MISSING LINK DISCOVERY IN WIKIPEDIA: A COMPARATIVE STUDY

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ÖMER SUNERCAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

JANUARY 2010

Approval of the thesis:

MISSING LINK DISCOVERY IN WIKIPEDIA:

A COMPARATIVE STUDY

submitted by **ÖMER SUNERCAN** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen _____
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Müslim Bozyiğit _____
Head of Department, **Computer Engineering**

Dr. Ayşenur Birtürk _____
Supervisor, **Computer Engineering Department, METU**

Examining Committee Members:

Prof.Dr. İsmail Hakkı Toroslu
Computer Engineering, METU _____

Dr. Ayşenur Birtürk
Computer Engineering, METU _____

Dr. Markus Schaal
Computer Engineering, Bilkent University _____

Asst. Prof. Dr. Pınar Şenkul
Computer Engineering, METU _____

Asst. Prof. Dr. Tolga Can
Computer Engineering, METU _____

Date: _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: ÖMER SUNERCAN

Signature :

ABSTRACT

MISSING LINK DISCOVERY IN WIKIPEDIA:
A COMPARATIVE STUDY

Sunercan, Ömer
M.S., Department of Computer Engineering
Supervisor : Dr. Ayşenur Birtürk

January 2010, 70 pages

The fast growing online encyclopedia concept presents original and innovative features by taking advantage of information technologies. The links connecting the articles is one of the most important instances of these features. In this thesis, we present our work on discovering missing links in Wikipedia articles. This task is important for both readers and authors of Wikipedia. Readers will benefit from the increased article quality with better navigation support. On the other hand, the system can be employed to support authors during editing.

This study combines the strengths of different approaches previously applied for the task, and proposes its own techniques to reach satisfactory results. Because of the subjectivity in the nature of the task; automatic evaluation is hard to apply. Comparing approaches seems to be the best method to evaluate new techniques, and we offer a semi-automatized method for evaluation of the results. The recall is calculated automatically using existing links in Wikipedia. The precision is calculated according to manual evaluations of human assessors. Comparative results for different techniques are presented, showing the success of our improvements.

Our system employs Turkish Wikipedia (Vikipedi) and, according to our knowledge, it is the

first study on it. We aim to exploit the Turkish Wikipedia as a semantic resource to examine whether it is scalable enough for such purposes.

Keywords: Link Analysis, Link Discovery, Wikipedia.

ÖZ

WİKİPEDİA'DA EKSİK BAĞLANTILARI BULMA:
KARŞILAŞTIRMALI BİR ÇALIŞMA

Sunercan, Ömer
Yüksek Lisans, Bilgisayar Mühendisliğ Bölümü
Tez Yöneticisi : Dr. Ayşenur Birtürk

Ocak 2010, 70 sayfa

Hızla gelişen çevrimiçi ansiklopedi kavramı, bilişim teknolojilerinden faydalanarak özgün ve yenilikçi özellikler sunmaktadır. Makaleleri birbine bağlayan bağlantılar bu özelliklerin en önemlilerinden birisidir. Bu tezde, Wikipedia makalelerinde eksik olan bağlantıların bulunmasına yönelik çalışmamızı sunuyoruz. Bu işlev, Wikipedia'yı hem okuyanlar hem de güncelleyenler için önemlidir. Okuyanlar, makaleler arasında daha iyi dolaşma imkanı sayesinde artan makale kalitesinden faydalanacaktır. Diğer taraftan, sistem yazarların güncellemelerini yapması esnasında onlara destek vermek üzere de kullanılabilir.

Bu çalışma, bu alanda daha önceden uygulanmış farklı yaklaşımların güçlü yanlarını birleştirmekte ve bunun üzerine, tatmin edici sonuçlara ulaşmayı sağlayan kendine özgü teknikler önermektedir. Konunun doğası itibariyle sahip olduğu öznellik nedeniyle otomatik değerlendirilmesi zordur. Farklı yaklaşımları birbiriyle karşılaştırmak yeni teknikler için en uygun değerlendirme yöntemi olarak görünmektedir ve biz yarı otomatikleştirilmiş bir değerlendirme yöntemi öneriyoruz. Sonuçların kapsayıcılığı varolan bağlantılar kullanılarak otomatik olarak değerlendirilmektedir. Sonuçların doğruluğu ise değerlendiricilerin kişisel yargıları

dikkate alınarak ölçülmektedir. Farklı yöntemler için değerlendirme sonuçları karşılaştırmalı olarak sunulmaktadır ve bu sonuçlar iyileştirmelerimizin başarısını yansıtmaktadır.

Geliştirdiğimiz sistem Türkçe Wikipedia'yı (Vikipedi) kullanmaktadır ve bildiğimiz kadarıyla, bu onun üzerinde yapılan ilk çalışmadır. Türkçe Wikipedia'yı anlamsal bir kaynak olarak kullanarak bu tip amaçlar için ölçeklenebilirliğini incelemeyi hedefliyoruz.

Anahtar Kelimeler: Bağlantı Analizi, Eksik Bağlantı Bulma, Wikipedia.

To my dear wife

ACKNOWLEDGMENTS

I am deeply grateful to my supervisor Dr. Ayşenur Birtürk, for her contribution to my education, showing me the directions to follow in this study, giving her time and helping constantly, and especially for motivating me since the very beginning of my study.

I would like to thank to the anonymous reviewers of the conferences we have submitted our study (ACL-IJCNLP, EMNLP, WIKIAI, e-Challenges, AAAI Spring Symposia) who have contributed with their comments especially for the evaluation of our study.

I would like to thank Hatice Kevser Sunercan, Yusuf Karagöl, Çağla Okutan, Kezban Demirtaş and Wikipedia author Ant Somers for their valuable support for the manual evaluations of the study.

I would like to thank my colleagues in TUBİTAK-UEKAE/G222 Unit, who always encouraged and supported me, and shared their experiences liberally.

I would like to thank my parents and sisters for all their support during my whole life. Without their love and self-sucrifice, I should not have been the person I am now.

Finally, I would specially like to thank my dear wife Kevser, her endless love, support, encouragement and patience always motivated me during this study.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

Wikipedia [1], the online encyclopedia, involves a large amount of structured, reliable knowledge created collaboratively by authors from all over the world. It improves the traditional encyclopedia concept with abilities of information technologies and provides practical features to users for accessing knowledge faster. Additionally, it is an up to date, dynamic resource growing constantly by comprising a very large subject domain.

Although the World Wide Web contains enormous knowledge, since there is no control over the content, it is mostly not standardized. Therefore, most of this knowledge can not be benefited by intelligent applications employing semantic relationships. When knowledge approaches to a more standardized format, it becomes easier to be processed for introducing more reliable and less faulty intelligent systems. Therefore, the effort spent for the standardization is increasing, for example the creation of large ontologies like Wordnet [2], Cyc [3], SUMO [4] etc. Although they present high quality standards to meet requirements of applications, they could not reach the desired success since the creation mostly needs human effort and consequently suffer from high coverage of up-to-date world knowledge. On the other hand, online systems like Wikipedia dynamically collects the knowledge in parallel with the enrichment of the content according to the needs of the users. The standardized knowledge in Wikipedia has a valuable semantic quality, especially gained by hypertext structures like links, categories and info-boxes forming a semantic network between articles. These features also offer low processing cost than would be the case for processing whole textual content. As a result, Wikipedia increasingly attracts the attention of various kinds of research areas.

1.1 Motivation

Wikipedia's reliability of content is one of its most valuable properties. The main source of this reliability is the auto-control system depending on the large number of authors collaboratively creating and checking the content. There are guidelines instructing authors about the principles of high quality content. Moreover, social mechanisms across the authors like assigned responsibilities for quality assurance, are established. Although obtrusive errors like vandalism and absence of references can be usually detected by these mechanisms, small mistakes that are encountered more are often overlooked. Missing links in articles are example of such mistakes. Another problem is the large amount of articles that are relatively new and have not reached a mature state with respect to the size and quality of content. Since the authors usually do not add all links while entering or updating the content, some articles with missing links remain for a long time unless an author revisits them. For example, textual part of the article *Kaynak Tanımlama Çerçevesi (Resource Description Framework - RDF)* from Turkish Wikipedia is relatively long with about 1500 words. But, it contains only a single internal link by January 2010, although it was created in March 2007. Especially the smaller Wikipedia instances like Turkish Wikipedia face such problems more because of the relatively small number of users. On the other hand, the fast growing instances like the English one also encounter these problems because of the speed of the change. Consequently, the motivation for automatic discovery of missing links is to improve the quality of articles and to help authors during editing. The enrichment of the link structure of Wikipedia might also be important for similar studies which need a high coverage of links for utilizing semantic relationships between concepts.

The Wikipedia guideline concerning the creation of links between articles [5], suggests adding links if they are relevant to the context of the article. Also, links should increase the understandability by providing necessary navigation support. The technical terms, names of people and places should be selected as links instead of ordinary words in the language unless they are important for the context. The terms selected are needed to be relevant to the topic or interesting that could attract readers to explore further. On the other hand, irrelevant or insignificant links decrease the readability of the article. Additionally, again for readability purposes, if a concept is linked once in the article, it should not be linked again unless necessary.

In our study, our aim was to detect missing links according to these general principles. The

challenge for such a system is that it should be aware of the context and should recommend relevant links that do not harm the semantic consistency of the article. Previously, two general, unsupervised approaches were developed to eliminate irrelevant link candidates. The first is trying to match all the terms of the text to the titles of articles in Wikipedia and then specifically filtering out irrelevant ones that are not suitable, as in [6]. The second approach is firstly restricting the scope of the possible link candidate set to ensure the relevancy, then matching these candidates with terms of the text to detect discoveries, as in [7]. We observe the strengths and weaknesses of these previous approaches and apply a combination to eliminate their weaknesses. We also present our contributions to these approaches for improving their success. We offer a novel semi-automatized evaluation approach that increases the objectivity. We have developed an application that allows examining the results of the missing link discovery system. It has also been employed for collecting manual evaluations of human assessors.

1.2 Thesis Organization

The outline of this thesis is organized as follows. In Chapter 2, firstly the features of Wikipedia and its content will be introduced. It will be followed by the literature review on the studies exploiting Wikipedia as a semantic resource and specifically studies on Wikipedia link discovery. Chapter 3 gives the preprocessing applied on Wikipedia data and the environment established for the system. Chapter 4 explains our approach for the missing link discovery problem. In Chapter 5 the comparative results of our experiments will be detailed and discussions on evaluations will be given. Finally, we conclude with the findings of our study and point to the possible future extensions in Chapter 6.

CHAPTER 2

BACKGROUND INFORMATION AND RELATED WORK

This chapter aims to give background information about the domain and present the related studies in the literature with detail proportional to their relevancy with our study. Firstly Wikipedia is introduced and the features of Wikipedia content and the related terminology are given. It is followed with the studies that make use of Wikipedia as a semantic resource. Next section discusses the approaches on Wikipedia link discovery task in detail. The chapter is concluded with general background information on technologies for accessing and processing the Wikipedia content in an efficient way.

2.1 Wikipedia

The term encyclopedia is defined as "a comprehensive written compendium that holds information from either all branches of knowledge or a particular branch of knowledge" [8] in Wikipedia. They are usually used as references to access the specific knowledge related to a subject. Therefore, they are generally organized as articles on specific subjects and mechanisms like alphabetic sorting, indexes are included to allow easy access to needed data.

Online encyclopedia is a modern encyclopedia concept that is published on the World Wide Web media instead of printed form. Online encyclopedias facilitate the access to encyclopedic knowledge by applying the advantages of information technologies and the Internet. The history of online encyclopedias roots in the Interpedia project [9] in 1993 that was composed of some proposals and could not be realized. It was planned as a *free encyclopedia* that are open for contributions of everyone. Since then, large number of online-encyclopedia projects were carried out. Some projects were involved of digitalizing old printed encyclopedias, one suc-

cessful example is the online version of Columbia Encyclopedia [10]. Most of the successful projects were free encyclopedias gaining advantage of the widespread access of users from all over the world [11]. Because, professional examples like Microsoft Encarta project [12] which is discontinued by the end of 2009 were not capable to compete with free opponents because of their limited size. Nupedia was a stable, free encyclopedia established in 2000[13] whose content was created by experts but licensed as free. It has been the predecessor of the free wiki encyclopedia, Wikipedia [1] whose content is contributed by volunteers.

Wikipedia was launched in 2001 and it is currently the leading online encyclopedia. It is a free, collaborative, multilingual encyclopedia project supported by the non-profit Wikimedia Foundation [14]. By 2004, it has become the largest encyclopedia with 300,000 articles. Currently it contains over than 3 millions of articles in English Wikipedia. Also, there are more than 250 language editions and the overall total of article number exceeds 10 millions. The Turkish Wikipedia is the edition of Wikipedia in the Turkish language. By November 2009, it contains about 140,000 articles making it the 19th in ranking.

Articles are the basic entries in Wikipedia, which describe and define an entity or an event. Each article is identified with its name that might be composed of one or more words separated with spaces [15]. Each article is accessed via a standard URL: `http://en.wikipedia.org/wiki/name_of_article`, where the sub-domain 'en' is the language code corresponding to the language of Wikipedia. In the URL, the words in the article name are connected with underscores. Wikipedia involves different types of pages, the article pages are the essential kind which other types are organized around. There is a *talk* page for each article that contains all discussions about the changes in the article content. Multimedia files uploaded like images are displayed in *media* pages. For each user, there is a *user* page to give information about himself. *Category* pages represent the categories of articles that will be explained below. These pages include links to the subcategories and the articles in the category. *Template* pages are the generic ones to be shown inside other pages, like info-boxes (which are structures providing standardized information for same type of entities across related articles, for example a city info-box contains basic attributes to introduce a city). *MediaWiki* and *Help* pages include documentation about the software [16]. As another important feature of Wikipedia, for each page of any kind, the history of changes is preserved and a version of the page at any time can be reached by the users.

The textual content of Wikipedia pages are written with a kind of hypertext language which allows HTML markup and also employs some additional syntactic features to express Wikipedia specific structures like links, category relationships, template usages etc. These additional syntax elements are parsed and interpreted by the special software hosting the Wikipedia pages. The syntax is kept very simple to enable users from all computing skills to learn and edit the content easily.

An article is a hypertext document with hyperlinks to other pages within or outside Wikipedia. Since this study is interested in the links between Wikipedia articles we call them shortly as *links*, and the links to the outside of Wikipedia are called as *external links*. Another terminology used for classification of links is according to their direction; for a specific article, a link inside this article, referring to another one is as called an *outgoing* link and a link inside another article, referring to this article is called as an *incoming* link [16, 17, 18]. The Wikipedia articles are also oftenly called as *Wikipedia concepts* in the studies which the articles are treated as items to represent concepts with semantic value.

The links establish a semantic relationship between two articles. The preliminary prerequisite for an occurrence of a link between articles is the possibility of the users to navigate through the linked article. The cause of the user navigation need might vary, resulting in the variations of kinds of relationships obtained with links. For example, a user might need to navigate to a linked article for the need of more information on the topic. As another reason, a link might seem interesting or intriguing for the reader and a demand to examine that topic might occur. Also some important named entities and concepts (like people, places, events etc.) might be useful to navigate for exploring different aspects intersecting with the topic. More suggestions on deciding to add a link and other important points about linking is given on the Wikipedia linking guideline [5].

The syntax for inserting a link into an article is encapsulating the name of the linked article with double brackets. For example, the sample sentence below contains the links to articles *Russian people*, *writer*, *philosopher*, *Crime and Punishment* and *The Brothers Karamazov (novel)*.

F. M. Dostoyevsky was a [[Russian people\|Russian]] [[writer]] and [[philosopher]], known for his novels "[[Crime and Punishment]]" and "[[The Brothers Karamazov (novel)\|The Brothers Karamazov]]"

A link might be represented with a different anchor text on different articles. The text to be displayed on the page might be determined by appending it after a '|' (pipe) character. For example, the *[[Russian people|Russian]]* link from the sample sentence, connects to the article *Russian people*, but it is displayed as *Russian* on the page. *[[The Brothers Karamazov (novel)|The Brothers Karamazov]]* link is another sample of this usage.

There is another issue to mention with the *[[The Brothers Karamazov (novel)|The Brothers Karamazov]]* sample. The *(novel)* part denotes that various articles with the name *The Brothers Karamazov* exist and the article intended to be linked is determined with this extra identifier. In other words, disambiguation of articles with same titles is accomplished by appending a word in parenthesis to describe the context of the article the disambiguation points to. Another relevant and special kind of articles is the disambiguation pages. These pages list all alternative disambiguations of a title and accessed by using *disambiguation* as parenthesised identifier. For example, the page *The Brothers Karamazov (disambiguation)* lists a novel, two different films and a television series with the same name.

Redirect pages are another kind of special articles. The function of these pages is handling the synonymous article names by making them refer to the same single article to prevent confusions. For concepts with different synonyms the most obvious alternative is selected to be the article including the content. For the rest of the synonyms, redirect pages are created. The redirect pages only contain a link to the actual article with content. For example for the article *Fyodor Dostoyevsky* also the name *Dostoyevsky* might interchangeably used. Therefore, the content is created under *Fyodor Dostoyevsky* name and the *Dostoyevsky* page redirects to it. There is a standard syntax for identifying the redirect pages, which consists of a keyword and a link to the actual article page, like: *#REDIRECT [[Fyodor Dostoyevsky]]*. The synonyms are not the single reason for redirections. For example, if a topic is a subtopic of or needed to be explained inside another one, the redirection might be used, like redirecting from the article *Nords* (the people of Norway) to the article *Norway*. Also, lots of different cases where interchangeable names might occur like abbreviations, misspellings, different tenses etc. can be dealt with redirect pages. The details can be found in the Wikipedia redirecting guideline [19].

Link structure of Wikipedia is also valuable in terms of the multilingual support. Users speaking more than one language can match the same articles in different language editions of

Wikipedia. As a result, a rich cross-lingual resource occurs that goes beyond a traditional dictionary matching concepts in the wide range of the encyclopedia. Matching two concepts is accomplished easily by inserting a link with the article name in the corresponding language and by prefixing it with a language code. For example to link the article *Norway* in English Wikipedia to the Turkish version, the link *[[tr:Norveç]]* should be inserted into the English version.

Category structure of Wikipedia is the other important network consisting of related articles of Wikipedia. The categorization of articles in Wikipedia is an existing feature from the beginning, but association method of articles and categories have been changed in time. Formerly, the categories were a list of articles and list of subcategories. The users were appending articles to categories and sorting them by their topic. The latter category system which is still in use was introduced in May 2004 and it gave a large increase in categorization efforts of users. This system relies on tagging the articles themselves to identify their categories [20].

Categories are tagged into articles in a syntactically similar way with linking. Only difference is inserting *Category:* prefix to the link text. For example to adding the article *Fyodor Dostoyevsky* to the *Russian Writers* category the tag *[[Category:Russian Writers]]* should be added to anywhere in the text of the article [21]. This prefixing is a general syntax style for separating mentioned page types of Wikipedia into different namespaces. For example, to identify user pages, similarly, the *User:* prefix is used. These prefixes are also used in the URL's of these pages. The namespace including the articles is called as *main namespace* or *article namespace* and this is the default namespace of Wikipedia. Therefore the article pages do not have a prefix to be identified [22]. Besides, these prefixes differ for each language edition of Wikipedia. For example, in Turkish Wikipedia the categories are identified with *Kategori:* prefix.

There are no restrictions on the organization of categories, users can add an article to any category and also make a category a subcategory of another. But there is a natural tendency of the structure to be a tree as a taxonomy of articles. Each category contains a list of subcategories containing more specific articles. According to their content, the articles might belong to an arbitrary number of categories at any depth. The Wikipedia categorization guideline [21] briefly describes the structure as: "Wikipedia's categories form a hierarchical structure, consisting in effect of overlapping trees. (Because subcategories can have more than one im-

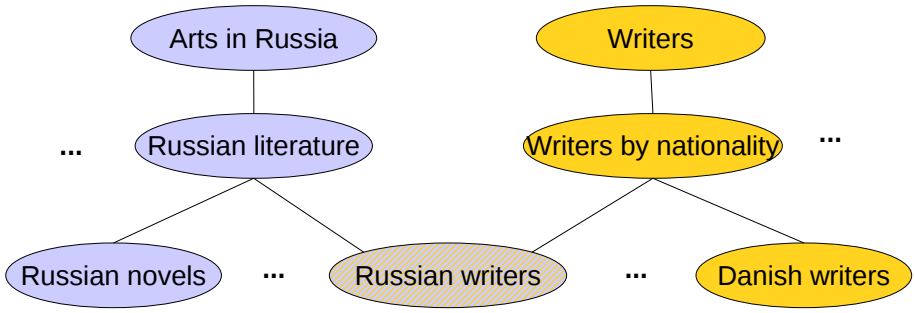

Figure 2.1: Sample of two overlapping category trees

mediate parent, the system as a whole is not a tree, but rather approximates a directed acyclic graph.)". An example overlapping category tree is given in the Figure 2.1.

There are two basic types of categories, list categories and topical categories [21]. The list categories bring concepts of the same type together (*like 'Writers'*). The topical categories are created according to the thematic relatedness of articles (*like 'Writing'*).

2.2 Literature Review on Exploiting Wikipedia as a Semantic Resource

The explained structures in Wikipedia outputs a large amount of standard, machine-processable encyclopedic knowledge. As a result, this resource attracts the interest of researchers from a variety of fields like Natural Language Processing, Data Mining, Information Retrieval etc and Wikipedia is increasingly being subject to important studies. Also scientists from different ares like social sciences also examine the Wikipedia as a social medium where the knowledge is shared.

The link and category structures as a network or graph were analyzed by various studies. The study [23] discusses link and category graphs and gives a graph-theoretical analysis of the category structure to investigate the ability to use them for solving NLP problems. It is shown that they the Wikipedia category graph carries similar characteristics with Wordnet [2] and Roget's thesaurus [24]. As a case study, semantic relatedness measuring techniques for Wordnet were applied on it. The results were satisfactory for measuring semantic relatedness (association by meaning) but worse for semantic similarity (lexical relations like synonymy and hypernimy) algorithms. Voss [16] gives an analysis of measures on Wikipedia's fun-

damental components like articles, authors, edits, and links. It is shown that the Wikipedia grows exponentially. In another study [20], he investigates the category structure of Wikipedia and compares it to the *del.icio.us* collaborative tagging system [25] and hierarchical Dewey Decimal Classification [26]. It is shown that, according to its structural and statistical properties, Wikipedia category structure is a collaboratively developed thesaurus used for indexing subjects. The study [27] also gives an analysis of the category structure. Additionally, the categories are visually layed out on a map as clustered by their subjects and statistics like content similarity, edit time and authors are represented on this visualization.

Wikipedia link structure is also explored as a network of articles by its statistical properties and growth [28, 29]. In these studies network characteristics are compared with World Wide Web and between different language versions of Wikipedia. [30] also covers the article network of Wikipedia and compares it to three traditional text-based network models. As a result, it argues that a different, suitable network model should be developed to represent the network between Wikipedia articles. To analyse the semantic value in the article network, Bellomi and Bonato [31] applied the two well known relevant metrics: HITS [32] and PageRank [33]. According to the results of their case study, they argue that Wikipedia contains facts about cultural biases. Buriol *et al.* [34] explore the temporal information obtained from the edit history of the articles. They analyse evolution of the the article graph, which they call as Wikigraph, and summarize their findings according to the aspects of growth and maturity level of the content.

An important research field employing the Wikipedia knowledge is measurement of semantic relatedness between terms or concepts. Most of the semantic tasks including link discovery requires the determination of relatedness between terms or concepts. Methods using Wikipedia have outperformed the traditional methods for this task. The first instance of the studies is WikiRelate! [35] which compares the effectiveness of Wikipedia for an older method using WordNet. Another technique called ESA (Explicit Semantic Analysis) [36] relies on the text of the articles. For a given text, the matching Wikipedia articles are extracted as a vector of concepts. Relatedness of two texts is calculated by applying a similarity measure on the vectors of these texts. This was the pioneer technique for using Wikipedia for the task by performing much better results than traditional approaches. The method in [37] uses the incoming and outgoing links of the articles to calculate scores for relatedness but can not reach to the success of ESA. [38] is the most recent method that applies some additional link based

techniques to improve ESA, and it reports a small improvement. [39] offers an efficient relatedness measure based solely on the link structure of Wikipedia and calculates the similarity of two Wikipedia articles. They validate their technique by applying this measure to word sense disambiguation.

Detection of key terms and named entities and word sense disambiguation are similar application fields to the link discovery and studies utilizing Wikipedia have been suggested. The algorithm LRT_{wiki}[40] is an improved version the *LRT* (Likelihood Ratio Test) [41] which is a method to identify key terms of documents according to the frequencies of terms in relevant and irrelevant documents. LRT_{wiki} improves this method by including domain specific knowledge obtained from Wikipedia articles. A novel approach for key term extraction is given in study [42], which creates a graph of terms in a document by connecting them according to their semantic relatedness. During construction of this graph, for measuring semantic relatedness of terms in the document, a previously mentioned Wikipedia based approach in [39] was applied over corresponding articles. The key terms are selected according to the density of their interconnections in the graph. Also the dense subgraphs were accepted as signals for different themes in the document. [18] proposes a method that, for a given concept, extracts related text snippets from related Wikipedia articles. Different approaches like content-based, link-based and layout-based methods are compared. Bunescu and Paşca [43] apply a supervised machine learning technique and trains a classifier for detection and disambiguation of named entities using the knowledge in Wikipedia. [44] describes a mostly language independent method that uses Wikipedia to create and maintain gazetteers for named entity recognition.

Another related research area is text classification. Gabrilovich and Markovich [45] exploits the world knowledge in Wikipedia to improve text classification by extending the representation of documents with Wikipedia concepts. For a given text to classify, instead of employing solely the words appearing in the text, they apply retrieval techniques to obtain relevant Wikipedia concepts that carry additional information about the topic. Another study on text categorization [46], considers the Wikipedia categories of the named entities extracted from text to be classified. Schönhofen [47], proposes another method for the same task using the titles and categories of articles, and validates his method with an application to determine categories for Wikipedia articles.

There are studies exploiting the cross-lingual link structure of Wikipedia. Sorg and Cimiano [48] studied enriching the cross-lingual link structure of Wikipedia for exploiting it to solve further cross-lingual natural language processing tasks. For matching articles in different language editions of Wikipedia, they train a classifier with some features based on similarities of link and category structures. [49] extracts similar sentences from two corresponding Wikipedia articles in different languages. Using Wikipedia's cross-lingual link structure, they create a bilingual lexicon. By using this lexicon, similar sentences in two articles are detected according to ratio of links they have in common. For translation of the queries for cross-lingual information retrieval, [50] utilizes the cross-lingual links of Wikipedia along with a dictionary, a stemmer and a text describing the query. For the same task, [51] uses the Wikipedia as the single resource. [52] presents an approach that applies multilingual knowledge of Wikipedia to a cross-lingual question answering system.

In this section, the approaches exploiting the Wikipedia that are related to our work have been summarized. The studies specifically on link discovery problem will be detailed in following section.

2.3 Wikipedia Link Discovery

There has been studies on hypertext creation and analysis on World Wide Web even before the establishment of Wikipedia. [53] gives a summary of applied information retrieval approaches for the task. The subsequent applications of hypertext analysis surveyed in [54]. Automatic insertion of links into the text was experienced with Microsoft Smart-Tag and Google Autolink applications. But these were criticised by modifying the pages with commercial aims and have not been widely accepted. A more recent study [55] has suggested the Creo and Miro systems providing personalized link suggestions according to the user's goal. These are implemented as a Web browser that enable the user to train and modify the behaviour of it. The success of the systems were better when applied to specific and smaller domains. For instance, Drenner *et al.* [56] proposed a system that suggests links and conversations to a movie recommendation site which are brought from a movie discussion forum site. Another system called ARIA [57] recommends relevant photos by analysing the content of an e-mail message. As a result, it is clear that applying link discovery generally over the World Wide Web is different than focusing on a specific domain or resource. The general solutions need

techniques like data mining, search engines queries, user involvement etc. and heavily depend on textual similarity measures because of non-standardized structure of the content. On the other hand, specific solutions take the advantage of domain specific properties. For example, when the Wikipedia is used as a link resource, the structures like links, categories etc. provide insightful evidences about the semantic relationships between concepts. Therefore, most of the studies on the task benefits from the specific features of Wikipedia instead of analyzing whole textual contents of articles.

There are several studies on Wikipedia link discovery. The first study on the field is [7] which introduces the LTRank algorithm that clusters articles according to their relevance and assigns a set of related articles for each article. In the link discovery phase, all of the links in the relevant article set are traversed to find the matching terms. With the relevancy assurance gained by the LTRank, all matches found in this phase are accepted as relevant valid link discoveries. The Wikify system [6] applies a different method which firstly extracts all possible link candidates by checking term matches with all article titles. Then, a keyword extraction technique was applied to measure the relevancy of all candidates and a suitable fixed ratio of discoveries with highest scores are accepted as valid discoveries. Three different keyword extraction technique was implemented and compared in the study. Another study [17] suggests a supervised machine-learning technique. A classifier was trained with features like relatedness, generality, location and keyphraseness which was suggested by the previously mentioned study. A very recent study [58], which is solely based on the links between articles, proposes a method that applies a known statistical technique on the link structure. In the following sections, these approaches will be discussed in detail.

According to their functionality, these studies branches into two different approaches. The first approach aims to suggest links for an article without any links. This suits to the need for suggesting links on a newly created Wikipedia article or enriching the non-Wikipedia articles with links to Wikipedia pages. The second approach is the discovery of the missing links in the existing articles that already have links. The second one is more challenging because it is more subjective and involves a small set of candidate new links for inserting to an already manually annotated article. Additionally, the evaluation is harder because there is not an existing ground truth data set. On the other hand, missing link discovery approach takes the advantage of using the existing links as a resource. Accordingly, the studies [6] and [17] adopts the former approach and suggests links to an article from the rough, whereas the [7]

and [58] adopts the latter approach and aims to discover missing links. Our work might also be accepted as a missing link discovery system, but it also works for the articles without any links along with a probable decrease in efficiency because of loosing the contributions of the article links.

2.3.1 Extracting All Possible Link Candidates and Selecting Relevants

This approach [6] firstly extracts all possible link candidates by traversing the titles of all articles in Wikipedia to find matching anchors in the text. This procedure ends up with a high number of candidates most of which are irrelevant to the article. The recall of these result naturally converges to 100%, but the irrelevancy of the candidates decreases the precision to a very low level. To handle this situation the second step of the approach applies a filtering to the link candidates to eliminate the irrelevant links.

For the implementation of first step, a keyword vocabulary including titles of all articles is constructed. In addition to the appearances in the titles, all different variations of surface forms occurring in the text of articles are added to the vocabulary. It is exemplified [6] with "dissection" link which has two more different appearances that are "dissecting" and "dissections".

In the study [6], the problem is approached as a keyword extraction task. Therefore, for the filtering step, three different keyword extraction techniques are examined to score candidates for selecting relevant ones. These techniques are TF-IDF [59], χ^2 independence test [60] and keyphraseness:

TF-IDF (Term Frequency-Inverse Document Frequency) It is a widely used information retrieval measure that assesses the specificity of a term to a given document. TF is the frequency of the term in the document and determines the importance of the term for the document. DF is the number of documents in the collection in which the term appears. This value gives the general importance of a term. TF-IDF is calculated by multiplication of TF and the logarithm of inverse of DF to evaluate the importance of the term for the specified document.

χ^2 Independence Test This is a statistical method for observing if two events occur together more often than by chance. When applied to keyword extraction domain, in a similar

manner with TF-IDF, it measures if a phrase occurs in a document more frequently than it would occur by chance [6]. The calculation will not be detailed here because it is out of the scope of this text, but it is meaningful to express that a 2 * 2 contingency table is constructed for calculation using the frequencies of the term in the document and in the collection, and frequencies of all other terms in the document and in the collection.

Keyphraseness This measure is addressed by the study [6] and depends on how often a word is preferred as a keyword in the collection. Keyphraseness is calculated by division of the number of documents for which the word is a keyword ($count(D_{key})$) over the number of documents it appears ($count(D_W)$). This gives the probability of the term to be a keyword and can be formulated for word W as:

$$P(keyword|W) = \frac{count(D_{key})}{count(D_W)} \tag{2.1}$$

When this concept is specialized to the link discovery domain, it turns to the probability of a term to be selected as a link. For calculation, number of articles in which the term is a link is divided by the number of articles in which it appears. Briefly, if a term is generally preferred to be used as a link in other articles, it has a higher rank to be selected as a link in the target article too.

After ranking the link candidates according to the keyword extraction method, 6% of them are selected as discoveries. This ratio is determined by the average number of existing links over number of all terms in a document. The study [6] reports that the most successful keyword extraction technique applied was keyphraseness.

The Wikify system in this study was also employed as an application for linking terms in educational materials to Wikipedia articles [61]. The aim of the study was improving the educational resources by linking keywords, technical terms and important concepts to encyclopedic knowledge. Users would benefit from this application by comfortable navigation to articles that are interesting for the topic and accessing to deeper knowledge easily. For evaluation of the system 60 students were asked questions related to study materials that requires further encyclopedic research. Half of the study materials were given as wikified (terms linked to the Wikipedia articles) and the results showed that the wikified materials improved both the correctness and time requirements of the answers.

2.3.2 Discovering Links from Related Articles

The study of Adafre and de Rijke [7] introduces an algorithm called LTRank to find out related articles for a given article. These related articles are used as a source of relevant links to insert into given article. Main innovation of this approach is the selection of links from related articles. Since the challenging part of the missing link discovery problem is determining the relevancy of links, this approach tries to eliminate irrelevant links in an early phase by selecting only the links in related articles.

The LTRank algorithm identifies the similarity of pages by their incoming links and article names. The intuition for this approach comes from SimRank [62]. The SimRank is a measure to calculate similarity of two objects in a graph according to the similarity of their structural context formed by their relationships with other objects. By adapting this idea to the Wikipedia link structure, the incoming links are accepted as the relationships with other objects. LTRank accepts two articles as similar if they both have incoming links from same articles. LTRank clusters the similar articles for a given article in two main steps. Figure 2.2 summarizes the general flow of the algorithm. First step outputs an initial list of similar articles. For each article, article names for all incoming links are collected. It results with a set of related terms extracted from these article names, then a Lucene index is established that stores these terms for each article. Lastly, for each article, a query formed by these related terms is sent to the index to obtain a initial ranked list of articles similar. But linking to an article does not always indicate a direct conceptual relationship, instead, a link might be provided for navigation to potential intriguing articles around the topic. Because of this, all articles found out in this step are not firmly related to the to target article. In the second step, a filtering mechanism is applied to eliminate the less relevant ones. Initially, for each article, 10 most relevant articles from first step are selected. Then, a new Lucene index is established which stores the names of the correspofing selected articles for each article. Then, for each article, a query is sent to new index with these set of stored names. According to the results of this query, articles scoring over a certain threshold are accepted as final related article set.

The resulting system has been run over 144,211 Wikipedia articles and in average 4 missing links per article are suggested. To evaluate the precision of the approach 100 sample discoveries are manually evaluated and 68 of them were found relevant.

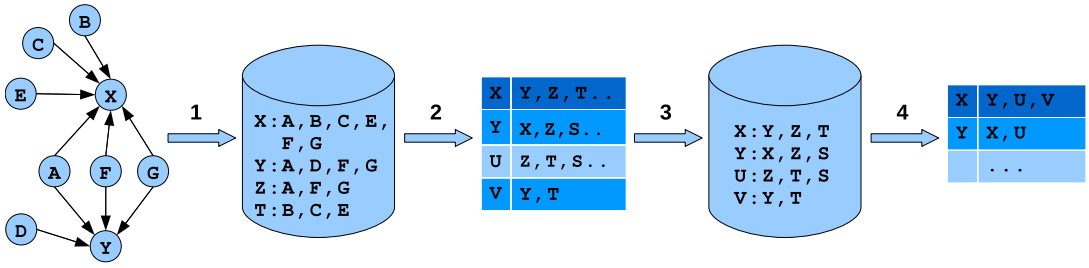

Figure 2.2: Steps of the LTRank algorithm: 1) Index by names of incoming links 2) Find initial relevant article set by querying with these names 3) Re-index with names of most relevant 10 articles 4) Find final relevant article set by querying with these names

We think that there are two apparent drawbacks of this approach. First one is the possibility of insufficiency of number of links gathered from the limited related article set. This number can be affected by the number of the related articles and number of links in these articles. Second drawback is the absence of any relevancy evaluation over the links coming from related articles. All links from the selected related articles might not be relevant or suitable in the context of target article. This might lead to decrease in the correctness of the discoveries.

2.3.3 Learning to Link with Wikipedia

This approach [17] applies a supervised machine learning technique to discover links using existing Wikipedia link structure. The authors criticise the study [6] explained in Section 2.3.1, because of the lack of context information in the application of keyphraseness measure explained above. They suggest a machine-learning approach considering the contextual information obtained from the text in which the term appears.

The technique firstly applies the approach in [6] by selecting a very low threshold for the filtering phase. Thereby some nonsense phrases and stop words are eliminated. Then, disambiguation on selected discoveries is applied. This initial set of discoveries are used to train a classifier by determining the positive and negative examples manually. For training, the contextual features extracted from text and the links from the initial discovery set are employed. They are briefly described below [17]:

Link Probability This feature is the keyphraseness measure suggested in study [6]. This

feature is used because it is a proven technique to determine the suitableness of a link without contextual information.

Relatedness For the disambiguation of discoveries a relatedness measurement is applied by using the approach from a study of same authors [37]. This measure depends on commonness of the incoming links from other Wikipedia articles. This measure is accepted as a feature since relatedness is an important factor to select a term as a link.

Disambiguation Confidence In the disambiguation phase, a probability for all ambiguous alternatives are calculated. This feature is employed in the training as the disambiguation probability of the selected term, because a more ambigous term is more probable to be a wrong selection.

Generality From the view of guiding to readers, the general, mostly known terms are less meaningful to be linked for reaching unknown information. Therefore, the generality of the term is applied as a feature to decrease the chance to be selected. Generality is measured by the depth of the term in the category structure of Wikipedia, where the higher nodes represents more generality.

Locational Features Four different features depending on the locations of the term in the text are applied. First one is the *frequency* of the term in the text. Also, occurrence in the introduction [63] and the conclusions of the text are indicators of importance for a term. Therefore, *first occurrence* and *last occurrence* locations are accepted as features. Last feature is called *spread* which is the distance between first and last occurrences, for considering the consistency in occurrences of term. Last three features are normalized by the length of the text.

2.3.4 Missing Link Discovery with Statistical Dimensionality Reduction

This approach is described in a recent study [58] which employs a statistical method called principal component analysis (PCA). This method is a well-known mathematical generalization technique. Briefly, this procedure transforms a number of possibly correlated variables into a smaller number of uncorrelated variables called principal components. Details of this technique will not be given here since it is out of our scope. For application of this technique to link discovery, firstly a data structure is created using the link structure. An N*N adjacency

matrix of articles is created where N is the number of articles and if there is a link between corresponding articles in the matrix, a value for the link is calculated and put into the box for the link. This value is reverse proportional with the number of total incoming links of the target article. Then, the procedure of PCA is applied which involves a projection and a reconstruction phases over the matrix. The resulting structure contains values for some boxes on the matrix, which were empty for the initial state. These values point to the link suggestions which were absent initially. Consecutively, the approach does not apply to any heuristics or employment of any textual content, instead, utterly the statistical method is relied to intuitively find out missing links. The authors indicate that their approach can be used to point out missing topics of an article and for clustering articles according to their semantic relatedness.

2.4 Accessing Wikipedia Content

For a system that needs to access and process the large amount of data in Wikipedia archive, execution time becomes a considerable problem to solve. For instance, involving all of the edit histories of all pages, the XML dump of the Turkish Wikipedia which is relatively small, is currently about 1.8 GBs as compressed where it increases up to 20 times when uncompressed. The Wikimedia Foundation freely publishes the XML and SQL dumps of all versions of Wikipedia [64].

First possibility to access Wikipedia data is deploying the data in SQL dump to a relational database and accessing programmatically to this database. But this type of access is problematic since it is a raw data that is optimized to present pages on the Web and does present a flexible access mechanism from a software which needs a variety of queries and relationships between objects. Also, it is poor by data access time to support such a software system.

A solution to this problem was tried to be brought by developing an optimized database organization and an object oriented programming API based on Java language [65, 66] called JWPL (Java Wikipedia Library). This API presents a model of Wikipedia structures and provides easy programmatic access with relations between these structures like categories and links of an article. The drawback of this approach is the unsatisfiable execution time for specific requirements of applications which needs complex relationships or queries.

The most widely used approach among the studies on Wikipedia is employment of a text in-

dexing tool and search engine because of the high execution performance. There are variety of open-source and commercial solutions on the area. The most widely preferred alternative is open-source Apache Lucene tool [67]. Beyond being a high performance full text indexing tool and a search engine, it is an Information Retrieval (IR) library [68] which supports searching over text documents with state of the art IR techniques, provides interfaces for customized approaches and utilities like text parsing and stop word removal.

In Lucene, an index is composed of a collection of units called documents. To be able to search on collection the indexing over a collection should be applied once. For indexing, documents in the actual resource are enumerated and index documents are stored into special index files. A document contains a determined set of fields which each of them maps to specific values according to the nature of a document and application. For example, if a set of academic papers are indexed, generally users might apply searches over this collection by title, text and keywords, therefore, the documents should be indexed with these fields. Since Lucene allows assigning multiple values to a field in a document, multiple keywords of the paper are stored for the paragraph field.

Applications and users of the index are provided a query interface to apply different types of queries to retrieve meaningful information over the collection. Lucene allows a combination of two types of query models [68] which are the pure Boolean model and the vector space model. Shortly, a query is composed of a set of Boolean constraints over the results to retrieve. After the set of documents satisfying the Boolean model are retrieved, the vector space model is applied. The vector space model identifies both the query and the retrieved documents as a vector of terms included. For each document, a relevance or similarity score is calculated according to the vector distance measure between the document and the query vectors. A simple query example is given below:

title:Wikipedia AND (keyword:“link discovery” OR keyword:“link analysis”)

This query searches for documents which contains *Wikipedia* in its title and *link discovery* or *link analysis* as a keyword. The search fields and values are given as key value pairs separated with colons and fields are connected with Boolean expressions.

If a term is wanted to be boosted in the query for making it more determiner on the results, a boosting factor might be specified like in the example below:

keyword:“link discovery”^5 OR keyword:“information retrieval”

There also exists an open-source, user-friendly client application that enable users to access and modify existing Lucene indexes called Luke [69].

CHAPTER 3

DATA PREPROCESSING AND ENVIRONMENT

For accessing Wikipedia content programmatically, we have established an environment and applied some preprocessing on the data; this chapter gives explanations about this phase.

3.1 Data

For our study we have used the XML dump of Turkish Wikipedia from April 7th, 2008. This dump does not involve the edit histories of the pages and its size is 418 MBs as uncompressed. This dump contains about 134,000 article and category pages excluding the other kind of pages like user pages. This XML document contains all pages as separate elements and these elements are parsed for index creation. For text indexing and searching, we have used Lucene [67]. The need for performing different types of queries on this data caused creation of two different index structures to be able to meet performance requirements. These indexes are called as article index and inverted index. Details about these indexes are given below.

3.2 Article Index

The article index is the ordinary index structure which stores a document for each corresponding Wikipedia article. The fields in this index can be summarized as:

text Text of the the article is stored in this field.

main_title This field corresponds to the title of the article by preserving the cases in the original article name. It uniquely identifies the article in the index.

title_in_lowercase This field also corresponds to the title of the article, but it is stored in lower case. This field is needed to be able to search case-insensitively. Since the links in the articles most of the time occur in lowercase and this kind of searching is an inevitable need. This field is also a unique identifier for the article.

title This field is another field for indexing the titles of articles which handles the redirection utility of Wikipedia. As mentioned in Chapter 2, the same articles might be referred by different titles. The additional titles have redirect pages pointing to the actual article page like symbolic links in a file system. All different redirect titles to the article are stored in this field. This enables searching and accessing the content of the actual article even for links referring to the redirect pages.

internal_link The set of internal links of an article are extracted from the textual content and stored in this field for retrieving and searching the links of an article.

category Similar to the internal links, the categories of an article are extracted from the textual content and stored in this field.

We have developed an object oriented data model over this index structure by mapping fields of index documents to the object attributes. By exploiting this model, the article and category graphs could easily be traversed independently from the underlying index structure. As a result, an efficient access method is obtained compared to other methods employing databases. At the same time, this index is used for searches like retrieving articles according to their categories, links involved etc.

3.3 Inverted Index

An inverted index [70, 71] is a kind of index where the parts in the content refers to the containing document. For example, for many information retrieval tasks, the words in a collection are usually indexed to point the documents which they occur, sometimes along with the information of location in the document. This structure increases the performance of some kinds of accesses like determining document frequency for TF*IDF calculation which is number of documents the word occurs. Our system has also needed some similar queries based on the titles of Wikipedia articles and we have established an inverted index. This index

contains a document for each Wikipedia article title. This document has three fields which are explained below:

title This field stores the title which is indexed.

containing_article This field contains the set of Wikipedia articles that the title occurs in its textual content.

linking_article This field stores the set of Wikipedia articles which contains a link to the article identified with the indexed title.

This index is used for performing queries like calculation of term and link frequencies in the article collection.

3.4 Index Contents

Only the article and category pages have been inserted into the index. Other kind of pages like user pages, discussion pages etc. were not included since they do not carry meaningful information about the articles and would inaccurately affect the results of the experiments.

Another consideration about the content involved in the indexes was the articles and links for dates. The date articles contains lists of events that occurred on a named date, and actually do not contain any topic. On the other hand, we have observed that Turkish Wikipedia contained a plenty of date links especially in articles like ones describing historical events, biographies etc. To obtain an idea about the amount of them, we have made a little experiment and observed that for randomly selected 592 links, 80 of them (about 13.5%) were date links. From the view of linking, since pages for the dates usually do not contain important information about a specific content, links going to those pages did not seem to us very meaningful. We decided to investigate the linking conventions for date articles in English Wikipedia. We have found out that some discussion on the topic has been made during 2008 and as a result they were cleaned out from the articles. Consecutively, we have decided to ignore all date articles and links and excluded them from the indexes. Due to this constraint, our system also does not discover any date links.

3.5 Parsing the Article Text

As it was explained in Chapter 2, in addition to HTML markup, the Wikipedia pages uses special syntax elements to express particular structures like links and categories. To determine these entities during the creation of indexes, this information embedded in the article text should be extracted and a parsing should be applied. As an open-source project, the Lucene library contains a contrib branch which contains custom applications based on core features. One of these applications is the Wikipedia analyzer component which tokenizes the text into tokens by classifying them according to their types. These token classification involves eighteen types like alphanumeric text, internal link, external link, citation, company name, heading, subheading etc. This component uses a Java based lexical analyzer generator tool called JFlex [72]. The JFlex tool needs a syntax definition file containing the keywords and syntactic relationships. Since we have used the Turkish Wikipedia, we applied some changes to this file like replacing the keywords with Turkish equivalents. Also, since the analyzer component was in an experimental phase, we have found some bugs and features unsatisfying our needs both in the source code and the definition file. Therefore, we have applied some necessary improvements on the analyzer.

CHAPTER 4

WIKIPEDIA MISSING LINK DISCOVERY

In this chapter, we will explain details of our approach on the Wikipedia missing link discovery problem. The chapter starts with details of our method which adapts the general idea from the study explained in Section 2.3.2 and modifies its implementation by suggesting various comparable techniques to find out best resulting alternatives. The next section will discuss the approach explained in Section 2.3.1. It will be followed by the section which details our proposal to improve the approach in the previous one. The last section will explain our method which is composed of a combination of the two improved approaches.

4.1 Discovering Links from Related Articles

The first method we have employed is locating related links from a set of related articles similar to the approach in [7] which was described in Section 2.3.2. In this approach, the relatedness of links is assumed to be ensured by the assumption of related links are included in related articles. Therefore, all links in the selected related articles are accepted as valid discoveries if they occur in the article to be enriched which we call as the *target article*.

As it was explained, the method suggested by Adafre and de Rijke which is called as LTRank algorithm, applies a two steps technique to create an index containing a set of related articles according to the incoming links of articles. This approach, which is called as LTRank algorithm, has seemed us unnecessarily complex and we wanted to show that a simpler method that is based on the articles or categories involved in the target article would be sufficient to determine the related articles to be a source of related links. Therefore, we follow a different approach for the deciding related articles than the LTRank. To experimentally evaluate and

compare, we define different related article sources that each of them will be detailed in following sections. After collecting the candidate related articles, we apply a filtering mechanism to select a set of mostly related articles according to the scores from a relatedness measurement technique. This measurement relies on the overlap of links involved in the articles to be compared.

After determining the set of related articles; links in the related articles are matched in the text of the target article. The steps performed can be summarized as follows:

1. Collect candidate related articles.
2. Calculate the score for each article according to the link overlap measure.
3. Select the best scoring articles as related articles.
4. Traverse the links in the related articles and find out the matches in the text to be the discoveries.

Details of these steps will be explained below.

4.1.1 Collecting Related Article Candidates

As mentioned above, we wanted to show simple techniques using links and categories of the article might be employed to provide necessary amount of suitable related articles containing related links. The experimental results that will be detailed later also support our opinion. For this purpose, we have experimented and compared five different techniques. The common point for these five different approaches is employment of only links and categories to determine relatedness. The details of these approaches are given below.

4.1.1.1 Using Articles in Same Categories

As it was explained, the category structure of Wikipedia is an important semantic resource and the entities of the same type or entities with some thematic relationship are grouped into categories. This nature of categories clearly implies a relatedness between the articles belonging to them. Therefore, all articles which belong to one the categories of the target

article is accepted as candidate related article. For example, the article *Gülhane Park* (which is a park in historical Topkapı Palace in İstanbul) has the categories *Parks in İstanbul* and *Topkapı Palace*. Sample candidate related articles retrieved from these categories are given below:

- Other well-known parks in İstanbul like *Emirgan Park, Fethi Paşa Korusu, Yıldız Park* etc.
- Other historical places related to Topkapı Palace like *Imperial Harem, Procession Kiosk, Topkapi Palace, Palace Basilica* etc.

This approach might be problematic for very general categories with high number of articles, since the relatedness of articles would decrease because of the commonness of the topic. A solution for this problem might be the elimination of categories containing over a number of articles. Also categories for a weak relation type that do not stress a particular topic would be misleading. For example, the category *1970 births* connects the biographical articles for people born in 1970 but does not imply a direct topical relatedness. We expect that the filtering of related articles by the following step might mostly be able to eliminate this relatedness issues.

To increase the comprehensiveness of the technique, the subcategories of the actual categories are also considered. The subcategories of a Wikipedia category generally contains more specific articles on the topic. On the other hand, this feature might cause some topically differentiated, very specific articles to be selected. Additionally, we do not consider the parent categories since they contain more abstract, general articles. As mentioned above, the general categories are not expected to improve the performance of the system.

4.1.1.2 Articles Linked by the Target Article

This technique directly employs the relationship obtained by the links. All articles that are linked by the target article are accepted as candidate related articles. For example, the article *Gülhane Park* links to the articles like *Topkapı Palace*, *Procession Kiosk*, *İstanbul* and *urban park* which are related and possible to contain different related links. Since almost all of the linked articles are related to the target article, this approach seems reliable, but a possible

problem is the case of insufficient number of candidates when the target article has a small number of or even no links.

4.1.1.3 Articles Linking to the Target Article

This technique employs the reverse approach taken by previous one. Similar to the method in [7], the incoming links to the target article are considered. All articles that are linked to the target article are accepted as candidate related articles. For example, the article *Gülhane Park* is linked from articles like *Topkapı Palace*, *Yıldız* Park and *İstanbul Archaeology Museums* (a museum located near Gülhane Park).

The three techniques explained above are similar in using the specific set of relationships of an article which are categories, outgoing links and incoming links. These features of an article are usually employed in different tasks and we also aim to provide a comparison of the informativeness of them. The following two techniques differ from them in applying searches over the whole index by considering all articles in the collection. This approach has the chance to evaluate all possible articles instead of relying on a limited subset of articles. On the other hand, it seems more prone to select articles which may not be actually related since a direct relationship is not needed.

4.1.1.4 Index Query over Links

Instead of investigating a set of articles by direct incoming or outgoing link relationship, this technique searches the related articles from all index according to the commonness by outgoing links they share. As it was mentioned in Section 2.4, Lucene supports vector space query model to score matching results according to vector distance measures. For our query, we represent the target article with its title and the set of outgoing links it contains. The query is applied over the link field of the documents in the index. As a result, the outgoing link sets of target and candidate articles are used as the vectors to be compared. A sample query (shortened for readability) applied for the article *Gülhane Park* is like:

internal_link:“sarayburnu parkı”^4 OR internal_link:“gülhane parkı”^4 OR internal_link: “istanbul” OR internal_link:“topkapı sarayı” OR internal_link:“osmanlı imparatorluğu” OR internal_link:“istanbul boğazı” OR internal_link:“romalılar” OR internal_link:“bahçe”...

In English: *internal_link:“sarayburnu park”^4 OR internal_link:“gülhane park”^4 OR internal_link:“istanbul” OR internal_link:“topkapı palace” OR internal_link:“ottoman empire” OR internal_link:“bosphorus” OR internal_link:“ancient rome” OR internal_link:“garden”...*

As it is seen in the query, all main and redirect titles are included to the query and they are boosted by the factor of 4 which means the articles with a direct link to the target article are favored during scoring. After the query, best scoring results are selected as candidate related articles.

4.1.1.5 Index Query over Text

This technique uses basically a similar approach with the previous one with a difference in a single aspect. Instead of querying the documents in the index according to their links, it considers querying according to the text of the documents. As a result, the vector of the documents is the set of the words they contain. The target article is represented with the same way which is the title (with all redirections) boosted by 4 and the outgoing links. The example below shows that the only difference in the query is the field it was applied on:

text:“sarayburnu parkı”^4 OR text:“gülhane parkı”^4 OR text:“istanbul” OR text:“topkapı sarayı” OR text:“osmanlı imparatorluğu” OR text:“istanbul boğazı” OR text:“romalılar” OR text:“bahçe”...

4.1.2 Selecting Related Articles from Candidates

After collecting the candidate articles as explained above, each candidate is evaluated according to the number of links it shares with the target article. For this aim, we apply a simple relatedness measure over the articles according to their link overlap by inspiring from the study [49]. In that study, Adafre and de Rijke (2006) use this measure to detect similar sentences in articles from different languages. We thought it should be even more applicable for our problem, since the article scope contains many more links to obtain a better result than the sentence scope. Another reason for preferring this measure is that it lies parallel to our aim; if we are looking for semantic relatedness to other articles in order to find similar links, then the best measure might be similarity of sets of links. Similarity is measured using the Jaccard similarity [73] over links of articles. The Jaccard similarity is a simple similarity measure

between two sets and calculated by dividing the size of their intersection by the size of their union [74]:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{4.1}$$

This calculation is applied to score the relatedness between the target article and candidate article by using their set of links and can be formulated as:

$$score_c = \frac{shared_{tc}}{n_t + n_c - shared_{tc}} \tag{4.2}$$

where n_t is the number of links in the target article, n_c is the number of links in the candidate article and $shared_{tc}$ represents the number of shared links of the target and candidate articles.

After all candidates are scored, best scoring articles are accepted as the related articles of the target article. These articles are used as the sources to look up for link discoveries in the next step.

4.1.3 Searching for Discoveries

This step examines the non-link terms in the target article to find out new links by mathcing the linked terms in the related articles from the previous step. In addition to the related articles, the category pages of the target article are also used as another new link source. Since the category pages contain links to all articles in the category and they are topically related to the target article, these links are considered as candidates to insert into the target article. For example, the article *Enflasyon (Inflection)* is in the category *Finans (Finance)* and this category contains many links on this topic that would also occur in the text of the *Enflasyon* article.

For this step, initially, the text is tokenized to its words and these tokens are iterated to find matches with links from the sources explained. All matches occurring during this phase are accepted as a new link discovery. Since links point to article titles, they frequently occur as multiple words. So, not only single token matches are investigated, also matches of n-grams are checked. For example, for the text *Mustafa Kemal Atatürk* who is the founder of Republic

of Turkey, if we do not consider multiple word links, we may only detect *Mustafa* which links to an article about the name itself, which is unrelated to the context. If there are multiple matches of varying token numbers, the longest n-gram is preferred.

Since Turkish is an agglutinative language, it heavily uses suffixes for all inflection and derivation operations. With regard to link discovery, as a result of this situation, the inflections cover the stems of the possible links during matching with candidates, and this cause missing a considerable amount of matches. For example, the article *İstanbul* contains the term *kalkolitik çağ-da* meaning *at the Copper Age*. The meaning of *at* comes from the suffix *-da*. As a result, to be able to identify the link to the article *Kalkolitik Çağ* this suffix should be removed.

To deal with this problem, we have applied a stemming approach using the Zemberek library [75]. Zemberek is an open-source NLP library for Turkish and other Turkic languages. It has the capabilities like spell checking, morphological parsing, stemming, word construction, word suggestion, converting words written only using ASCII characters (with ASCII replacements of specific Turkish characters) and extracting syllables. It has an expendable architecture to support all Turkic or even agglutinative languages. It employs a dictionary of the roots and suffixes in the language. The roots are classified by their type like noun, verb etc. The suffixes are needed to be supplied with their special conditions and relative relations with other suffixes. Each suffix is identified with a name and they do not carry an information of their type like being inflectional or derivational. At run time, the library loads the roots into a Direct Acyclic Word Graph (DAWG) which is a kind of tree to provide fast search and access to all roots. The morphological parser involved in the library processes a given word to extract all possible root and suffixes of the word. It can also make the reverse of this process by composing a word in surface form for a given root and suffix set.

For our aim, the extraction of the root of the word by eliminating derivations was not a satisfying solution to check matching links. Firstly, as a general convention observed in Wikipedia links, a stem of a derived term is not tagged as a link, instead the whole derivation is used if it is actually related to the topic. Also, trying to link the stems increases the ambiguity by causing irrelevant terms to be selected as links. For example, for the word *birlik* which corresponds to the name of a troop unit, the stem is *bir* which has the meaning of the number *one*. This stem frequently occur in most of the articles and in most cases, it has no chance to be a relevant term to be a link. As a result, for link discovery, the removal of only the

inflections to find out matching links seems to be the optimum solution. First of all, since the article names correspond to the names of concepts only the noun type stems are considered. Since the Zemberek does not provide the type information of suffixes, there is not a clear separation between derivations and inflections. Therefore, from all alternative stems of a word, the longest noun stem is accepted as the candidate. This technique has met the requirements of the cases like *kalkolitik çağ-da* given above.

After applying this technique, we have examined sample results and observed that two exceptional cases are needed to be handled for covering the satisfying ratio of the matchings with stemming need. The first one was the terms having the suffix called as *ISIM_ILGI_CI* in Zemberek which takes forms of *-cı, -ci, cu, cü, -çı, -çi, -çu* and *-çü*. This is an agent suffix which makes nouns from nouns which has very similar meaning to the *-er* suffix in English. According to the implementation of Zemberek, the stems with this suffix are not retrieved as the longest nouns stems although it is a derivational suffix. This suffix occurs very frequently in our domain especially for biographical articles mentioning the professions of the person introduced. There are lots of exemplifying terms like *oyun-cu (actor), sanat-çı (artist), gazate-ci (journalist), futbol-cu (footballer)* etc. To cope with this situation, during stemming, it is checked if the morphological analyze of the word contains this suffix, then a term is generated using Zemberek by appending this suffix and its predecessors to the suggested stem, and this term is accepted as the longest noun stem of the word. As a result, for example, for the occurrences like *oyun-cu-nun*, the longest noun stem is accepted as *oyuncu* instead of *oyun*. This situtation might occur for different kinds of suffixes handled by Zemberek, but by examining sample results we have observed that, the gain that might be obtained by a deep analysis of all other suffixes would not be considerable. Therefore, similar occurrences with other suffixes are ignored.

The second exceptional case is multiple word article names occurring as compound noun clauses. The clause, *Avrupa yakasında* occurring in the article *İstanbul* might be used as an example. *Avrupa yakası* which corresponds to an article means *European side* and the second term takes an inflection suffix called *ISIM_TAMLAMA_I* in Zemberek and can be morphologically parsed as *yaka-(s)ı* (*s* is needed to connect vowels). This suffix might have the forms *-ı, -i, -u, -ü* and it is needed in most of the noun clauses to establish a relationship with the preceding term of the clause. For matching a link to the article for *Avrupa yaka-(s)ı-(n)da* terms, a similar approach with the previous case is taken. Firstly the last term of the clause

is stemmed and it is checked if it contains the *ISIM_TAMLAMA_I* suffix. If it is met, the stem of the word is appended with this suffix and its predecessors to generate the term *yakası*. This situation occurs frequently because many of the multiple word titles are composed of a noun clause like *Dünya Voleybol Şampiyona-(s)ı (Volleyball World Championship), Osmanlı İmparatorluğ-u (Ottoman Empire)* and *Topkapı Saray-ı (Topkapı Palace)*.

4.2 Discovering Links from Article Titles

In the previous section, we have explained our approach to find new related links from related articles. The second method we have explored propeses a different approach that originating from the study [6] explained in Section 2.3.1. Firstly, we have implemented the method described in [6] on Turkish Wikipedia. We have used this implementation for comparison with the previous approach and also to examine the weaknesses and determine our contributions.

Briefly to remind, this method uses the titles of all Wikipedia articles as candidate links if they occur in the text of the target article. The checks for matching links are exactly same as explained in Section 4.1.3. Since this implementation needed to traverse all titles in Wikipedia for all terms in the target article text, we had to reconsider the performance issues. The index structures explained did not satisfy the requirements and as a result we have created another file based cache structure that do not use Lucene. On startup, this file is loaded once and accessed during all operations. The data structure of this index is exemplified in Figure 4.1. Shortly, a hash structure is employed which uses the first words of the title as keys to access the list of titles starting with this word. During the iteration of the tokens in the text, for the each token, the list of titles starting with it are retrieved and checked for a match. Additionally, the tokens are also stemmed with the approach explained above and titles starting with them are also examined. For multiple word titles the following tokens in the text are examined. By this way, the need for traversing all titles of Wikipedia is eliminated and the performance is optimized to a satisfying degree.

The second step of this method is eliminating the candidate discoveries that are not related to the topic of the target article. For this aim, a scoring based on a keyword extraction technique is applied to the candidates. Since [6] shows that the keyphraseness is the best performing technique, we have implemented this technique. As it was explained, this technique measures

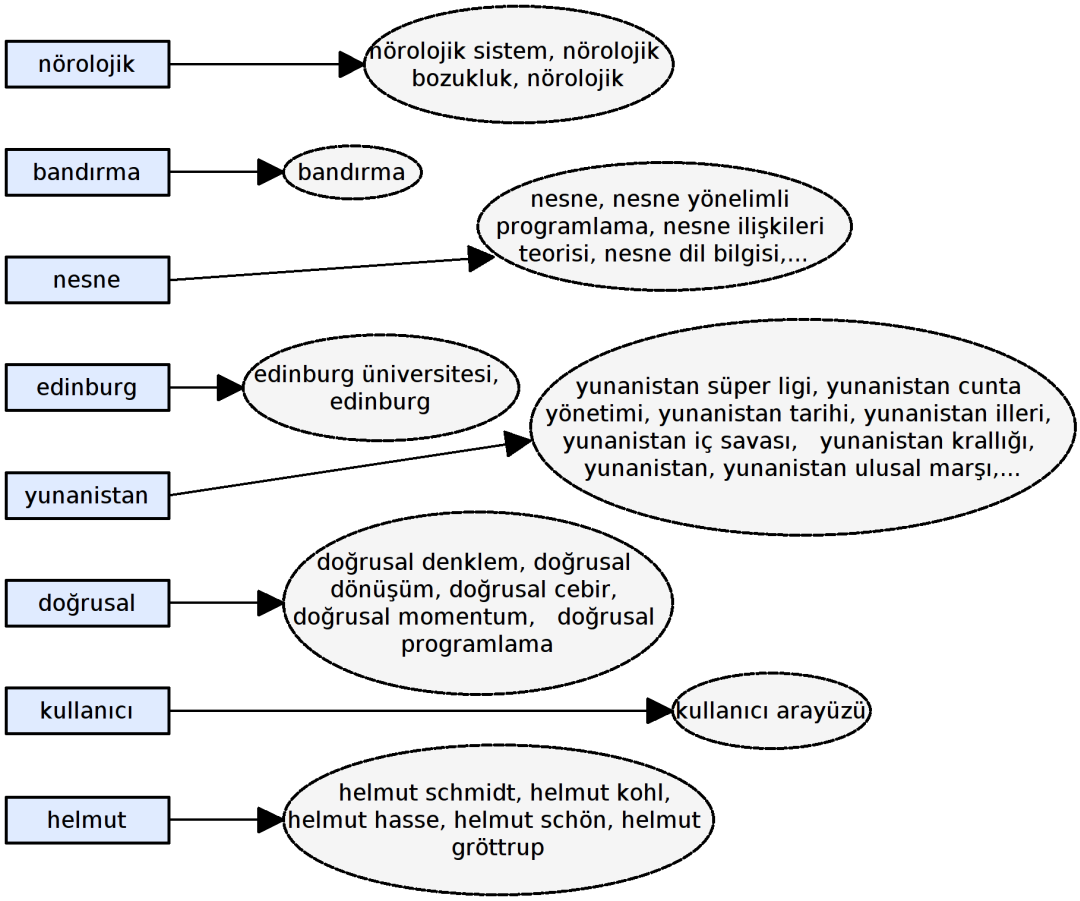

Figure 4.1: Sample records from the hash structure for accessing article titles

the probability of a term being used as a link and calculated by the ratio of occurrences as a link over all occurrences in the collection. As a result, if it is used as a link in most of the articles, it is also accepted as a link for the target article. From here, we will call this technique as *linkness filter* since it expresses our domain and usage aim better.

This method selects the best scoring 6% of the candidate discoveries as valid discoveries which corresponds to the average ratio of links over all terms in the articles of the collection. This approachs is a restricting one that discards the variability between articles. Instead of using this approach, we decided to determine a threshold as a validity limit for the scores to evaluate each candidate independent from the others. As threshold, we have determined the value which maximizes the results of our experimental evaluations as will be explained later.

Another phase in this step is the removal of identical link discoveries with different terms

which corresponds to the redirections of same articles, including the ones pointing to the target article. Because, most of the time, alternative phrases for the same concept and especially links to the target article are not marked as links in Wikipedia.

4.3 Adding Contextual Information to the Linkness Filter

Applying the linkness filter seems to us as a natural way that mimics to the behaviour of humans up to some extent. For example, for a fresh Wikipedia author, a mostly applied way to decide linking to a term is referring to the other similar articles to check whether this term is generally used as a link. But according to us, the problem of this approach is the lack of considering the contextual similarity of the articles on which this check is performed. Consecutively, it only takes into account the general usage of the term, instead of specific relation with the article context. As a result, if a term is an important concept for the target article, but is not frequently linked in other articles, it loses its chance to be selected. For example, the term *yeşil (green)* is not mostly used as link (75 times in 2010 occurrences), but when it is used in article *gökkuşağı (rainbow)* it should be marked as a link (as it is in the current version of Turkish Wikipedia) because it is specifically used in a context that is related to colors. On the other hand, from the opposite direction, a term which is a key concept and is linked frequently by many articles, may be irrelevant for a specific article. For example, the term *tür (kind, species, type)* frequently occurs as a link (in 1303 of 7126 articles occurred) because of being a key term in articles about organisms and biology. But for the use of it in the article *Çankırı (a city in Turkey)*, it has the general meaning of *type* and should not be linked.

The solution of this problem lies in applying the linkness probability evaluation over a set of contextually related articles to the target article, instead of all articles that the term occurs. By this way, the usage of the term can be examined specifically on the domain it occurs and if it is a key concept on the domain, it could more clearly come into prominence. For example, if an occurrence of the term *tür* in a biological article is being evaluated, since it is a key concept in this domain, considering only biological articles would cause more probability for it to be marked as a link.

To introduce contextual information to the linkness filter, we have decided to expand the

query applied for calculating the linkness using contextual terms from the article. Below, the queries of linkness filter for determining consecutively, the number of articles in which the term occurs and the number of articles in which the term is linked are exampled:

text:tür (7126 results)
internal_link:tür (1303 results)

For example, we can expand both queries to constraint the domain strictly to the biological articles by adding *biyoloji (biology)* term and the queries turn to:

text:tür AND text:biyoloji (154 results)
internal_link:tür AND text:biyoloji (74 results)

The results show that constraining with domain knowledge have increased the probability of the linkness of the term *tür* up to about 0.5 from about 0.2.

The challenging part about expanding the query is determination of the terms for applying the contextual constraints. If the number and the generality of these terms is high, this would cause a broad set and the query can not represent the context with the appropriate scope. For example using all of the terms or all of the links in the target article would enlarge the query very much and would cause losing the contextual focus. On the hand, if the number and generality of these terms is very low, it might cause the query not to represent the context sufficiently and retrieve only a limited set. For the *tür* example, the term *biology* is not singly sufficient to determine whole context and the number of results could not cover all articles for the context. Therefore, a possible solution we have considered which uses the article title as a single contextual term could not suffice our requirements. As a result of our exploration for a balanced solution, the linked terms in the descriptional beginning part of the articles have seemed us as a promising source to provide a concise representation of the article context.

The leading section of most of the Wikipedia articles are conventionally summarizing parts that include definition and high level introduction to the topic. It is placed at the beginning before the first heading or table of contents. The *Lead Section* part of the Wikipedia style guidelines [76] specifically determines the considerations and the style of this section, a brief description from the guideline:

> The lead should be able to stand alone as a concise overview of the article. It should define the topic, establish context, explain why the subject is interesting or notable, and summarize the most important points-including any notable controversies. The emphasis given to material in the lead should roughly reflect its importance to the topic, according to reliable, published sources, and the notability of the article's subject should usually be established in the first sentence.

The first sentence of a leading section is also a specific sentence that usually gives a definition of the essential concept focused in the article. The guideline states that, this sentence should be a short declaration sentence introducing the subject of the article and why it is notable. The first sentences usually contain important links to tightly relevant concepts occurring in the definition. Also the name (mostly the title) and synonyms of the concept explained in the article are stressed in the first sentence and in markup, they are expressed by surrounding with three inverted commas (''). An example leading section is shown in Figure 4.2 from the article *Voltaire*. As it is seen, actual name and the pen name of the author are stressed. In general, according to the detail level of the article, the lead section might be composed of only the first sentence, only a single paragraph or a few paragraphs.

In literature, there are studies employing first sentence or leading section sentences of Wikipedia articles. For named entity recognition, [77], [78] and [79] extract the category of a term from the first sentence of corresponding article. [80] exploits the definitions in first sentences for extracting subject-predicate-object triples from Wikipedia. Similarly, [81] extracts the hypernymy (is-a) relationship between concepts. The common feature of all these studies is employment of NLP techniques for syntactic analysis of sentences.

We have observed that, the features of the leading sections of articles were conforming to our needs by providing basic and directly context related information. For our purpose, instead of a deep syntactic analysis of the sentences, we decided to use directly the links and stressed words in the section. Because as mentioned before, according to the editing guidelines the links should appear at the most leading position possible. Also it is allowed only to add only related concepts with the article. As a result the most important concepts in the leading section are also marked as links. Also the stressed words are the key terms to represent the actual article concept. In further explanations, these links and stressed words will be called as context terms.

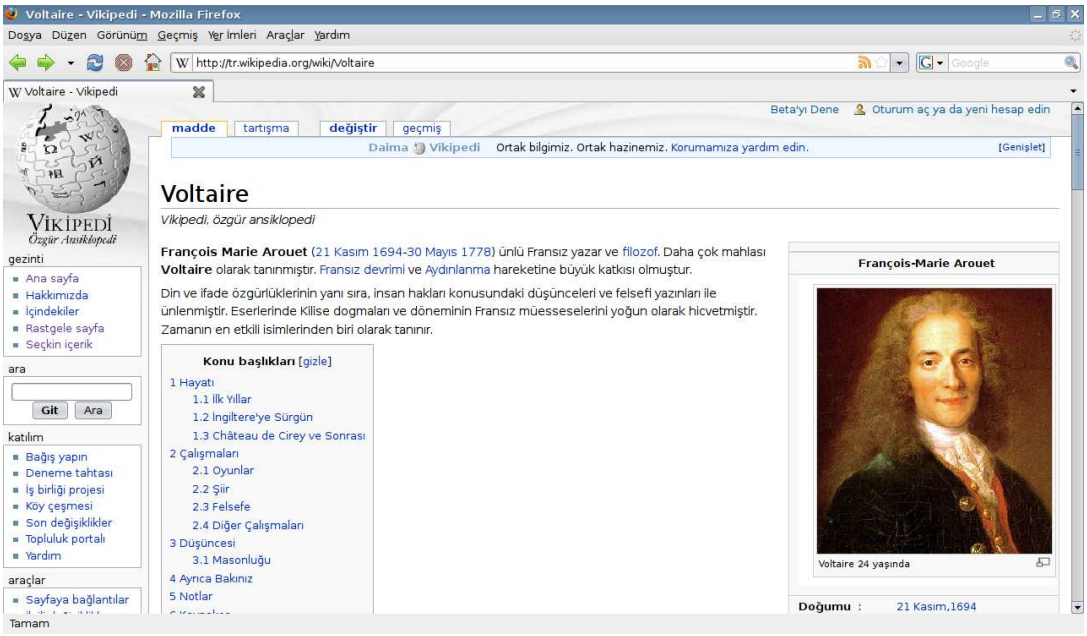

Figure 4.2: The leading section of the article *Voltaire* in Turkish Wikipedia

Consequently, using the context terms, we have modified the linkness filter by expanding the query to include contextual information from the target article, this filter will be called as *contextual linkness filter*. When multiple terms are used to extend the query, which is different than the *biyologi* example above, a decision for the structure of the Boolean expression to connect these terms is required. We have experienced that if all terms are connected with *AND* expression the resulting query becomes too constraining and reduces the number of results to inapplicable level. Therefore, the contextual terms are connected with OR expressions among themselves and this group is combined to the link candidate with AND. Consequently, if an article contains the term together with at least one of the context terms, this article is added to the query results. Below, the markup of the leading sentences with context terms from the article *Voltaire* is given:

"'François Marie Arouet"' ([[21 Kasım]] [[1694]]-[[30 Mayıs]] [[1778]]) ünlü Fransız yazar ve [[filozof]]. Daha çok mahlası "'Voltaire"' olarak tanınmıştır. [[Fransız devrimi]] ve [[Aydınlanma]] hareketine büyük katkısı olmuştur.

("'François Marie Arouet"' ([[November 21]], [[1694]]-[[May 30]], [[1778]]) was a famous French writer and [[philosopher]]. He is better known with his pen name "'Voltaire"'. He had important contributions to the [[French Revolution]] and [[Enlightenment]] move-

ments.)

The context words in the sentences are *François Marie Arouet, 21 Kasım, 1694, 30 Mayıs, 1778, filozof, Voltaire, Fransız devrimi* and *Aydınlanma*. As it was explained in the previous chapter, since the date articles are general articles without any specific context, our system ignores all date links which are *21 Kasım, 1694, 30 Mayıs* and *1778* in this example. Then, for example, if the term *demokrasi (democracy)* is a candidate link to be evaluated with the contextual linkness filter, the queries applied to find occurrence numbers as text and link are consecutively:

text:"demokrasi" AND (text:"François Marie Arouet" OR text:"filozof" OR text:"Voltaire" OR text:"Fransız devrimi" OR text:"Aydınlanma") (101 results)

internal_link:"demokrasi" AND (text:"François Marie Arouet" OR text:"filozof" OR text: "Voltaire" OR text:"Fransız devrimi" OR text:"Aydınlanma") (21 results)

The term democracy might occur in various kinds of topics but this article is a more specifically related one because it gives the biography of an important person for the process of the establishment of modern democracies. The number of results for same term with linkness filter are consecutively 1959 and 134 and accordingly, the contextual linkness decreases the number of results by the constraining them with contextual knowledge. As a result, the linkness ratio is multiplied with three and increases from 0.07 to 0.21.

As mentioned before, the balance of the broadness of the expanded query representation is an important issue. Therefore, to be able to evaluate, we have implemented two alternatives which employs the context terms in only the first sentence and in the first paragraph of the leading section. Detailed descriptions and comparisons of the alternatives will be given in Chapter 5. Also, another policy is applied which constrains that if the number of results for the second query is below 5, the ordinary linkness measure is preferred to ensure the reliability of the score.

4.4 Filtering the Discoveries from Related Articles

Two methods we have explained have different characteristics by their approach to the relatedness verification of candidate link discoveries. One of them selects candidates from related articles and other one employs a filtering approach. The former one is a convincing solution but since it does not evaluate the candidates specifically, this general solution seems very close to allow some irrelevant discoveries since they come from a related article. As a result, it can not guarantee a high amount of accuracy. On the other hand, the latter approach applies a specific evaluation for each discovery candidate. But, though it seems to be more accurate for the decision of relevancy, since this method accepts all matching article titles as a candidate there are much more candidates to evaluate compared to the former method. This could also increase the chance to allow some irrelevant candidates to be accepted by weaknesses of the filtering approach.

With the observation of the contrast in strengths and weaknesses of two approaches, we have decided to apply a combination of both methods to mutually eliminate weaknesses of each other. As a result, we have decided to select the candidate discoveries from related articles in first step. Then at the second step, the filtering mechanism from the second method is applied to detect irrelevant discovery candidates from the related articles. By this way, both the number of candidates could be controlled and also all candidates retrieved are specifically checked for appropriateness.

A single apparent problem of this approach would be the decrease in the number of discoveries because of the two phase elimination. We have anticipated to eliminate this consequence by employing satisfactory number of related articles employed in the first step.

CHAPTER 5

EXPERIMENTAL RESULTS AND EVALUATIONS

In this chapter, details of our experiments for comparative evaluation of the methods applied are given. Firstly, the evaluation metrics employed which are standard metrics for information retrieval will be introduced. Then, the overall performance of the best-performing configuration of each method will be examined according to the experimental results. It will be followed by the section which analyzes the effects of stemming application for discovery matching. The next section discusses the determination of threshold values for different methods. Then, experiments for determining the size of the first sections used for optimum contextual linkness filtering will be discussed. The last section will compare and discuss the results of related article retrieval techniques offered.

5.1 Evaluation Metrics

Comprehensive metrics for missing link discovery problem should comprise evaluation of both correctness and completeness of the suggestions. For this aim, we have applied well known statistical metrics that are widely applied for evaluating the studies on information retrieval [82].

5.1.1 Precision

Precision is a metric for measuring the accuracy of the results. For our task, it corresponds to the ratio of valid discoveries over all discoveries and can be formulated as:

$$Precision = \frac{|D_{valid}|}{|D_{total}|} \tag{5.1}$$

where D_{valid} is the set of discoveries marked as valid and D_{total} is the set of all discoveries made.

5.1.2 Recall

Recall is the metric for measuring completeness of the results. For our task, it corresponds to the ratio of valid discoveries over the number of all missing links that are expected to be discovered by the system and can be formulated as:

$$Recall = \frac{|D_{valid}|}{|D_{expected}|} \tag{5.2}$$

where D_{valid} is the set of discoveries marked as valid and $D_{expected}$ is the set of all discoveries expected to be made.

5.1.3 F-Measure

F-Measure is a combination of the precision and recall values that provides a single result by considering both of them. It is generally accepted as the mean of both measures, but since both precision and recall values are between 0 and 1, it is calculated by harmonic mean of them and can be formulated as:

$$F = \frac{2}{\frac{1}{precision} + \frac{1}{recall}} = \frac{2 * precision * recall}{precision + recall} \tag{5.3}$$

This metric can be applied in a weighed manner by favoring one of the precision or recall over the other. For this calculation the formula becomes to the generalized form of:

$$F_\beta = \frac{(1 + \beta^2) * precision * recall}{(\beta^2 * precision) + recall} \tag{5.4}$$

In this formula, the β value determines the weight of recall over precision. For example, F_2 measures recall twice as much as precision and $F_{0.5}$ measures precision twice as much as recall, which are both widely used metrics in information retrieval.

5.2 Evaluation of the Overall Performance

This section will give the evaluations of the overall performance for different methods we have suggested. For this aim, best parametric configurations for each method is employed.

The general approach for automatic evaluation of link discovery [6, 17] can be briefly summarized as:

- Remove all links from the test articles
- Execute the link discovery procedure
- Measure the recall and precision by comparing the discoveries with existing links before removal. Only the pre-existing links are accepted as valid discoveries.

This approach assumes the links in the current articles as a reference, the links in the articles must be the most accurate and complete set to satisfy the needs of the evaluation. In other words, this approach requires a reliable ground truth data set in which all articles contain the ideal number of relevant links. But, the determination of the ideal link set for an article needs human judgement and becomes a subjective approach. Also, for the missing link discovery task, since the aim is suggesting new links that are missing, usage of the current articles as reference is not applicable.

To be able to decrease the subjectivity in evaluations, we have applied a semi-automatic way. For measuring the recall of the system we have performed an automated approach. The precision is measured according to the manual evaluations by judgments over the discoveries of our system. Another issue increasing the objectivity is the comparisons between methods which is independet from the experimental setup. The experiments for measuring the recall and precision and the discussions will be detailed below.

During following explanations, for increasing the readability, we will prefer to call the method collecting discoveries from the related articles as *related articles* method, and the other method

collecting candidate from titles and filtering them as the *article titles* method.

5.2.1 Measuring Recall

Since recall represents the coverage of valid discoveries over missing links, for calculation of it, the set of the missing links to be discovered must be determined. This determination is hard because because of the human factor, it can not be guaranteed that all missing links are detected correctly and it would probably tend to substantially vary for different people. Another approach is, as explained before, the removal of all links from the article and checking the discoveries. But this approach mostly reflects the aim of link discovery instead of missing link discovery. Since our methods aim for missing link discovery and benefit from the existing links of the articles as a semantic resource, the efficiency will certainly be affected negatively by removal of all links.

As a consequence of these considerations, we have developed a different evaluation technique for calculation of recall. For a single article, a randomly selected link is removed and this link is selected as the expected missing link target. If this link is discovered by the system the recall for this article corresponds to 1.0, if it could not be discovered the recall is 0.0. If this process is applied to a high number of articles, the ratio of the successful articles with recall 1.0 is accepted as the recall of the overall system. By removing only a single link from an article, the efficiency of the system is affected by a very small degree. On the other hand, the most of these removed links are reliable as a missing link, because they have already been manually inserted before.

For employment in the experiments, 2000 articles with arbitrary (more than 0) number of links have been randomly selected from the index. Then, for each of these articles, a single link is randomly selected and removed from the article as target missing links. As a result, the ratio of discoveries of these 2000 candidates is accepted as the recall of the system.

5.2.2 Measuring Precision

Inserting a new link to an article is not an objective task even for humans, since decisions about the suitableness of the link may vary between people. Therefore, a fully automatized and objective precision evaluation have not seemed realizable to us. Since, we have decided to

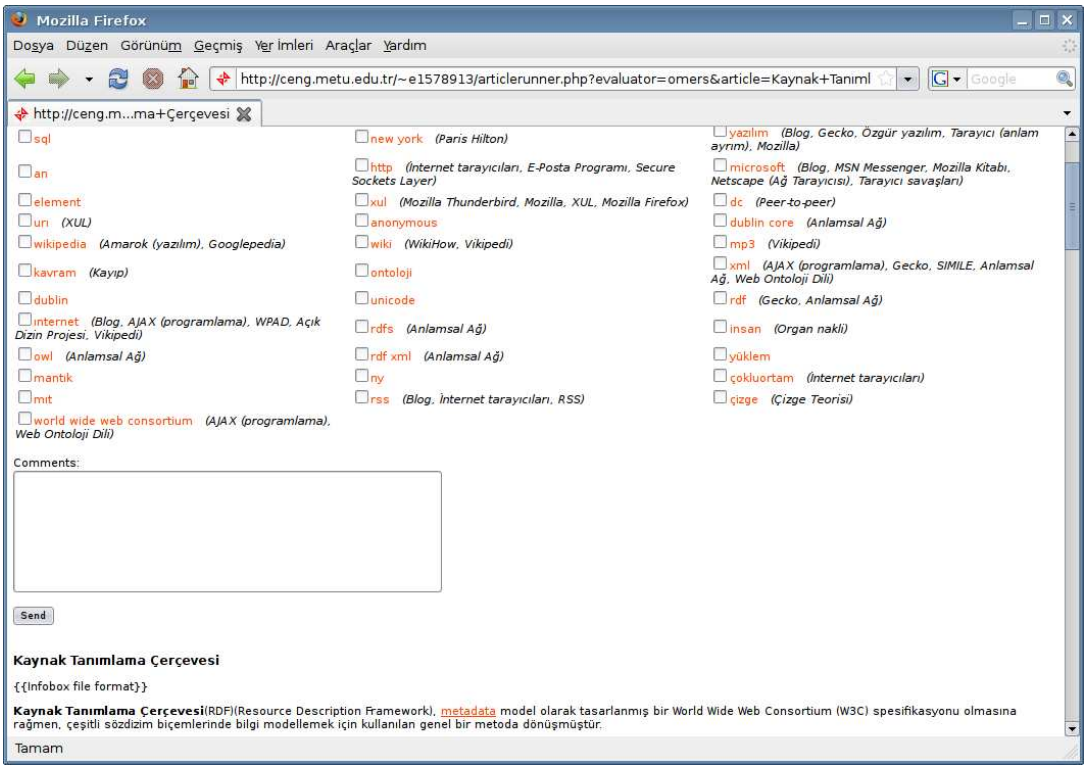

Figure 5.1: A section from the application for collecting the evaluations of assessors

collect statistical information from various number of human assessors who are experienced Internet (and Wikipedia) users with graduate level education. We have also tried to contact very experienced Wikipedia authors but could retrieve evaluations from one of them.

We have developed a user interface for the evaluations that represents the discoveries with a different color at the first occurrence location in the article text. The discoveries are also listed above the text with checkboxes to select the valid discoveries. For the links discovered from related articles, we have also showed some of these related articles to be examined as a reference for their decision. Also, a comment field is provided to allow them to mention their specific comments about their preferences, by this way we could resolve the misunderstandings. The Figure 5.1 shows a sample evaluation form. We applied the different link discovery approaches and collected almost all possible discoveries that might be suggested by any of them, for presenting to the evaluators. The assessors were informed about the general principles of linking in Wikipedia and asked to evaluate according to these principles, their insight to see the suggestions as a link and comparison with the linking approaches in similar articles.

As a result, by using this application, 6 assessors have evaluated totally 89 different articles covering approximately 700 different link suggestions that might be discovered by any of the approaches.

5.2.3 Results and Discussions

Table 5.1 gives the evaluation results of the methods we have applied. Figure 5.2 presents the graph for these results. The approach that collects discoveries from the related articles have been applied solely (as in [7]), and also it has been experimented it along with linkness and contextual linkness filterings. The second approach collecting discoveries from the article titles employs the linkness filtering as suggested in [6] and the contextual linkness filtering.

Table 5.1: Results for the missing link discovery methods applied *(R.A.: discovery from related articles, A.T.: discovery from article titles)*

Method	Precision	Recall	F-Measure	F_2-Measure
R.A. / no filter	69.3	85.4	76.5	81.5
R.A. / linkness filter	**86.1**	80.5	83.2	81.6
R.A. / contextual linkness filter	85.8	83.1	84.4	83.6
A.T. / linkess filter	78.6	84.2	81.3	83.0
A.T. / contextual linkness filter	81.2	**88.6**	**84.7**	**87.0**

The best results according to F-Measure, with a very small difference, are gained by the approach that performs contextual linkness filtering over discoveries from article titles. Especially, its high recall value, despite relatively low precision brings its success. On the other hand, the results for application of filtering over discoveries from related articles are successful by precision and they also provide more balanced precision and recall values compared to the former one. Additionally, the results for performing linkness filtering over both methods show that the related articles method outperforms the article titles method because of the increasing gap between the precision values.

The results show that the approach collecting discoveries from article titles could not provide balanced results by recall and precision metrics. Although it provides a high amount of discoveries, the accuracy of them is relatively low. This points to the fact that if the application

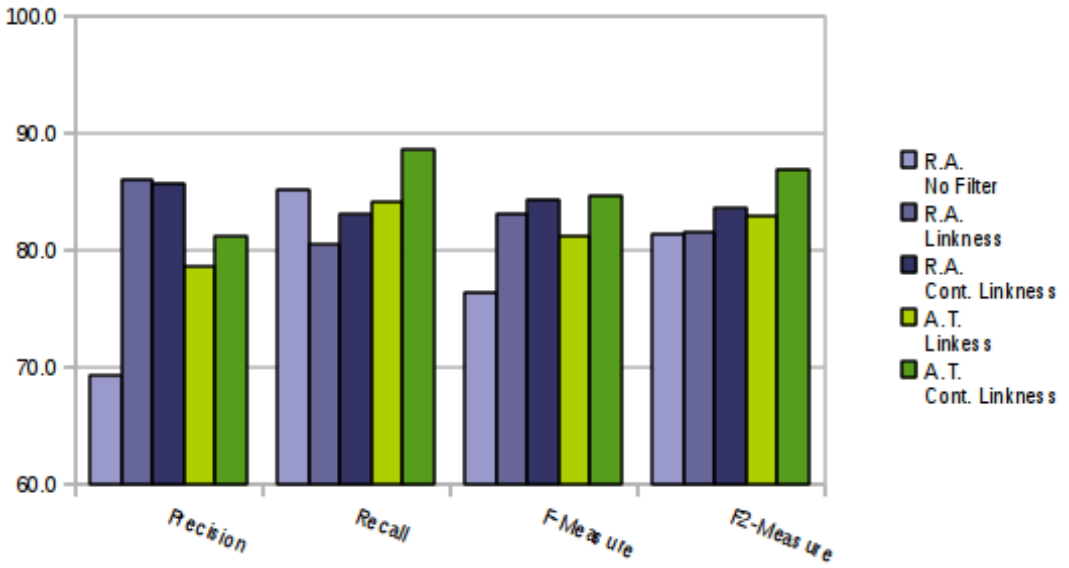

Figure 5.2: Result graph for the missing link discovery methods applied *(R.A.: discovery from related articles, A.T.: discovery from article titles)*

needs high accuracy, filtering might not meet this requirement singly.

Filtering brings a serious increase in the achievement of the related articles method. Especially, the precision increases and comes to a balanced point with relatively small decrease in recall. The increase gained from the filtering confirms our proposal that all links brought from related articles can not be directly accepted as relevant discoveries and a discovery specific check is needed to ensure its accuracy. On the other hand, this might also be interpreted as an indicator of a need for more precise selection of related articles, but it would probably cause unsatisfactory number of discoveries and consequently a sharper decrease of recall.

Another inference from the experiment results is the considerable improvement obtained by the contextual linkness filter over linkness filter and it is the preferred filtering mechanism for both methods. It increases both the precision and the recall values except the small decrease of the precision of related articles method. It is especially more efficient for its application to the discovery from article titles compared to the other approach. This situation originates from the already existing relatedness filtering that is implicitly applied by the latter one. But when the contextual linkness is applied as the single filtering mechanism as in the former one, it significantly outperforms the linkness filter by improving both the precision and recall.

From a different point of view, the application environment might also affect the method to

prefer. If the accuracy of the discoveries is more important for the application than the number of them, the related article approach might be preferred. For a batch process application that applies the discoveries on the background, this method would be preferred. The so called *bots* in Wikipedia correspond to this kind of application. But when it is deployed as a recommendation system that suggests users to insert the discoveries, the high number of alternatives might be more preferable instead of high accuracy because of the human-controlled process. For checking the validity of this proposal, we have also examined the F_2-Measure results of the experiment which are given at the last column of the Table 5.1. As it was explained, this measure favors the recall of the systems two times more than the precision. The results support our idea and the article title approach with contextual linkness filtering quitely outperforms other methods.

The precision result for the related articles approach without any filtering is interestingly very close to the result reported in the previous study [7] which was 0.68. Since the experimental setup is not same, direct comparison could not be accepted as a factual result. But, since the evaluation method is same by manually determining the validity of suggested discoveries, this closeness might indicate the parallelness of our implementation and evaluations. Also, it might show the success of our simple techniques for related article retrieval against the complex approach in that study.

5.2.4 Evaluation by Discovery Amount

This experiment analyzes the amount of links obtained by application of missing link discovery. By the results of the experiments, we aimed to comparatively show the scale of enrichment on the link structure by applying the methods proposed. On the other hand, since it primarily takes care about the amount of discoveries, this experiment could also serve to verify our proposal about employing the application as a recommendation system in previous section.

The results given are obtained from the recall experiment which performs discovery of new links for 2,000 different articles. Initially, these articles have been containing a total of 34,508 links, and this corresponds to about 17.3 links per article. Table 5.2 gives the results for each method. The first column a gives the number of total discoveries obtained by the method. But, considering solely the number of discoveries would be misleading about the performance of

Table 5.2: Experiment results according to the amount of discoveries

Method	#Discoveries	#Discoveries Normalized	Increase of links (%)
R.A. / no filter	**13,874**	**9,615**	**27.9**
R.A. / linkness filter	7,385	6,355	18.4
R.A. / contextual linkness filter	7,702	6,608	19.1
A.T. / linkness filter	10,469	8,229	23.4
A.T. / contextual linkness filter	10,524	8,545	24.8

the method since each of them has a different precision as explained in previous section. Therefore, we have applied a normalization over these numbers by calculating a projection of them according to these precision values. The second column gives the normalized discovery numbers which is obtained by multiplying the value in the first column with the precision of the method which was given in Table 5.1. The third column gives the total ratio of increase in the amount of links according to the normalized values. The results show that a considerable enrichment of the links might be gained by performing our missing link discovery approach.

The most remarkable finding from the results is the success of the methods with high recall values despite their relatively low precision and thus the F-measure values. This supports the idea of employing them as a recommender system. With about 25% percentage of enrichment, three high-recall methods discovers in average 4 valid links per article. In general distribution, this number would be especially higher for articles with very few links and would bring serious enrichment on their link coverage. Especially, the article titles method with contextual linkness filtering provides a competent enrichment amount besides its relatively good precision compared to the other high-recall methods.

5.3 Effect of Stemming

In Section 4.1.3, we have explained our approach for the application of stemming which aims resolving the discoveries that are embedded inside inflected words. We have performed an experiment to examine the effect of stemming. The results of the experiment are given in Table 5.3. These results showed us that applying stemming could not increase the overall success of the system, and we excluded it from the best configurations of the methods. The

Table 5.3: Effect of applying stemming *(the parenthesized values are without stemming from Section 5.2.3)*

Method	Precision		Recall		F-Measure		F_2-Measure	
R.A. / no filter	61.2	*(69.3)*	88.5	*(85.3)*	72.4	*(76.5)*	81.3	*(81.5)*
R.A. / linkness filter	**82.5**	***(86.1)***	83.2	*(80.5)*	82.8	*(83.2)*	83.0	*(81.6)*
R.A. / cont. linkness filter	81.7	*(85.8)*	85.9	*(83.1)*	**83.8**	*(84.4)*	85.1	*(83.6)*
A.T. / linkness filter	71.4	*(78.6)*	87.0	*(84.2)*	78.4	*(81.3)*	83.3	*(83.0)*
A.T. / cont. linkness filter	72.2	*(81.2)*	**91.6**	***(88.6)***	80.8	***(84.7)***	**86.9**	***(87.0)***

main impact of stemming, which is valid for all methods, is the sharp decrease of the precision value in contrast to the considerable increase of recall. The reason behind this result is simple, although the detection of embedded links increases the discovery amount, ambiguous results

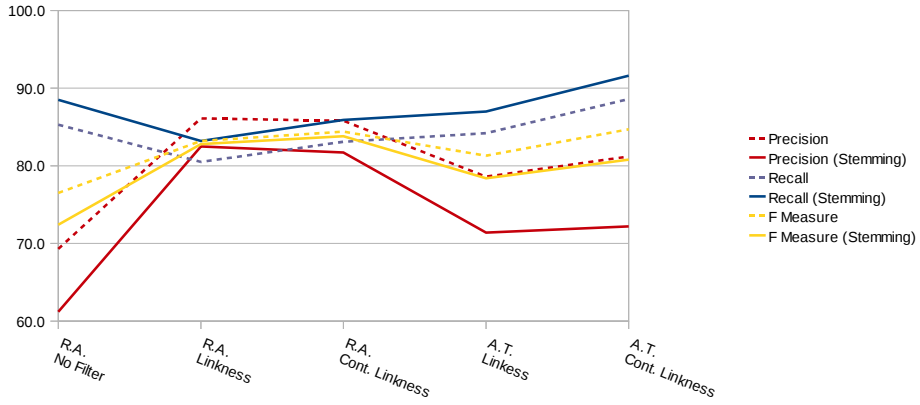

Figure 5.3: Effect of applying stemming graph

Figure 5.3 presents the graph for these results. It is observed that the change in results are parallel for each method. But with the application of stemming, results of the related articles method is better than the article titles method. Actually, we have expected that linkness and contextual linkness filtering could seriously eliminate the ambiguous discoveries cause by stemming, but it was not realized. For example, the approach using article titles employs only these filterings to ensure its precision. But this precision could not be preserved when the stemmed candidates are evaluated with these filterings. This result shows that discoveries from article titles before any filtering are more meaningful than the discoveries found by

stemming. Because, stemming cause discoveries of very irrelevant terms that could not even occur in the text of the article.

In spite of the explained deficiencies, because of the serious increase in recall, the results becomes competitive for F_2-Measure metrics. This shows that, if the application is employed as a recommender system, the decrease of the precision might still be discarded because of the gain by number of discoveries.

5.4 Determination of Threshold Values

In this section, we will present our experiments for determining the threshold values for linkness and contextual linkness filter scores to accept a discovery as a valid one. Since the characteristics of the related articles and article titles methods are different, we have separately experimented these approaches for both filters. As a general approach for the experiments, firstly an experiment with broad values are examined to detect the value roughly, then finer values are experimented to make the thresholds definite. For the experiments in this section and further ones, we have used a randomly selected 500 articles for the evaluation of recall because of execution time considerations. The precision is evaluated in exactly the same way.

The threshold values selected by these experiments are used for all other experiments as the best configuration. Since the main metric considered in other experiments is F-Measure, the main criterion for the selection was the success of value for this metric. For equalities of F-Measure results of different tresholds values, more balanced values by precision and recall metrics are preferred.

5.4.1 Linkness Filter Threshold for Related Articles Method

Table 5.4: Comparison of broad threshold values for linkness filter over related articles method

Threshold	Precision	Recall	F-Measure	F_2-Measure
0.10	86.1	**81.4**	**83.7**	**82.3**
0.20	89.4	75.6	81.9	78.0
0.30	92.5	70.6	80.1	74.1
0.40	**93.8**	65.6	77.2	69.8

Table 5.4 gives the results for broad threshold values for linkness filter applied to the related articles method. Especially because of very low recall and consequent decrease in F-Measure, values more than 0.2 do not seem acceptable and finer values are needed to be examined about 0.1 level.

Table 5.5: Comparison of fine threshold values for linkness filter over related articles method

Threshold	Precision	Recall	F-Measure	F_2-Measure
0.04	80.9	**85.6**	83.2	**84.6**
0.08	84.9	82.0	**83.4**	82.5
0.12	86.2	80.2	83.1	81.3
0.16	88.4	78.6	83.2	80.4
0.24	**92.1**	73.4	81.7	76.5

Table 5.5 gives the results for finer threshold values. It is observed that, for the values below 0.2 the F-Measure results are similar.

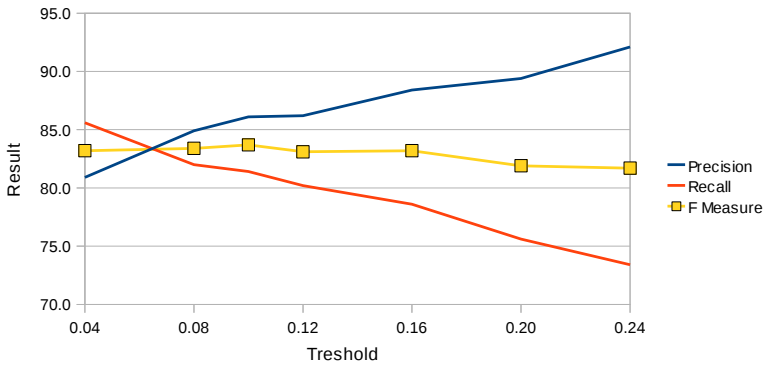

Figure 5.4: Graph for the results of candidate linkness filter threshold values for related articles method

The graph in Figure 5.4 shows the results for all experimented threshold values up to 0.24. The best performing value according to F-Measure metric is **0.1** and selected as the threshold value.

5.4.2 Linkness Filter Threshold for Article Titles Method

Table 5.6 gives the results for broad threshold values for linkness filter applied to the article titles method. As different from the previous experiment, results about 0.2 are also competitive to the 0.1 level. Therefore, threshold value should be searched about 0.1 and 0.2 values.

Table 5.6: Comparison of broad threshold values for linkness filter over article titles method

Threshold	Precision	Recall	F-Measure	F_2-Measure
0.10	74.7	**87.0**	80.4	**84.2**
0.20	80.6	80.4	**80.5**	80.4
0.30	**82.8**	74.2	78.3	75.8
0.40	82.3	68.4	74.7	70.8

Table 5.7 gives the results for finer threshold values. The success starts to decrease with the 0.24 value, therefore further values are not experimented.

Table 5.7: Comparison of fine threshold values for linkness filter over article titles method

Threshold	Precision	Recall	F-Measure	F_2-Measure
0.04	72.0	**91.6**	80.6	**86.9**
0.08	74.3	87.6	80.4	84.6
0.12	75.7	85.8	80.4	83.6
0.16	78.6	83.4	**80.9**	82.4
0.24	**82.2**	78.2	80.1	79.0

The graph in Figure 5.5 shows the results for all experimented threshold values up to 0.24. According to these results, **0.16** is selected as threshold by having best F-Measure and sufficiently balanced recall and precision.

5.4.3 Contextual Linkness Filter Threshold for Related Articles Method

Table 5.8 gives the results for broad threshold values for contextual linkness filter applied to the related articles method. This results show that the gap between precision and recall

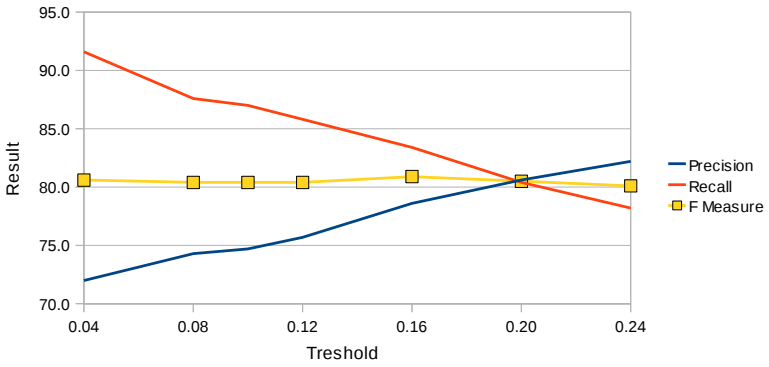

Figure 5.5: Graph for the results of candidate linkness filter threshold values for article titles method

increases and F-Measure decreases after 0.3 value. Therefore, the threshold is searched below this level.

Table 5.8: Comparison of broad threshold values for contextual linkness filter over related articles method

Threshold	Precision	Recall	F-Measure	F_2-Measure
0.10	85.8	**83.0**	**84.4**	**83.5**
0.20	88.5	80.2	84.1	81.7
0.30	92.6	77.0	84.1	79.6
0.40	**93.5**	74.5	83.0	77.7

Table 5.9 gives the results for finer threshold values. Since the F-Measure starts to decrease after 0.16 value, values further than 0.24 are not considered.

Table 5.9: Comparison of fine threshold values for contextual linkness filter over related articles method

Threshold	Precision	Recall	F-Measure	F_2-Measure
0.04	80.0	**84.6**	82.2	**83.6**
0.08	83.5	83.4	83.4	83.4
0.12	86.3	82.6	**84.4**	83.3
0.16	87.5	81.2	84.2	82.3
0.24	**90.7**	78.4	84.1	80.6

The graph in Figure 5.6 shows the results for all experimented threshold values up to 0.24. By F-Measure results both 0.1 and 0.12 are the best performing values. As the result, the **0.1** value is selected as the threshold since it gives more balanced precision and recall.

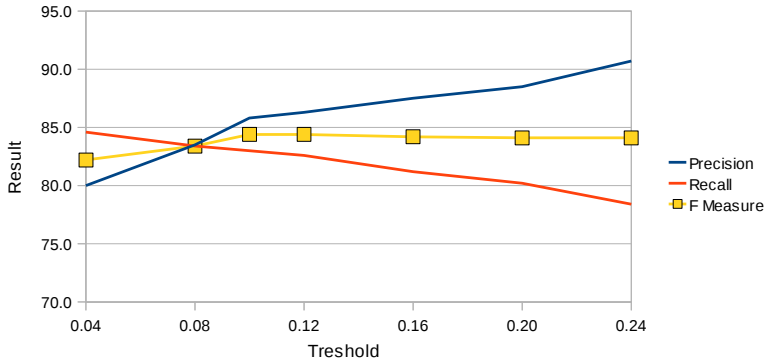

Figure 5.6: Graph for the results of candidate contextual linkness filter threshold values for related articles method

5.4.4 Contextual Linkness Filter Threshold for Article Titles Method

Table 5.10 gives the results for broad threshold values for contextual linkness filter applied to the article titles method. According to the F-Measure results, the best performing threshold should be searched about 0.1 and 0.2 levels.

Table 5.10: Comparison of broad threshold values for contextual linkness filter over article titles method

Threshold	Precision	Recall	F-Measure	F_2-Measure
0.10	78.7	**88.2**	83.2	**86.1**
0.20	81.9	85.0	**83.4**	84.3
0.30	**84.1**	81.4	82.7	81.9
0.40	**84.1**	78.8	71.3	79.8

Table 5.11 gives the results for finer threshold values up to 0.24 where the F-Measure results start to decrease.

Table 5.11: Comparison of fine threshold values for contextual linkness filter over article titles method

Threshold	Precision	Recall	F-Measure	F_2-Measure
0.04	73.9	**90.2**	81.3	**86.4**
0.08	77.1	88.6	82.4	86.0
0.12	79.2	87.8	83.2	85.9
0.16	81.2	86.2	**83.6**	85.1
0.24	**83.5**	83.2	83.3	73.2

The graph in Figure 5.7 shows the results for all experimented threshold values up to 0.24. According to these results, **0.16** is selected as threshold by having best F-Measure and sufficiently balanced recall and precision.

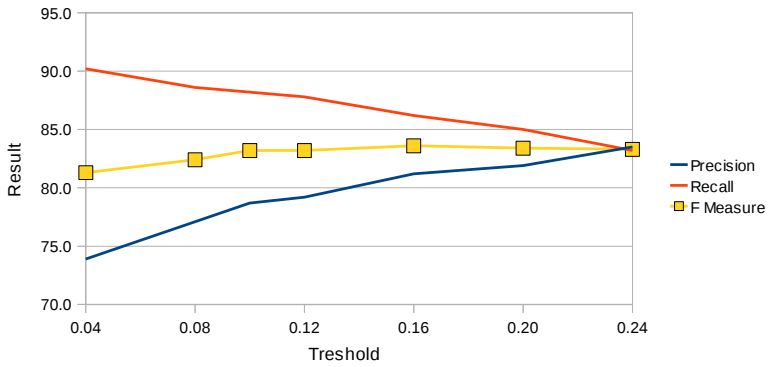

Figure 5.7: Graph for the results of candidate contextual linkness filter threshold values for article titles method

5.4.5 Discussion

Firstly, as it is expected, the higher threshold selection brings a more strict filtering and accordingly, the precision increases despite the decrease in recall. As it was mentioned before, the threshold values are selected according to the F-Measure metric. But the results of the experiments show that, both approaches and both filters can be configured to give better precision or better recall by decrease in the opposite side. Additionally, it is observed again that the related articles method is more suitable for high precision aim and article titles method is more suitable for high recall aim. As a result the configuration according to the nature of the

application should be established both selecting suitable method and thresholds.

Another observation is that the article titles method naturally requires more strict filtering with high threshold value. Threshold is lower for the related articles method which applies a primary filtering by employing only related articles. An interesting result is both linkness and contextual linkness thresholds are same for same methods which are 0.16 for article titles and 0.10 for related articles.

5.5 Determination of First Section Size for Contextual Linkness

We have initially applied the contextual linkness filtering by considering only the first sentences of the articles. The idea behind this was the special descriptional characteristic of this sentence. We assumed that, this feature would provide the focused information about the context. As it was mentioned in Section 4.3, if the number of the query results of contextual linkness is not satisfactory (less than 5), not to allow misleading results, the linkness filter is preferred instead. During observing filtering results for sample articles, we have noticed that a serious portion of the candidates were evaluated by linkness filtering instead of contextual linkness. By examining those samples, the fact occurred that a serious number of articles in Turkish Wikipedia did not contain a proper first sentence with enough linking. Therefore, we have decided to employ a broader part from the first section of the article to obtain more links about the context. To solve this issue, employment of complete first section was a candidate. But similarly, a serious number of Turkish Wikipedia articles do not have a clear separation between first section and the rest especially since their content is relatively small. As a result we decided to try employing only the first paragraph of the first section which is clearly separated from the rest. For checking the success of this approach we have applied another experiment. Table 5.12 gives the result of the experiment which compares the usage of first sentence and first paragraph.

According to the results, employment of first paragraphs outperforms employment of first sentence for all of the metrics and both methods. Especially, the precision of the results are considerably increased. As a result, the best configurations in other experiments use first paragraphs. In fact, the improvement obtained also shows the success of contextual linkness

Table 5.12: Results for different first section units of contextual linkness filter*(R.A.: discovery from related articles, A.T.: discovery from article titles)*

Method	First Section	Precision	Recall	F-Measure	F_2-Measure
Related Articles	Sentence	82.7	83.4	83.0	83.2
	Paragraph	**85.8**	**83.8**	**84.7**	**84.2**
Article Titles	Sentence	79.6	86.0	82.6	84.6
	Paragraph	**81.2**	**86.2**	**83.6**	**85.1**

filter over the linkness filter. Because, this change aimed to increase the number of articles that are evaluated by contextual linkness filtering and this preference of contextual linkness over linkness filter has improved the results.

5.6 Evaluation of Related Article Retrieval Techniques

As it was explained in Section 4.1.1, we have applied five different techniques for retrieving related articles for the related articles method. Our aim for applying these techniques was the proposal of simple and effective approaches for related article retrieval against the complex method of LTRank algorithm [7]. Aditionally, by performing different kind of techniques, we have aimed to examine the relatedness feedback obtained from different Wikipedia relationships like categories, incoming and outgoing links of an article.

Table 5.13 gives a comparison of results of these techniques with the order of explanation in that section. For each technique the alternatives of no filtering, linkness filtering and contextual linkness filtering have been experimented to notice possible correlations of techniques and filtering methods.

The results of the techniques can be separated into two main groups which are obviously parallel to the characteristic features of the techniques. The first three techniques explore related articles with a direct link or category relationship. For this group, the results seem satisfactory in terms of precision but the recall of these techniques is very low. On the other hand, techniques in the second group apply an index search on whole articles depending on the overlap with the links of the target article. For this group, while precision is preserved, the recall increases to the same level.

Table 5.13: Comparison of related article retrieval techniques applied

Method	Filter	Precision	Recall	F-Measure
Same category	None	78.2	**38.4**	51.5
	Linkness	89.9	34.8	50.2
	Cont. Linkness	***92.0***	35.8	**51.5**
Linked by target	None	70.0	**39.4**	50.4
	Linkness	87.0	37.0	51.9
	Cont. Linkness	**89.2**	37.2	**52.5**
Linked to target	None	65.2	**34.4**	45.0
	Linkness	81.7	31.4	45.4
	Cont. Linkness	**87.9**	31.8	**46.7**
Link based query over links	None	65.8	**83.8**	73.7
	Linkness	84.7	78.2	81.3
	Cont. Linkness	**88.4**	78.6	**83.2**
Link based query over text	None	69.6	**84.0**	80.6
	Linkness	84.9	78.4	79.6
	Cont. Linkness	**86.6**	79.2	**82.7**
All combined	None	69.3	***87.2***	77.2
	Linkness	**86.1**	81.4	83.7
	Cont. Linkness	85.8	83.8	***84.7***

The difference of recall for the two groups can be associated with the number of articles they return. Techniques in the first group evaluate a small set of articles because direct relationship constraints, although the ones in the second group make the selection from a broader range of articles. On the other hand, for second group, preservation of the precision in spite of increase in article number indicates the success of related article scoring applied.

An appreciable result is the precision obtained by the articles collected from same categories. Although its recall is not satisfactory because of insufficient number of articles, it provides the best precision values for all filter types. This can be interpreted as a clue to expose the fact that category relationship might be the most important semantic information that reveals the relatedness of articles.

Another interesting result is the parallelness of the behaviour of second and third approaches. These two techniques exploit, consecutively, the outgoing links from the target article and the incoming links to the target article. The similarity of results might indicate the sets of incoming and outgoing links of an article are highly intersects. Therefore, it may point to the article clusters in Wikipedia article space, which are classified by having high density of

interconnections by means of links.

The results show that variety and number of articles considered increase the success. Therefore, as a most comprehensive alternative, we experimented with combining all these related article retrieval techniques and the system achieved the best results as it is listed in the last part of Table 5.13.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

In this thesis, we explained our study on missing link discovery task over Turkish Wikipedia. Exploiting from the structured semantic relationships and up-to-date world knowledge in Wikipedia is increasingly being subject to studies. The link discovery for Wikipedia articles is one of the important sub-fields that very recent studies are focusing on. As a result of our experience from this study, we can propose two reasons for the importance of the task. The first one is its applicability to solve a real-life problem of users involving both the readers and authors of the articles. Secondly, the approaches employed for this task are common with the studies using Wikipedia specific structures as a semantic resource. Therefore, the approaches for link discovery problem might affect and also be affected by the approaches in other studies. Our approach for improving the older methods was similar. We have investigated two older studies and observed their advantages and disadvantages to combine them to resolve their deficiencies. We have also proposed the contextual linkness filtering by inspiring from studies on different tasks employing Wikipedia. We have performed a comparative evaluation approach to show the efficiencies and deficiencies of the approaches. By this way, the objectiveness of the evaluations was ensured, since especially the precision experiments were using manual assessment of human evaluators and might be misleading.

The comparisons of different approaches in the evaluation showed that the variety of methods provide alternative opportunities to select different style of applications. We have shown that some approaches provide results with relatively high accuracy. By configuring the threshold values to force maximizing the accuracy of the system, it can be employed as a batch application which periodically scans all articles in Wikipedia to automatically detect and insert missing links. Since all change history of the articles can be viewed by the users, the insertions might be easily tracked. On the other hand, the approaches which provide high

recall and results with large amount of discoveries are candidate to be deployed as an online recommendation system. It could serve to both readers and authors, because all readers have right to edit the content. Allowing readers to determine link insertions would result with high feedback amount. For this usage, the threshold values might be configured to maximize the recall of the system to increase number of discoveries. Because, the decrease in the accuracy of the system might be ignored since the actual insertions of the missing links are applied with the approval of a user.

For our system, the execution time was a critical consideration which has frequently interrupted our focus on the problem. In Section 2.4 and Chapter 3 we have detailed the general and specific solutions for accessing the large textual content of Wikipedia. We hope these explanations might be benefited by other researchers for the further studies on the area without same interruptions.

As another conclusion, we are pleased to obtain promising results from Turkish Wikipedia by providing satisfactory amount of semantic information to support our solution. On the other hand, except the part which employs stemming, the methods applied are independent from language and be employed for any of the Wikipedia language versions. The ignorance of the stemming would not be a problem because of two reasons. Firstly, it was shown that the overall performance decreases when the stemming has been applied for matching the discoveries in the article text. Cause of this situation is the serious amount of ambiguous results which are irrelevant with the topic. But it might be preferred when the recall is favored over the precision, because it provides more number of discoveries although the accuracy is not very high. The second reason for the ignorance of stemming is less necessity of it for most of the languages like English, because they relatively contain less inflections than Turkish.

Wikipedia allows the cross-lingual matching of articles which results to a rich multilingual resource. Exploiting the cross-lingual link structure of Wikipedia seems as a promising approach for various kind of multilingual tasks. But, it might also be benefited to improve the missing link discoveries by considering the existing links in other language versions of the articles. Small instances like Turkish Wikipedia might especially benefit seriously from the more elaborate ones like English version, by terms of both precision and recall.

Another future research direction for this study might be the determination of the best location for inserting links. Because, since the link terms are important terms for the topic, they

usually occur in multiple locations of the text. Wikipedia guidelines suggest linking as early as possible, but local (with scope of paragraphs or subheadings) relevancy should also be considered for the decision. Therefore, the system should select the most leading location which is locally relevant enough. The investigation of this location might require an analyze of the article text and this need might generally change the direction of the study from information retrieval to natural language processing.

As a last future extension, we think that the category suggestion for Wikipedia articles might have common issues with the missing link discovery approach. It might be developed by using similar methods to select from different candidate categories according to their relationships with other articles and categories.

REFERENCES

[1] "Wikipedia." http://www.wikipedia.org/, 15 November 2009.

[2] C. Fellbaum, ed., *WordNet: an electronic lexical database*. MIT Press, 1998.

[3] C. Matuszek, J. Cabral, M. Witbrock, and J. DeOliveira, "An introduction to the syntax and content of cyc," in *Proceedings of the 2006 AAAI Spring Symposium on Formalizing and Compiling Background Knowledge and Its Applications to Knowledge Representation and Question Answering*, 2006.

[4] I. Niles and A. Pease, "Towards a standard upper ontology," in *FOIS '01: Proceedings of the international conference on Formal Ontology in Information Systems*, (New York, NY, USA), pp. 2–9, ACM, 2001.

[5] Wikipedia, "Wikipedia linking guideline." http://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style_(links), 15 November 2009.

[6] R. Mihalcea and A. Csomai, "Wikify!: linking documents to encyclopedic knowledge," in *CIKM '07*, (New York, NY, USA), ACM, 2007.

[7] S. F. Adafre and M. de Rijke, "Discovering missing links in wikipedia," in *LinkKDD '05: Proceedings of the 3rd international workshop on Link discovery*, (New York, NY, USA), pp. 90–97, ACM, 2005.

[8] Wikipedia, "Encyclopedia." http://en.wikipedia.org/wiki/Encyclopedia, 15 November 2009.

[9] Wikipedia, "Interpedia." http://www.wikipedia.org/Interpedia, 15 November 2009.

[10] "Online reference service including the content of columbia encyclopedia." http://www.encyclopedia.com/, 15 November 2009.

[11] Wikipedia, "List of online encyclopedias." http://en.wikipedia.org/wiki/List_of_online_encyclopedias, 15 November 2009.

[12] Wikipedia, "Microsoft encarta." http://en.wikipedia.org/wiki/Encarta/, 15 November 2009.

[13] Wikipedia, "Nupedia." http://en.wikipedia.org/wiki/Nupedia, 15 November 2009.

[14] "Wikimedia foundation." http://www.wikimediafoundation.org/, 15 November 2009.

[15] R. Mihalcea, "Using wikipedia for automatic word sense disambiguation," in *North American Chapter of the Association for Computational Linguistics (NAACL 2007)*, 2007.

[16] J. Voß, "Measuring wikipedia," no. PREPRINT 2005-04-12, 2005.

[17] D. Milne and I. H. Witten, "Learning to link with wikipedia," in *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management*, (New York, NY, USA), pp. 509–518, ACM, 2008.

[18] S. Fissaha Adafre, V. Jijkoun, and M. de Rijke, "Fact discovery in wikipedia," in *2007 IEEE/WIC/ACM International Conference on Web Intelligence*, November 2007.

[19] Wikipedia, "Wikipedia redirection guideline." `http://en.wikipedia.org/wiki/Wikipedia:Redirect`, 21 November 2009.

[20] J. Voss, "Collaborative thesaurus tagging the wikipedia way," *CoRR*, vol. abs/cs/0604036, 2006.

[21] Wikipedia, "Wikipedia categorization guideline." `http://en.wikipedia.org/wiki/Wikipedia:Categorization`, 21 November 2009.

[22] Wikipedia, "Wikipedia namespaces." `http://en.wikipedia.org/wiki/Wikipedia:Namespace`, 21 November 2009.

[23] T. Zesch and I. Gurevych, "Analysis of the Wikipedia category graph for NLP applications," in *Proceedings of the Second Workshop on TextGraphs: Graph-Based Algorithms for Natural Language Processing*, (Rochester, NY, USA), pp. 1–8, Association for Computational Linguistics, 2007.

[24] "Roget's thesaurus." `http://thesaurus.reference.com/`, 23 November 2009.

[25] "del.icio.us." `http://delicious.com/`, 23 November 2009.

[26] D. Vizine-Goetz, "Classification schemes for internet resources revisited," *Journal of Internet Cataloging*, vol. 5, 2002.

[27] T. Holloway, M. Bozicevic, and K. Börner, "Analyzing and visualizing the semantic coverage of wikipedia and its authors: Research articles," *Complex.*, vol. 12, no. 3, pp. 30–40, 2007.

[28] A. Capocci, V. D. P. Servedio, F. Colaiori, L. S. Buriol, D. Donato, S. Leonardi, and G. Caldarelli, "Preferential attachment in the growth of social networks: The internet encyclopedia wikipedia," *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)*, vol. 74, no. 3, 2006.

[29] V. Zlatic, M. Bozicevic, H. Stefancic, and M. Domazet, "Wikipedias: Collaborative web-based encyclopedias as complex networks," *Physical Review E*, vol. 74, p. 016115, 2006.

[30] A. Mehler, "Text linkage in the wiki medium-a comparative study," 2006.

[31] F. Bellomi and R. Bonato, "Network analysis for wikipedia," in *The First International Wikimedia Conference*, 2005.

[32] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *J. ACM*, vol. 46, no. 5, pp. 604–632, 1999.

[33] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," in *Computer Networks and ISDN Systems*, pp. 107–117, 1998.

[34] L. S. Buriol, C. Castillo, D. Donato, S. Leonardi, and S. Millozzi, "Temporal analysis of the wikigraph," in *WI '06: Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, (Washington, DC, USA), pp. 45–51, IEEE Computer Society, 2006.

[35] M. Strube and S. P. Ponzetto, "Wikirelate! computing semantic relatedness using wikipedia," in *AAAI'06: proceedings of the 21st national conference on Artificial intelligence*, pp. 1419–1424, AAAI Press, 2006.

[36] E. Gabrilovich and S. Markovitch, "Computing semantic relatedness using wikipedia-based explicit semantic analysis," in *IJCAI'07: Proceedings of the 20th international joint conference on Artifical intelligence*, (San Francisco, CA, USA), pp. 1606–1611, Morgan Kaufmann Publishers Inc., 2007.

[37] D. Milne and I. H. Witten, "An effective, low-cost measure of semantic relatedness obtained from wikipedia links," in *In Proceedings of the AAAI 2008 Workshop on Wikipedia and Artificial Intelligence (WIKIAI 2008)*, 2008.

[38] E. Yeh, D. Ramage, C. D. Manning, E. Agirre, and A. Soroa, "Wikiwalk: Random walks on wikipedia for semantic relatedness," in *Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing (TextGraphs-4)*, (Suntec, Singapore), pp. 41–49, Association for Computational Linguistics, August 2009.

[39] D. Turdakov and P. Velikhov, "Semantic relatedness metric for wikipedia concepts based on link analysis and its application to word sense disambiguation," in *SYRCoDIS* (S. D. Kuznetsov, P. Pleshachkov, B. Novikov, and D. Shaporenkov, eds.), vol. 355 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2008.

[40] N. Jakob, M.-C. Müller, and I. Gurevych, "Lrtwiki: Enriching the likelihood ratio test with encyclopedic information for the extraction of relevant terms," in *Proceedings of the WikiAI 09 - IJCAI Workshop: User Contributed Knowledge and Artificial Intelligence: An Evolving Synergy*, (Pasadena, CA, USA), Jul 2009.

[41] T. Dunning, "Accurate methods for the statistics of surprise and coincidence," *Comput. Linguist.*, vol. 19, no. 1, pp. 61–74, 1993.

[42] M. Grineva, M. Grinev, and D. Lizorkin, "Extracting key terms from noisy and multi-theme documents," in *18th International World Wide Web Conference (WWW2009)*, April 2009.

[43] R. C. Bunescu and M. Paşca, "Using encyclopedic knowledge for named entity disambiguation," in *EACL*, The Association for Computer Linguistics, 2006.

[44] A. Toral and R. Munoz, "A proposal to automatically build and maintain gazetteers for Named Entity Recognition by using Wikipedia," *EACL 2006*, 2006.

[45] E. Gabrilovich and S. Markovitch, "Overcoming the brittleness bottleneck using wikipedia: enhancing text categorization with encyclopedic knowledge," in *AAAI'06: proceedings of the 21st national conference on Artificial intelligence*, pp. 1301–1306, AAAI Press, 2006.

[46] T. Weale, "Utilizing wikipedia categories for document classification.".

[47] P. Schonhofen, "Identifying document topics using the wikipedia category network," in *WI '06: Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, (Washington, DC, USA), pp. 456–462, IEEE Computer Society, 2006.

[48] P. Sorg and P. Cimiano, "Enriching the crosslingual link structure of wikipedia - a classification-based approach -," in *Proceedings of the AAAI 2008 Workshop on Wikipedia and Artificial Intelligence (WikiAI'08)*, 2008.

[49] S. F. Adafre and M. de Rijke, "Finding Similar Sentences across Multiple Languages in Wikipedia," *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 62–69, 2006.

[50] I. B. Péter Schönhofen, András Benczúr and K. Csalogány, "Performing cross-language retrieval with wikipedia," 2008.

[51] D. Nguyen, A. Overwijk, C. Hauff, R. B. Trieschnigg, D. Hiemstra, and F. M. G. de Jong, "Wikitranslate: Query translation for cross-lingual information retrieval using only wikipedia," in *Evaluating Systems for Multilingual and Multimodal Information Access*, vol. 5706 of *Lecture Notes in Computer Science*, (Berlin), pp. 58–65, Springer Verlag, 2009.

[52] S. Ferrández, A. Toral, O. Ferrández, A. Ferrández, and R. Munoz, "Applying wikipedia's multilingual knowledge to cross-lingual question answering," pp. 352–363, 2007.

[53] M. Agosti, F. Crestani, and M. Melucci, "On the use of information retrieval techniques for the automatic construction of hypertext," *Inf. Process. Manage.*, vol. 33, no. 2, pp. 133–144, 1997.

[54] M. Henzinger, "Hyperlink analysis on the world wide web," in *HYPERTEXT '05: Proceedings of the sixteenth ACM conference on Hypertext and hypermedia*, (New York, NY, USA), pp. 1–3, ACM, 2005.

[55] A. Faaborg and H. Lieberman, "A goal-oriented web browser," in *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, (New York, NY, USA), pp. 751–760, ACM, 2006.

[56] S. Drenner, M. Harper, D. Frankowski, J. Riedl, and L. Terveen, "Insert movie reference here: a system to bridge conversation and item-oriented web sites," in *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, (New York, NY, USA), pp. 951–954, ACM, 2006.

[57] H. Lieberman and H. Liu, "Adaptive linking between text and photos using common sense reasoning," in *AH '02: Proceedings of the Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, (London, UK), pp. 2–11, Springer-Verlag, 2002.

[58] R. West, D. Precup, and J. Pineau, "Completing wikipedia's hyperlink structure through dimensionality reduction," in *CIKM '09: Proceeding of the 18th ACM conference on Information and knowledge management*, (New York, NY, USA), pp. 1097–1106, ACM, 2009.

[59] G. Salton and C. Buckley, "Term weighting approaches in automatic text retrieval," tech. rep., Ithaca, NY, USA, 1987.

[60] C. D. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*. Cambridge, MA: MIT Press, 1999.

[61] A. Csomai and R. Mihalcea, "Linking educational materials to encyclopedic knowledge," in *Proceeding of the 2007 conference on Artificial Intelligence in Education*, (Amsterdam, The Netherlands, The Netherlands), pp. 557–559, IOS Press, 2007.

[62] G. Jeh and J. Widom, "Simrank: a measure of structural-context similarity," in *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, (New York, NY, USA), pp. 538–543, ACM, 2002.

[63] C. David, L. Giroux, S. Bertrand-Gastaldy, and D. Lanteigne, "Indexing as problem solving: A cognitive approach to consistency," *Proceedings of the ASIS Annual Meeting*, vol. 32, 1995.

[64] "Wikimedia downloads." `http://download.wikimedia.org/`, 5 December 2009.

[65] T. Zesch, I. Gurevych, and M. Mühlhäuser, "Analyzing and accessing wikipedia as a lexical semantic resource," in *Biannual Conference of the Society for Computational Linguistics and Language Technology*, 2007.

[66] T. Zesch, C. Müller, and I. Gurevych, "Extracting lexical semantic knowledge from wikipedia and wiktionary," in *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)* (European, ed.), (Marrakech, Morocco), may 2008.

[67] "Apache lucene." `http://lucene.apache.org/`, 5 December 2009.

[68] E. Hatcher, O. Gospodnetic, and M. McCandless, *Lucene in Action, Second Edition*. Manning, 2009.

[69] "Luke." `http://www.getopt.org/luke/`, 5 December 2009.

[70] D. Harman, E. A. Fox, R. A. Baeza-Yates, and W. C. Lee, "Inverted files," in *Information Retrieval: Data Structures & Algorithms*, pp. 28–43, 1992.

[71] J. Zobel, A. Moffat, and K. Ramamohanarao, "Inverted files versus signature files for text indexing," *ACM Trans. Database Syst.*, pp. 453–490, 1998.

[72] "Jflex scanner generator tool." `http://jflex.de/`, 5 December 2009.

[73] P. Jaccard, "Étude comparative de la distribution florale dans une portion des alpes et des jura," *Bulletin del la Société Vaudoise des Sciences Naturelles*, vol. 37, pp. 547–579, 1901.

[74] Wikipedia, "Jaccard coefficient article on wikipedia." `http://en.wikipedia.org/wiki/Jaccard_index`, 22 December 2009.

[75] A. A. Akin and M. D. A. , "Zemberek, an open source nlp framework for turkic languages." 2007.

[76] Wikipedia, "Wikipedia guideline for lead section." `http://en.wikipedia.org/wiki/Wikipedia:Lead_section`, 25 December 2009.

[77] W. Dakka and S. Cucerzan, "Augmenting wikipedia with named entity tags," 2008.

[78] J. Kazama and K. Torisawa, "Exploiting wikipedia as external knowledge for named entity recognition," in *Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 698–707, 2007.

[79] L. A. Pizzato and R. Schwitter, eds., *Proceedings of the Australasian Language Technology Association Workshop 2009*, vol. 7, (Sydney, Australia), Australasian Language Technology Association, December 2009.

[80] K. Nakayama, "Wikipedia mining for triple extraction enhanced by co-reference resolution," 2008.

[81] T. Kliegr, K. Chandramouli, J. Nemrava, V. Svatek, and E. Izquierdo, "Wikipedia as the premiere source for targeted hypernym discovery," *WBBT/ECML'08*, 2008.

[82] C. R. Project., M. Keen, J. Mills, and C. W. Cleverdon, *Factors determining the performance of indexing systems*. Cranfield,, 1966.