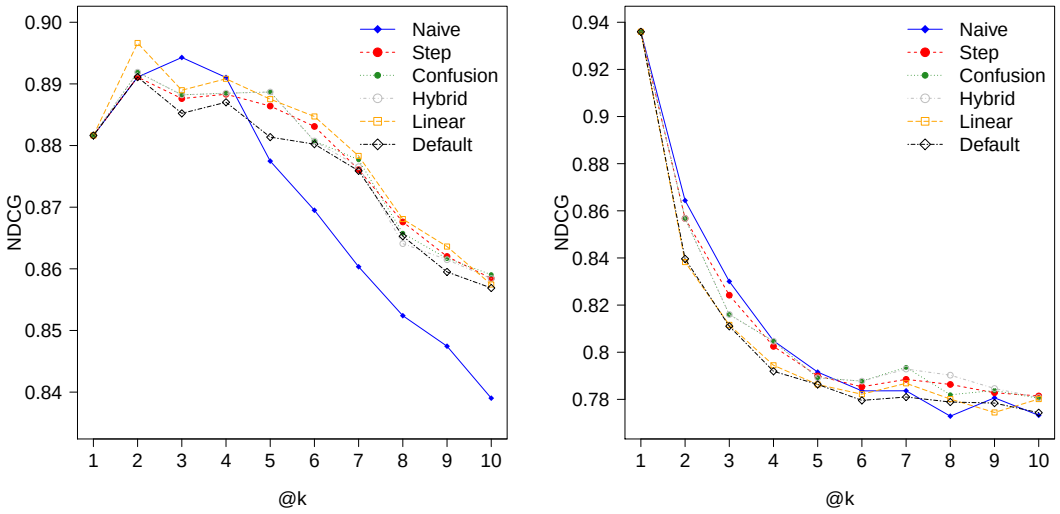
(a) Shorter Queries ($\leq$ 6 terms) (b) Longer Queries ($>$ 6 terms)

Figure 5.4: Changing Query Length NDCG Comparison

A non-factoid question, on the other hand, can be very hard. This type of questions can be in many forms such as opinion seeking, how, why. An example is given below:

> Q: Anadolu'da ve Mezopotamya'da yaşamış uygarlıkların insanlığa yararları nelerdir?
> *What are the benefits of ancient civilizations that existed in Anatolia and Mesopotamia?*
> A: Anadolu ve Mezopotamya medeniyetleri, insanlığın ilk yerleşim yerlerinden olup bilim, sanat, tıp gibi alanlara insanlığın ilk eserlerinin görüldüğü yerlerdir...
> *Anatolia and Mesopotamia, being one of the first settlements of humanity, hosted some of the first explorations in fields such as Science, Art and Medicine...*

In our dataset, we labeled 100 questions on being factoid or non-factoid. We found 73% to be factoid and 27% to be non-factoid. We run our methods on these questions. Figure 5.5a shows that relevancy is very high with factoid type queries and it does not diminish as it is in the other types of queries evaluated in



(a) Factoid Queries (b) Non-factoid Queries

Figure 5.5: Factoid and Non-factoid NDCG Comparison

this section. We also notice that the Linear method outperforms the others only in this type of queries. For non-factoid queries in Figure 5.5b, relevancy declines fast.

5.5.5 Significance Tests

In order to show that our results are significant, we use paired t-test at each rank comparing the resulting rankings of each method to the default ranking. We list the calculated p values at each rank for every method compared to the baseline in Table 5.3. Note that even though we have a small sample size, our results are well performing. Confusion and Hybrid methods seem to have p values smaller than 0.05 and even smaller than 0.001 most of the time. We see that significance increases as the rank increases since the number of comparison points also increases.

Table 5.3: Significance Tests p Values

Method	NDCG													
	@1	@2	@3	@4	@5	@6	@7	@8	@9	@10	@20	@30	@40	@50
Naive	=	0.11	0.18	0.39	-	-	-	-	-	-	-	-	-	-
Step	=	0.06	**0.03**	0.01	0.07	0.008	0.04	**0.06**	0.07	0.03	0.04	0.0008	0.03	0.01
Confusion	=	**0.03**	**0.03**	**0.005**	**0.03**	0.003	**0.007**	0.15	**0.0008**	0.002	**0.001**	0.0006	**0.009**	0.002
Hybrid	=	**0.03**	**0.03**	**0.005**	**0.03**	**0.001**	0.02	0.10	0.02	**0.0003**	0.05	**0.00005**	0.02	**0.0009**
Linear	=	0.16	0.20	0.14	0.29	0.07	0.08	0.23	0.26	0.17	0.27	0.007	0.11	0.09

5.6 Conclusion

We showed that the query and the query result classes matter while using web search for educational queries. Using the educational query classifier we developed and described in the previous chapter, we provide methods to improve the ranking of query results in the education context. We use a general purpose search engine as the baseline and improve the ranking of query results returned from this system. We use NDCG metric to evaluate our methods with respect to the baseline on the different types of queries. Our results show significant relevancy improvement for educational queries.

After showing that the ad-hoc methods work for our purpose, as a future work direction, we can consider a more generic learning to rank approach. For such an approach, possible learning features include the current position of the result, an estimation of the initial relevance, query-document classification similarity (the one we described in this chapter), cosine, euclidean and correlation distances derived from query and result class probability vectors. Of course, a learning to rank approach may require additional labeling of query results.

Chapter 6

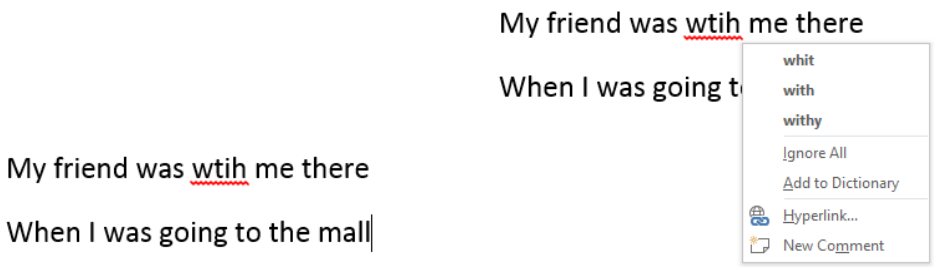
Implementation of a Spell Checker for Educational Queries

6.1 Introduction

In web search technologies, it is quite common for users to misspell their queries. According to Cucerzan and Brill [96] in general web search engines, this rate is more than 10%. In this work, we first examine two search services in the education field, Vitamin Eğitim and Eğitim.com in terms of spelling mistakes using their search logs and we report our findings. We implement an educational spell checker that outperforms context-insensitive state of the art spell checkers on various metrics.

6.2 Spell Checkers Background

According to [97], types of spell mistakes can be identified as insertion (e.g., this → thiss), deletion(e.g., this → thi), substitution (e.g., this → yhis) and transposition (e.g., this → thsi). This is, of course, a generalization and not related to



(a) Spell Mistake Detection (b) Spell Correction Suggestion

Figure 6.1: Microsoft Word Spell Checker

a particular language for which spell mistakes may have different characteristics. Additionally, a word may be in the dictionary but it should be spelled differently (e.g, I want *this* bicycle. → I want *thus* bicycle.). This is again a spelling mistake but this type of mistake is out of the scope of this work. These are called Real-word spell mistakes, whereas the other types are called Non-word spell mistakes [63]. Spell checker programs are for instance built into word processors and search engines. In Figure 6.1a, we see an example of Microsoft Word's spell mistake detection and in Figure 6.1b, we see how some suggestions are provided for correction.

Spell Mistake Detection Spell checkers fundamentally try to tokenize input text into words, then recognize the parts that do not belong in their dictionary as spell mistakes. Other spell mistake detection algorithms, for instance, involve n-gram analysis. In this method, n-grams of input text are compared against a set of n-grams stored in a dictionary [98]. Some statistical methods do not even use a dictionary. Based on statistics, they conclude whether a particular n-gram is likely to occur in a reasonable text [99]. For instance, a Turkish word cannot start with a 2-gram "ğa".

Spell Mistake Correction Correcting a spelling mistake is similar to its detection but more sophisticated. When a spell checker tries to correct a word that it cannot find in its dictionary, it is under the assumption that the misspelled word is a morphological permutation of some word in its dictionary. Then they

try to create a ranked list of possible corrections. Automatic spell correction is when a spell checker replaces the error with the best possible correction. Another approach is to include the user into the picture by actually suggesting her possible corrections and let her decide. Some of the most commonly recognized spell correction methods are Soundex, Edit Distance, Bayesian-Noisy Channel and N-Gram based algorithms.

The Soundex Method Similarity of sounds or pronunciations of words are taken into consideration in this method which goes back to 1918 [100]. In this method, we calculate a code for an input word. Since we know that similar sounding words are expected to have similar codes, we fetch the words having the same Soundex code. Steps of the code generation is as follows:

1. Keep the first letter uppercased.
2. Replace a,e,i,o,u,y,h,w $\rightarrow$ "-"
3. Replace b,f,p,v $\rightarrow$ 1
   c,g,j,k,q,s,x,z $\rightarrow$ 2
   d,t $\rightarrow$ 3
   l $\rightarrow$ 4
   m,n $\rightarrow$ 5
   r $\rightarrow$ 6
4. Merge repeating numbers
5. Remove "-"s.
6. Keep the first three numbers, complete to 3 numbers with 0's if necessary.
   Example: Horse $\rightarrow$ H-62- $\rightarrow$ H620
   Hoarse $\rightarrow$ H--62- $\rightarrow$ H620

Edit Distance An edit to a word is defined as one of the following: inserting a character, deleting a character, substituting a character, transposing characters

[101]. The idea is to match a string with the minimum number of edits. There are three versions based on this idea. Levenshtein Distance [102] is defined as the number of edits to produce a word using another. For instance, the distance between Horse and Hoarse is 1 with "a" as an additional letter. Hamming distance is applied to strings with the same length [103]. Another method is to find the Longest Common Subsequence between strings [104].

Bayesian-Noisy Channel This method is based on Bayesian probabilistic method. Assume that a string is corrupted during a noisy transfer. In this method, we try to find the likelihood of a word actually being another word [105].

$$w_{sel} = \arg\max_{w \in C}[P(e|w)P(w)] \tag{6.1}$$

In Equation 6.1, e represents the word with the error. We calculate the score for each word in the corpus C and select the one with the largest score (i.e., w_{sel}). $P(e|w)$ represents the probability of e being the incorrect version of w whereas $P(w)$ is the probability of seeing w in corpus C. $P(w)$ can be easily estimated as the number of occurrences of w in a text of n words, with $\sim w/n$. $P(e|w)$ is harder to estimate. This is calculated in [105] as the individual probabilities of insertion, deletion, substitution and transposing. For example, for a candidate word w if the operation to transform w is deletion, $P(e|w)$ is calculated as the number of times this deletion happens divided by the number of total occurrences in the set.

N-Gram Based N-gram based spell checkers start with the idea that texts having similar sets of n-grams can be possible spell corrections. For instance, 2-grams of words Horse and Hoarse are {-h,ho,or,rs,se,e-}(6 2-grams) and {-h,ho,oa,ar,rs,se,e-}(7 2-grams), respectively. The union of this set is {-h,ho,or,oa,ar,rs,se,e-} (8 2-grams). The intersection has {-h,ho,rs,se,e-}(5 2-grams). The similarity between two words is to be calculated as in Equation 6.2 where G_{w_1} represents the 2-grams of word w_1. Then $similarity(horse, hoarse) = 5/8 = 0.625$. Note that this formula is the Jackard coefficient.

$$similarity(w_1, w_2) = \frac{|G_{w_1} \cap G_{w_2}|}{|G_{w_1} \cup G_{w_2}|} \tag{6.2}$$

6.3 Proposed Educational Spell Checker

6.3.1 Data Sets

Vitamin [4] is an online service for students enrolled in K-12 level Turkish education system. In order to search through the various educational material, they have an inbuilt search mechanism and gave us some of the query logs issued by students.

Eğitim.com [5], on the other hand, is a free search engine for education material, again in the Turkish context. We were given some of the query logs of their system.

We have randomly selected 2000 non-empty queries out of 25649 from the February 2014 logs of Vitamin Eğitim data set. Similarly, from the Eğitim.com data set of 576821 queries issued between 27.04.2013 and 14.10.2014, we selected 2000 random queries. It should be noted that these queries are non-unique.

In addition, using an educational learning object set provided by Vitamin, we have compiled a list of educational phrases. A learning object is composed of a title (i.e., an educational topic), a description of the topic and a path (i.e., subject such as Math, Science, etc., grade information such as 8th grade). The set consists of 6264 instances and we use the title, description and path columns in our experiments.

6.3.2 Educational Spell Checker (ESC)

We use Zemberek [106], (an open source NLP library for Turkish) as our base spell checker. We create another spell checker using jSpellCorrect [107] (an open source spell checker) which uses our educational word list as its dictionary. Educational Spell Checker (ESC) is defined as the union of these two spell checkers. If the query is identified as misspelled by both Zemberek and jSpellCorrect, i.e., the query does exist in neither Zemberek's dictionary nor educational word list, then we evaluate the correction suggestions from both sources whether any of the suggestions, without considering suggestion order, corrects the spell mistake.

$$ESC.suggest(query) = Zemberek.suggest(query) \cup jSpellCorrect(query) \tag{6.3}$$

In this work, we consider correction suggestions not the automatic correction mechanisms. Also the order of suggestions given by spell checkers are not considered as we do not have their scores. Automatic spell correction can be examined in another context but we observe that automatic correction rates are expected to be low compared to having a correct suggestion in the list of unordered spell correction suggestions.

6.4 Experimental Results

Test queries are tried on Firefox spell checker [108], Chromium spell checker [109], Zemberek library [106], Google Docs, [110] Microsoft Word [111] and ESC. Since these queries are not regular text but in the form of web search queries, keywords, we entered them capitalized into the spell checker tools.

In our experiments, the queries are issued into the tools and we record whether the queries are misspelled and the type of misspelling with the guidance of Türk Dil Kurumu dictionary [112] and careful consideration of educational field. Our newly created spell checker and other tools are compared to each other in terms of the correction and misspelling identification rates.

Table 6.1: Vitamin Eğitim Non-unique Queries Results

	Firefox		Chromium		Zemberek		Google Docs		MS Word		ESC	
	#	%	#	%	#	%	#	%	#	%	#	%
Spell Mistakes	308	15	352	18	352	18	220	11	242	12	267	13
Correction	74	24	136	39	114	32	68	31	122	50	130	49

Table 6.2: Eğitim.com Non-unique Queries Results

	Firefox		Chromium		Zemberek		Google Docs		MS Word		ESC	
	#	%	#	%	#	%	#	%	#	%	#	%
Spell Mistakes	234	12	301	15	284	14	195	10	185	9	248	12
Correction	27	12	61	20	61	21	27	14	56	30	63	25

6.4.1 Non-unique Queries

For the Vitamin Eğitim data set, results using non-unique queries are given in Table 6.1. We have tested 2000 queries. "Spell mistakes", mentioned in the table is the number of queries identified as misspelled by the tools. For example, when Microsoft Word underlines a word with a red line, it is considered as a spelling mistake. It should be noted that not all identified ones are actual spell mistakes, i.e., spell checkers may identify a perfectly correct text as a spelling mistake. This is why some of the spell mistake identification numbers are higher than the actual value. *Correction* metric is whether a tool suggests the correct spelling of a word it identified as misspelled. The order of suggestions does not matter as long as there exists a correct spelling in the lists. According to the manually established ground truth, the number of actual spell mistakes in the set is 240 and it corresponds to 12% of all queries.

For the Eğitim.com data set, again for non-unique queries, results are given in Table 6.2. In total, we have tested 2000 queries. According to the ground truth, there are 129 spell mistakes, corresponding to 6.5% of all queries. Chromium corrects the largest number of queries. Google Docs is the least successful in this metric.

Table 6.3: Vitamin Unique Queries Results

	Firefox		Chromium		Zemberek		Google Docs		MS Word		ESC	
	#	%	#	%	#	%	#	%	#	%	#	%
Spell Mistakes	179	19	213	23	205	22	135	14	162	17	178	19
Correction	56	31	95	45	82	40	45	33	85	52	92	52

Table 6.4: Eğitim.com Unique Queries Results

	Firefox		Chromium		Zemberek		Google Docs		MS Word		ESC	
	#	%	#	%	#	%	#	%	#	%	#	%
Spell Mistakes	204	13	270	18	254	17	175	11	178	12	223	14
Correction	26	13	59	22	52	20	26	15	55	31	54	24

6.4.2 Unique Queries

We have also tested Vitamin Eğitim unique queries. We focus more on this type of queries; although the traditional web search environment witnesses re-occurrence of the same queries, in the context of spell checking it is healthier to consider unique queries so that our results are not biased by an outlying spell mistake that keeps happening so often. Table 6.3 shows the results for unique queries. Out of 937 unique queries, the number of actual spell mistakes is 157 and it corresponds to 17% of all queries. Spell mistakes are corrected the most by Chromium and the least by Google Docs.

Table 6.4. shows the results for unique queries for Eğitim.com data set. From this data set, we have tested 1538 queries. According to the ground truth, there exist 119 spell mistakes corresponding to the 8% of all queries.

6.4.3 Evaluation

We use the binary classification technique in order to understand which tools perform better. Let's explain the terminology. If a tool and the ground truth identify a query as correct, we classify this type of queries as *Correct queries*, or *True Negative (TN)*. If the query is marked as misspelled in the ground truth but a tool fails to identify it as so, this query is classified as *Missing spell mistake* or

Table 6.5: Vitamin Binary Classification Results

	Firefox	Chromium	Zemberek	Google Docs	MS Word	ESC
Extra Spell Mistakes (FP)	53	70	58	36	40	31
Missing Spell Mistakes(FN)	31	14	10	58	35	10
Correct Spell Mistakes (TP)	126	143	147	99	122	147
Correct Queries(TN)	727	710	722	744	740	749

Table 6.6: Eğitim.com Binary Classification Results

	Firefox	Chromium	Zemberek	Google Docs	MS Word	ESC
Extra Spell Mistakes (FP)	124	164	149	102	95	118
Missing Spell Mistakes(FN)	40	13	14	46	36	14
Correct Spell Mistakes (TP)	79	106	105	73	83	105
Correct Queries(TN)	1295	1255	1270	1317	1324	1301

False Negative (FN). If the query is actually correct but the tool identifies it as a mistake, this is classified as *Extra spell mistake* or *False Positive (FP)*. If both the ground truth and a tool mark the query as a mistake, then this is classified as *Correct spell mistake* or *True Positive(TP)*. For these four categories, *Correct queries, Missing spell mistake, Extra spell mistake, Correct spell mistake*, Table 6.5 shows the results for Vitamin Eğitim whereas Table 6.6 shows the results for Eğitim.com.

According to Table 6.5, Google Docs is with the highest number of false negatives whereas Zemberek and ESC have the least of them. Chromium finds the largest number of false positives. ESC finds the smallest. The highest number of true positives is by Zemberek and ESC and the least of them is by Google Docs.

According to Table 6.6, Google Docs finds the highest number of false negatives whereas the least are found by Chromium. False positives are found by Chromium the most and by MS Word the least. Chromium finds the highest number of true positives whereas Google Docs is the least successful.

In order to compare the tools, the numbers above are not enough and we need more accurate metrics. Therefore, we use widely common information retrieval metrics such as precision, recall ve F1 score. Firstly, let's redefine them.

$$precision = \frac{TP}{TP + FP} \tag{6.4}$$

Table 6.7: Vitamin Data Set Spell Checker Evaluation

	Firefox	Chromium	Zemberek	Google Docs	MS Word	ESC
Precision	0.70	0.67	0.72	0.73	0.75	0.83
Recall	0.80	0.91	0.94	0.63	0.78	0.94
F1Score	0.75	0.77	0.81	0.68	0.76	0.88
Correction	0.44	0.74	0.64	0.35	0.66	0.72

Table 6.8: Eğitim.com Data Set Spell Checker Evaluation

	Firefox	Chromium	Zemberek	Google Docs	MS Word	ESC
Precision	0.39	0.39	0.41	0.42	0.47	0.47
Recall	0.66	0.89	0.88	0.61	0.70	0.88
F1Score	0.49	0.54	0.56	0.50	0.56	0.61
Correction	0.25	0.56	0.49	0.25	0.52	0.51

$$recall = \frac{TP}{TP + FN} \tag{6.5}$$

$$F1\ Score = \frac{2TP}{2TP + FP + FN}\ or\ \frac{2 \times precision \times recall}{(precision + recall)} \tag{6.6}$$

Correction, on the other hand, is defined as a tool's ratio of correctly identifying spell mistakes and providing a correct suggestion as a fix (see subsection 6.4.1).

According to these metrics, comparisons of tools for Vitamin Eğitim and Eğitim.com datasets are given in Table 6.7 and Table 6.8, respectively.

According to Table 6.7, precision is highest with ESC (8% better than the closest) and lowest with Chromium. As for the recall, Zemberek and ESC (3% better than the closest) are the most successful and Google Docs is the least. Their harmonic mean, F1 Score, shows that ESC is the most successful (8% better than the closest) whereas Google Docs is the least. For the *Correction* measure, Chromium is the best and Google Docs is the worst.

According to Table 6.8, precision is highest with ESC and MS Word (5% better than the closest). It is at its lowest with Chromium and Firefox. As for the recall, Chromium is the most successful and Google Docs is the least. Their harmonic mean, F1 Score, shows that ESC is the most successful (5% better than

Table 6.9: Types of Spell Mistakes

Spell Mistake Type	Example		Ratio	
	Wrong Spelling	Correct Spelling	Vitamin	Eğitim.com
Fast Typing (FT)	Fiilimisler	Fiilimsiler	7.64	5.88
Wrong Letter (WL)	İntaraktif	İnteraktif	27.39	34.45
Extra Letter (EL)	Etkinlikleer	Etkinlikler	9.55	10.08
Missing Letter (ML)	Oynlar	Oyunlar	25.48	21.85
Compound Word (CW)	İne bahtı	İnebahtı	11.46	16.81
Totally Ambiguous (TA)	Qwerty*	-	18.47	10.92

the closest) whereas Firefox is the least. For the Correction measure, Chromium is the best and Google Docs and Firefox are the worst.

6.4.4 Types of Spell Mistakes

Classifying misspelled queries according to the type of mistakes is studied in the literature [113]. In our work, we also adapt these mistakes to Turkish context. According to our analysis on queries, the most common types of spell mistakes and the ratio of them are given in Table 6.9 based on unique queries.

We also analyzed all the tools for each of these spell mistakes thoroughly in terms of the ratio of correctly identifying each type and the ratio of correcting each type. For example, identifying and correcting *Fast Typing* mistakes is abbreviated as *FT-C*. Identifying but failing to correct the same type of mistakes is abbreviated as *FT-NC*. Therefore, *FT-C + FT-NC* sums up to the total ratio of identifying *FT* errors. The results are given in Table 6.10.

According to Table 6.10, in Vitamin Eğitim data set, the correction rate is highest with Extra Letter mistakes. Totally ambiguous typing is never corrected as expected. Firefox, Chromium, Zemberek, MS Word ve ESC are most successful with Extra Letter mistakes whereas Google Docs performs better with Compound Words.

Table 6.11 shows that, in Eğitim.com data set, the correction rate is highest with Extra Letter mistakes similar to the other data set. Totally ambiguous

Table 6.10: Spell Mistake Detection and Correction by Different Tools on Vitamin Data Set

	FT-C	FT-NC	WL-C	WL-NC	EL-C	EL-NC	ML-C	ML-NC	CW-C	CW-C	TA-C	TA-NC
Firefox	0.42	0.58	0.44	0.47	0.60	0.33	0.35	0.40	0.50	0.28	0.00	0.59
Chromium	0.67	0.33	0.74	0.21	0.87	0.13	0.68	0.28	0.83	0.00	0.00	0.76
Zemberek	0.42	0.58	0.72	0.23	0.73	0.27	0.60	0.33	0.61	0.17	0.00	0.97
Google Docs	0.33	0.58	0.47	0.35	0.33	0.20	0.18	0.43	0.50	0.17	0.00	0.31
MS Word	0.42	0.58	0.79	0.07	0.80	0.20	0.60	0.30	0.56	0.17	0.00	0.31
ESC	0.42	0.58	0.84	0.12	0.93	0.07	0.65	0.28	0.61	0.17	0.00	0.97
Average	0.44	0.54	0.67	0.24	0.71	0.20	0.51	0.33	0.60	0.16	0.00	0.65

Table 6.11: Spell Mistake Detection and Correction by Different Tools on Eğitim.com Data Set

	FT-C	FT-NC	WL-C	WL-NC	EL-C	EL-NC	ML-C	ML-NC	CW-C	CW-C	TA-C	TA-NC
Firefox	0.14	0.86	0.20	0.34	0.50	0.33	0.15	0.46	0.35	0.30	0.00	0.85
Chromium	0.43	0.57	0.51	0.34	0.50	0.33	0.62	0.31	0.65	0.20	0.00	1.00
Zemberek	0.00	1.00	0.49	0.32	0.50	0.33	0.65	0.35	0.45	0.35	0.00	1.00
Google Docs	0.00	0.86	0.20	0.37	0.42	0.25	0.23	0.31	0.35	0.30	0.00	0.69
MS Word	0.71	0.29	0.56	0.12	0.58	0.25	0.54	0.23	0.30	0.15	0.00	0.69
ESC	0.00	1.00	0.51	0.29	0.58	0.25	0.65	0.35	0.45	0.35	0.00	1.00
Average	0.21	0.76	0.41	0.30	0.51	0.29	0.47	0.33	0.43	0.28	0.00	0.87

typing is never corrected as expected. Firefox, Google Docs, MS Word are the most successful with Extra Letter mistakes whereas Chromium, Zemberek and ESC perform better with Missing Letter mistakes.

6.5 Conclusion

Generally, spell checkers identify more spell mistakes than there exists. We also observe the same thing in an education context. Because this context contains more foreign-rooted words or abbreviations like EBOB (GCD), EKOK (LCM) that do not exist in the dictionaries of these tools. As we show with ESC, if we extend the dictionaries of these tools with educational words, it is possible to create a more successful spell checker. The number of words that are identified as spell mistakes even though they are not, drastically reduces with this simple addition.

An obstacle, however, is whether the spell checkers of these tools are compatible with the web search. For example, Chromium spell checker may be more

successful than Google Docs just because it is supported by Google search engine.

Chapter 7

Conclusion

General purpose search engines (GSEs) and community question and answer (Q&A) web sites are two main destinations for students who are seeking information online in order to understand a subject or get homework done. The former comes with problems such as out of context, ineligible or wrong level web pages returned to the queries when searched on. In our work, we focused on bringing GSEs closer to the education context and do not pay attention to other problems for the time being. In that regard, we first perform a short analysis of the content of an educational Q&A website in order to understand user behavior in educational Q&A websites. Then, we create a method to classify educational question queries that are taken from an educational Q&A website. After that, we use classification based re-ranking methods to show that it is possible to improve the relevance of GSE result pages returned to educational queries. Finally, we propose a method that improves spell checkers in the education context.

We list some research directions. We used a rather small educational taxonomy in the classification part in order to match the data sources. However using a much larger taxonomy would be beneficial in building a real-world educational search engine. We approach the question classification problem from the perspective of query and text classification and it may be possible to increase classification accuracy by finding and adding other features such as headwords or catch phrases.

This would require such an extraction method for Turkish, which is yet to be implemented. There may also exist other features to discover that are specific to educational questions or the Turkish language.

Bibliography

[1] "iSEEK-Education." http://education.iseek.com/iseek/home.page. Accessed: 2015-05-04.

[2] "Intute — Jisc." https://www.jisc.ac.uk/website/legacy/intute. Accessed: 2015-05-04.

[3] "Sebit A.Ş. - Ana Sayfa." http://www.sebit.com.tr/. Accessed: 2016-06-22.

[4] "Vitamin Eğitim." http://www.vitaminegitim.com. Accessed: 2015-01-01.

[5] "Egitim.com Öğrenci, Öğretmen ve Veliler için Eğitsel Arama Motoru." http://egitim.com/. Accessed: 2015-01-01.

[6] A. Usta, I. S. Altingovde, İ. B. Vidinli, R. Ozcan, and Ö. Ulusoy, "How k-12 students search for learning?: Analysis of an educational search engine log," in *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pp. 1151–1154, ACM, 2014.

[7] A. Usta, "Optimization of an educational search engine using learning to rank algorithms," Master's thesis, Bilkent University, September 2015.

[8] "Fatih Projesi Web Sayfası." http://fatihprojesi.meb.gov.tr/tr/icerikincele.php?id=6. Accessed: 2015-05-04.

[9] "Eğitim Bilişim Ağı." http://www.eba.gov.tr/. Accessed: 2015-05-04.

[10] "EBA Arama Motoru." http://egitim.gov.tr. Accessed: 2015-05-04.

[11] T.C. Millî Eğitim Bakanlığı, *Millî Eğitim İstatistikleri Örgün Eğitim National Education Statistics Formal Education 2014/'15*. T.C. Millî Eğitim Bakanlığı Strateji Geliştirme Başkanlığı, 4 2015.

[12] "Quora - the best answer to any question." https://www.quora.com/. Accessed: 2015-05-04.

[13] "answers.yahoo.com UVs for May 2015 | Compete.." https://siteanalytics.compete.com/answers.yahoo.com/#.VaPeRfnl9dj. Accessed: 2015-05-04.

[14] "Stack exchange: Hot questions." http://stackexchange.com/. Accessed: 2015-05-04.

[15] "Yahoo! answers education & reference category." https://answers.yahoo.com/dir/index?sid=396545015. Accessed: 2015-05-04.

[16] "Brainly - career & press." http://brainly.co. Accessed: 2015-05-04.

[17] "Eodev.com - Ödevlerin yeni boyutu." http:/eodev.com. Accessed: 2015-05-04.

[18] "Stack Overflow." http://www.stackoverflow.com. Accessed: 2015-12-08.

[19] R. Gazan, "Social q&a," *Journal of the American Society for Information Science and Technology*, vol. 62, no. 12, pp. 2301–2312, 2011.

[20] J. Yang, K. Tao, A. Bozzon, and G.-J. Houben, "Sparrows and owls: Characterisation of expert behaviour in stackoverflow," in *User Modeling, Adaptation, and Personalization*, pp. 266–277, Springer, 2014.

[21] A. Pal and J. A. Konstan, "Expert identification in community question answering: Exploring question selection bias," in *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, (New York, NY, USA), pp. 1505–1508, ACM, 2010.

[22] F. M. Harper, D. Raban, S. Rafaeli, and J. A. Konstan, "Predictors of answer quality in online q&a sites," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, (New York, NY, USA), pp. 865–874, ACM, 2008.

[23] F. M. Harper, J. Weinberg, J. Logie, and J. A. Konstan, "Question types in social q&a sites," *First Monday*, vol. 15, no. 7, 2010.

[24] F. M. Harper, D. Moy, and J. A. Konstan, "Facts or friends?: Distinguishing informational and conversational questions in social q&a sites," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 759–768, ACM, 2009.

[25] X. Cao, G. Cong, B. Cui, C. S. Jensen, and C. Zhang, "The use of categorization information in language models for question retrieval," in *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, (New York, NY, USA), pp. 265–274, ACM, 2009.

[26] G. Wang, K. Gill, M. Mohanlal, H. Zheng, and B. Y. Zhao, "Wisdom in the social crowd: An analysis of quora," in *Proceedings of the 22nd international conference on World Wide Web*, pp. 1341–1352, International World Wide Web Conferences Steering Committee, 2013.

[27] L. Mamykina, B. Manoim, M. Mittal, G. Hripcsak, and B. Hartmann, "Design lessons from the fastest q&a site in the west," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, (New York, NY, USA), pp. 2857–2866, ACM, 2011.

[28] A. Barua, S. W. Thomas, and A. E. Hassan, "What are developers talking about? An analysis of topics and trends in stack overflow," *Empirical Software Engineering*, vol. 19, no. 3, pp. 619–654, 2014.

[29] D. Correa and A. Sureka, "Fit or unfit: Analysis and prediction of 'closed questions' on stack overflow," in *Proceedings of the first ACM conference on Online social networks*, pp. 201–212, ACM, 2013.

[30] D. Movshovitz-Attias, Y. Movshovitz-Attias, P. Steenkiste, and C. Faloutsos, "Analysis of the reputation system and user contributions on a question answering website: Stackoverflow," in *Advances in Social Networks Analysis and Mining (ASONAM), 2013 IEEE/ACM International Conference on*, pp. 886–893, IEEE, 2013.

[31] T. Liu, W.-N. Zhang, L. Cao, and Y. Zhang, "Question popularity analysis and prediction in community question answering services," *PLoS ONE*, vol. 9, p. e85236, 05 2014.

[32] A. Ghosh and J. Kleinberg, "Incentivizing participation in online forums for education," in *Proceedings of the fourteenth ACM conference on Electronic commerce*, pp. 525–542, ACM, 2013.

[33] J. Mao, "Social media for learning: A mixed methods study on high school students' technology affordances and perspectives," *Computers in Human Behavior*, vol. 33, pp. 213–223, 2014.

[34] B. Hecht, J. Teevan, M. R. Morris, and D. J. Liebling, "Searchbuddies: Bringing search engines into the conversation.," *ICWSM*, vol. 12, pp. 138–145, 2012.

[35] B. M. Evans, S. Kairam, and P. Pirolli, "Exploring the cognitive consequences of social search," in *CHI'09 Extended Abstracts on Human Factors in Computing Systems*, pp. 3377–3382, ACM, 2009.

[36] X. Si, E. Y. Chang, Z. Gyöngyi, and M. Sun, "Confucius and its intelligent disciples: Integrating social with search," *Proceedings of the VLDB Endowment*, vol. 3, no. 1-2, pp. 1505–1516, 2010.

[37] K. Komiya, Y. Abe, H. Morita, and Y. Kotani, "Question answering system using q & a site corpus query expansion and answer candidate evaluation," *SpringerPlus*, vol. 2, no. 1, pp. 1–11, 2013.

[38] T. Mori, M. Sato, and M. Ishioroshi, "Answering any class of japanese non-factoid question by using the web and example q&a pairs from a social q&a website," in *Proceedings of the 2008 IEEE/WIC/ACM International*

Conference on Web Intelligence and Intelligent Agent Technology-Volume 01, pp. 59–65, IEEE Computer Society, 2008.

[39] I. Gurevych, D. Bernhard, K. Ignatova, and C. Toprak, "Educational question answering based on social media content.," in *AIED*, pp. 133–140, 2009.

[40] F. Sebastiani, "Machine learning in automated text categorization," *ACM computing surveys (CSUR)*, vol. 34, no. 1, pp. 1–47, 2002.

[41] D. Shen, J.-T. Sun, Q. Yang, and Z. Chen, "Building bridges for web query classification," in *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 131–138, ACM, 2006.

[42] E. Gabrilovich, A. Broder, M. Fontoura, A. Joshi, V. Josifovski, L. Riedel, and T. Zhang, "Classifying search queries using the web as a source of knowledge," *ACM Transactions on the Web (TWEB)*, vol. 3, no. 2, p. 5, 2009.

[43] H. Cao, D. H. Hu, D. Shen, D. Jiang, J.-T. Sun, E. Chen, and Q. Yang, "Context-aware query classification," in *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pp. 3–10, ACM, 2009.

[44] R. Agrawal, X. Yu, I. King, and R. Zajac, "Enrichment and reductionism: Two approaches for web query classification," in *Neural Information Processing*, pp. 148–157, Springer, 2011.

[45] U. Hermjakob, "Parsing and question classification for question answering," in *Proceedings of the workshop on Open-domain question answering-Volume 12*, pp. 1–6, Association for Computational Linguistics, 2001.

[46] D. Zhang and W. S. Lee, "Question classification using support vector machines," in *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pp. 26–32, ACM, 2003.

[47] X. Li and D. Roth, "Learning question classifiers," in *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pp. 1–7, Association for Computational Linguistics, 2002.

[48] D. Metzler and W. B. Croft, "Analysis of statistical question classification for fact-based questions," *Information Retrieval*, vol. 8, no. 3, pp. 481–504, 2005.

[49] Z. Huang, M. Thint, and Z. Qin, "Question classification using head words and their hypernyms," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 927–936, Association for Computational Linguistics, 2008.

[50] J. Silva, L. Coheur, A. C. Mendes, and A. Wichert, "From symbolic to sub-symbolic information in question classification," *Artificial Intelligence Review*, vol. 35, no. 2, pp. 137–154, 2011.

[51] B. Loni, "Enhanced question classification with optimal combination of features," Master's thesis, Delft University of Technology, August 2011.

[52] M. Mishra, V. K. Mishra, and H. Sharma, "Question classification using semantic, syntactic and lexical features," *International Journal of Web & Semantic Technology*, vol. 4, no. 3, p. 39, 2013.

[53] B. Loni, "A survey of state-of-the-art methods on question classification," tech. rep., Delft University of Technology, June 2011.

[54] B. M. Vlasák, "Online school–educational content classification and recommendation," Master's thesis, Masaryk University, May 2015.

[55] H. Li, B. Samei, A. M. Olney, A. C. Graesser, and D. W. Shaffer, "Question classification in an epistemic game," in *3rd Workshop on Intelligent Support for Learning in Groups (ISLG) at the 12th International Conference on Intelligent Tutoring Systems*, Springer, 2014.

[56] A. Sangodiah, M. Muniandy, and L. E. Heng, "Question classification using statistical approach: A complete review," *Journal of Theoretical and Applied Information Technology*, vol. 71, no. 3, 2015.

[57] B. S. Bloom and M. D. Engelhart, *Taxonomy of Educational Objectives: The Classification of Educational Goals: By a Committee of College and University Examiners: Handbook 1.* David McKay, 1969.

[58] A. A. Yahya and A. Osman, "Automatic classification of questions into Bloom's cognitive levels using support vector machines," in *The International Arab Conference on Information Technology*, pp. 1–6, 2011.

[59] S. S. Haris and N. Omar, "A rule-based approach in bloom's taxonomy question classification through natural language processing," in *Computing and Convergence Technology (ICCCT), 2012 7th International Conference on*, pp. 410–414, IEEE, 2012.

[60] N. Yusof and C. J. Hui, "Determination of bloom's cognitive level of question items using artificial neural network," in *Intelligent Systems Design and Applications (ISDA), 2010 10th International Conference on*, pp. 866–870, IEEE, 2010.

[61] A. Figueroa and G. Neumann, "Context-aware semantic classification of search queries for browsing community question?answering archives," *Knowledge-Based Systems*, vol. 96, pp. 1 – 13, 2016.

[62] X. Li and D. Roth, "Learning question classifiers: the role of semantic information," *Natural Language Engineering*, vol. 12, no. 03, pp. 229–249, 2006.

[63] K. Kukich, "Techniques for automatically correcting words in text," *ACM Computing Surveys (CSUR)*, vol. 24, no. 4, pp. 377–439, 1992.

[64] A. R. Golding, "A bayesian hybrid method for context-sensitive spelling correction," *arXiv preprint cmp-lg/9606001*, 1996.

[65] A. R. Golding and D. Roth, "Applying winnow to context-sensitive spelling correction," *arXiv preprint cmp-lg/9607024*, 1996.

[66] A. R. Golding and D. Roth, "A winnow-based approach to context-sensitive spelling correction," *Machine learning*, vol. 34, no. 1-3, pp. 107–130, 1999.

[67] A. Carlson and I. Fette, "Memory-based context-sensitive spelling correction at web scale," in *Machine learning and applications, 2007. ICMLA 2007. sixth international conference on*, pp. 166–171, IEEE, 2007.

[68] E. Mays, F. J. Damerau, and R. L. Mercer, "Context based spelling correction," *Information Processing & Management*, vol. 27, no. 5, pp. 517–522, 1991.

[69] M. P. Jones and J. H. Martin, "Contextual spelling correction using latent semantic analysis," in *Proceedings of the fifth conference on Applied natural language processing*, pp. 166–173, Association for Computational Linguistics, 1997.

[70] J. Nielsen, "Internet-based spelling checker dictionary system with automatic updating," Feb. 23 1999. US Patent 5,875,443.

[71] Y. Zhang, "Contextualizing consumer health information searching: an analysis of questions in a social q&a community," in *Proceedings of the 1st ACM International Health Informatics Symposium*, pp. 210–219, ACM, 2010.

[72] K. Pata, P. Santos, and J. Burchert, "Social recognition provision patterns in professional q&a forums in healthcare and construction," *Computers in Human Behavior*, vol. 55, pp. 571–583, 2016.

[73] O. Kucuktunc, B. B. Cambazoglu, I. Weber, and H. Ferhatosmanoglu, "A large-scale sentiment analysis for yahoo! answers," in *Proceedings of the fifth ACM international conference on Web search and data mining*, pp. 633–642, ACM, 2012.

[74] L. Guo, E. Tan, S. Chen, X. Zhang, and Y. E. Zhao, "Analyzing patterns of user content generation in online social networks," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 369–378, ACM, 2009.

[75] L. A. Adamic, J. Zhang, E. Bakshy, and M. S. Ackerman, "Knowledge sharing and yahoo answers: Everyone knows something," in *Proceedings of*

the 17th international conference on World Wide Web, pp. 665–674, ACM, 2008.

[76] "T.C. Millî Eğitim Bakanlığı." http://www.meb.gov.tr/meb_duyuruayrinti.php?ID=7013. Accessed: 2015-01-01.

[77] "Bing Search API | Microsoft Azure Marketplace." http://datamarket.azure.com/dataset/bing/search. Accessed: 2015-12-08.

[78] "Okula Destek." http://msxlabs.com/okul. Accessed: 2015-01-01.

[79] "Büyük Terimler Sözlüğü Arşivi, Ders Terimleri." http://www.dersimiz.com/terimler-sozlugu/. Accessed: 2016-01-01.

[80] G. Eryigit, "ITU Turkish nlp web service.," in *14th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 1–4, 2014.

[81] "Conll-x shared task." http://ilk.uvt.nl/conll/#dataformat. Accessed: 2016-06-10.

[82] G. Eryiğit, J. Nivre, and K. Oflazer, "Dependency parsing of turkish," *Computational Linguistics*, vol. 34, no. 3, pp. 357–389, 2008.

[83] "Veri kümelerimiz." http://www.kemik.yildiz.edu.tr/?id=28. Accessed: 2016-06-04.

[84] E. Yazıcı and M. F. Amasyalı, "Kavramlar arası anlamsal ilişkilerin türkçe sözlük tanımları kullanılarak otomatik olarak çıkartılması automatic extraction of semantic relationships using turkish dictionary definitions," *EMO Bilimsel Dergi*, vol. 1, no. 1, pp. 1–14, 2011.

[85] "Büyük Türkçe Sözlük - Türk Dil Kurumu." http://tdk.gov.tr/index.php?option=com_bts&view=bts. Accessed: 2016-06-10.

[86] "Vikisözlük: Özgür Sözlük." https://tr.wiktionary.org/. Accessed: 2016-06-10.

[87] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: An update," *SIGKDD Explor. Newsl.*, vol. 11, pp. 10–18, Nov. 2009.

[88] "GitHub - hrzafer/resha-turkish-stemmer: A fast and less aggresive stemmer for Turkish in Java." `https://github.com/hrzafer/resha-turkish-stemmer`. Accessed: 2016-01-01.

[89] F. Can, S. Kocberber, E. Balcik, C. Kaynak, H. C. Ocalan, and O. M. Vursavas, "Information retrieval on Turkish texts," *J. Am. Soc. Inf. Sci. Technol.*, vol. 59, pp. 407–421, Feb. 2008.

[90] M. W. Bilotti, "Query expansion techniques for question answering," Master's thesis, Massachusetts Institute of Technology, May 2004.

[91] A. Jain and D. Zongker, "Feature selection: Evaluation, application, and small sample performance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, pp. 153–158, Feb. 1997.

[92] C. Zhai and J. Lafferty, "A study of smoothing methods for language models applied to ad hoc information retrieval," in *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 334–342, ACM, 2001.

[93] L. Azzopardi, "Theory of retrieval: The retrievability of information," in *Proceedings of the 2015 International Conference on The Theory of Information Retrieval*, pp. 3–6, ACM, 2015.

[94] J. Bian, Y. Liu, E. Agichtein, and H. Zha, "Finding the right facts in the crowd: Factoid question answering over social media," in *Proceedings of the 17th international conference on World Wide Web*, pp. 467–476, ACM, 2008.

[95] R. Soricut and E. Brill, "Automatic question answering: Beyond the factoid.," in *HLT-NAACL*, pp. 57–64, 2004.

[96] S. Cucerzan and E. Brill, "Spelling correction as an iterative process that exploits the collective knowledge of web users.," in *EMNLP*, vol. 4, pp. 293–300, 2004.

[97] F. J. Damerau, "A technique for computer detection and correction of spelling errors," *Communications of the ACM*, vol. 7, no. 3, pp. 171–176, 1964.

[98] E. M. Riseman and A. R. Hanson, "A contextual postprocessing system for error correction using binary n-grams," *Computers, IEEE Transactions on*, vol. 100, no. 5, pp. 480–493, 1974.

[99] R. Morris and L. L. Cherry, "Computer detection of typographical errors," *Professional Communication, IEEE Transactions on*, no. 1, pp. 54–56, 1975.

[100] R. Russell and M. Odell, "Soundex," *US Patent*, vol. 1, 1918.

[101] R. A. Wagner and M. J. Fischer, "The string-to-string correction problem," *Journal of the ACM (JACM)*, vol. 21, no. 1, pp. 168–173, 1974.

[102] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," in *Soviet physics doklady*, vol. 10, pp. 707–710, 1966.

[103] R. W. Hamming, "Error detecting and error correcting codes," *Bell System technical journal*, vol. 29, no. 2, pp. 147–160, 1950.

[104] L. Allison and T. I. Dix, "A bit-string longest-common-subsequence algorithm," *Information Processing Letters*, vol. 23, no. 5, pp. 305–310, 1986.

[105] M. D. Kernighan, K. W. Church, and W. A. Gale, "A spelling correction program based on a noisy channel model," in *Proceedings of the 13th conference on Computational linguistics-Volume 2*, pp. 205–210, Association for Computational Linguistics, 1990.

[106] "Zemberek." `https://code.google.com/p/zemberek/`. Accessed: 2014-12-01.

[107] "jSpellCorrect." http://developer.gauner.org/jspellcorrect/. Accessed: 2016-06-10.

[108] "Firefox'u indirin – Ücretsiz web tarayıcısı – mozilla." https://www.mozilla.org/tr/firefox/new. Accessed: 2014-12-01.

[109] "The Chromium Project." http://www.chromium.org/. Accessed: 2014-12-01.

[110] "Google Docs." https://docs.google.com/. Accessed: 2014-12-01.

[111] "Microsoft Word belge ve kelime işleme yazılımı." office.microsoft.com/tr-tr/word/. Accessed: 2014-12-01.

[112] "Türk Dil Kurumu." http://tdk.gov.tr/index.php?option=com_seslissozluk\&view=seslissozluk. Accessed: 2015-01-01.

[113] H. Duan and B.-J. P. Hsu, "Online spelling correction for query completion," in *Proceedings of the 20th international conference on World wide web*, pp. 117–126, ACM, 2011.

Appendix A

POS Tags

Table A.1: Part of Speech Tags found in the Dataset

Noun	Pers	Interj
Ques	WithoutHavingDoneSo	Without
Zero	NarrPart	Dim
Punc	PCDat	Related
Prop	Demons	Become
Adj	Agt	Abr
Conj	FutPart	AorPart
NAdj	Ly	PCIns
?	Det	Dup
Verb	Quant	AsLongAs
PastPart	ByDoingSo	InBetween
Adverb	Acquire	Inf3
PCNom	Neg	Since
PresPart	PCAbl	
Num	When	
Inf2	AfterDoingSo	
Postp	While	
Rel	JustLike	
With	Ness	
Inf1	Reflex	