CROSS-LINGUAL INFORMATION RETRIEVAL ON TURKISH AND ENGLISH TEXTS

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

BY

AKİF BOYNUEĞRİ

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

APRIL 2010

Approval of the thesis:

CROSS-LINGUAL INFORMATION RETRIEVAL
ON TURKISH AND ENGLISH TEXTS

submitted by **AKİF BOYNUEĞRİ** 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. Adnan Yazıcı _____
Head of Department, **Computer Engineering**

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

Examining Committee Members:

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

Dr. Ayşenur Birtürk
Computer Engineering Department, METU _____

Assist. Prof. Dr. Tolga Can
Computer Engineering Department, METU _____

Assist. Prof. Dr. Pınar Şenkul
Computer Engineering Department, METU _____

Dr. Markus Schaal
Computer Engineering Department, Bilkent University _____

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: AKİF BOYNUEĞRİ

Signature :

ABSTRACT

CROSS-LINGUAL INFORMATION RETRIEVAL
ON TURKISH AND ENGLISH TEXTS

Boynueğri, Akif
M.S., Department of Computer Engineering
Supervisor : Dr. Ayşenur Birtürk

April 2010, 66 pages

In this thesis, cross-lingual information retrieval (CLIR) approaches are comparatively evaluated for Turkish and English texts. As a complementary study, knowledge-based methods for word sense disambiguation (WSD), which is one of the most important parts of the CLIR studies, are compared for Turkish words.

Query translation and sense indexing based CLIR approaches are used in this study. In query translation approach, we use automatic and manual word sense disambiguation methods and Google translation service during translation of queries. In sense indexing based approach, documents are indexed according to meanings of words instead of words themselves. Retrieval of documents is performed according to meanings of the query words as well. During the identification of intended meaning of query terms, manual and automatic word sense disambiguation methods are used and compared to each other.

Knowledge based WSD methods that use different gloss enrichment techniques are compared for Turkish words. Turkish WordNet is used as a primary knowledge base and English WordNet and Turkish Wikipedia are employed as enrichment resources. Meanings of

words are more clearly identified by using semantic relations defined in WordNets and Turkish Wikipedia. Also, during calculation of semantic relatedness of senses, cosine similarity metric is used as an alternative metric to word overlap count. Effects of using cosine similarity metric are observed for each WSD methods that use different knowledge bases.

Keywords: Cross-lingual Information Retrieval, Word Sense Disambiguation, WordNet, Wikipedia

ÖZ

TÜRKÇE VE İNGİLİZCE METİNLERDE ÇOK DİLLİ VERİ ERİŞİMİ

Boynueğri, Akif
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi : Dr. Ayşenur Birtürk

Nisan 2010, 66 sayfa

Bu tezde Türkçe ve İngilizce metinlerde çok dilli veri erişim yaklaşımları karşılaştırılmıştır. Bunun yanında çok dilli veri erişim çalışmalarının çok önemli bir kısmını oluşturan anlam belirsizliklerinin giderilmesi metodları Türkçe kelimeler için karşılaştırılmıştır.

Çok dilli veri erişimi gerçekleştirilirken, sorgu metninin hedef dile çevrilmesi ve anlamlara göre indeksleme yaklaşımları kullanılmıştır. Sorgu metninin hedef dile çevrilmesi sırasında, otomatik ve manual olarak kelime anlamlarının tespit edilmesinin yanı sıra *Google translation* (çevrim) servisi kullanılmıştır. Anlam indeksleme yaklaşımında ise, dokümanlar içerdikleri kelimeler yerine kelimelerin sahip oldukları anlamlara göre indekslenmiştir. Veri erişimi ise yine kelimelere göre değil anlamlara göre yapılmaktadır. Anlam seviyesinde sorgu oluşturulurken manual ve otomatik anlam belirleme metodları kullanılmış ve karşılaştırılmıştır.

Çok dilli veri erişim yaklaşımlarının karşılaştırılmasının yanı sıra, Türkçe kelimelerde bilgi tabanı dayalı anlam belirsizliklerinin giderilmesi metotları karşılaştırılmıştır. Bu çalışmada bilgi tabanı olarak, Türkçe WordNet'e ilaveten anlam zenginleştirme çalışmaları için İngilizce WordNet ve Türkçe Wikipedia kullanılmıştır. Kelimelerin WordNet'te bulunan anlamsal

ilişkileri ve Türkçe Wikipedia kullanılarak, anlamlar arasındaki ilişkilerin daha açık bir şekilde ortaya çıkması sağlanmıştır. Bunun yanında, anlamlar arasındaki ilişkilerin belirlenmesi sırasında ölçüt olarak anlam tanımlarında bulunan ortak kelime sayısının yanı sıra *cosine similarity* ölçütü kullanılmıştır ve her bir zengileştirme metodu için karşılaştırılmıştır.

Anahtar Kelimeler: Çok Dilli Veri Erişimi, Anlam Belirsizliklerinin Giderilmesi, WordNet, Wikipedia

To My Family,

ACKNOWLEDGMENTS

I would like to express my deepest gratitude and profound respect to my supervisor Dr. Ayşenur Birtürk for her expert guidance and suggestions, positive approach throughout my master study and her efforts and patience during supervision of the thesis.

I would like to express my thanks to the jury members, Prof. Dr. İsmail Hakkı Toroslu, Assist. Prof. Dr. Tolga Can, Assist. Prof. Dr. Pınar Şenkul and Dr. Markus Schaal for reviewing and evaluating my thesis.

I would like to express my sincere appreciation to Özgür Bağlıoğlu, Alptuğ Dilek, Çağla Okutan, Ömer Sunercan, Can Çermikli, Özay Duman, Fatih Kaya, Ahmet Yortanlı and all of my colleagues for their encouragement and support during my thesis work.

I would like to thank to TÜBİTAK UEKAE / G222 for supporting my academic studies.

Finally special thanks to my family for bringing me up and making me who I am with their love, trust, understanding and every kind of support throughout my life.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

Information retrieval (IR) is the task of extracting documents that are relevant to user defined query. With the tremendous growth of WEB, IR has become on indispensable tool for the Internet users. Unstructured and unorganized information spread around the Internet makes information retrieval for more crucial today. Although most IR techniques are developed for English, the Internet is no longer just based on English content; non-English content is also growing rapidly. However, monolingual users can only reach documents covering just a small portion of the Internet using their native languages. Also for users who can understand but cannot express themselves in a different language, querying documents in different language is a challenging task. Thereby, due to rapid growth of multi-lingual content on the Internet, cross-lingual information retrieval (CLIR) has become more and more important.

CLIR deals with retrieval of documents based on a query that is formulated by a user using his natural language regardless of the query and document language [4]. Main focus of CLIR is to select and rank information expressed in a language different than the query language. There are several approaches for CLIR mentioned in Oard [5] and Pirkola [6]. These approaches are generally based on translation of query or document collection into a common language. However, the main challenge of CLIR is the ambiguity of languages during translation. Identification of indented meanings of the words, called word sense disambiguation (WSD), is considered as one of the most important issues for CLIR studies. There are many words that have more than meanings. For example Turkish word *oyun* has 10 senses, such as *"a contest with rules to determine winner"* and *"a theatrical performance of a drama"* [7]. WSD refers to resolution of lexical semantic ambiguity.

Identification of meanings of the words and using these meaning in CLIR has some challeng-

ing tasks. At first, global sense knowledge bases for both language is required. Also cross-lingual transitions must be defined between senses in these pools. Fortunately, these sense definition pools namely Turkish and English WordNet are constructed just as it is needed. WordNet is an ontological dictionary that includes semantic relations of terms and meanings of terms like synonymy, antonymy, hypernymy etc. These semantic relations are used in many research areas like WSD, IR, text classification. Turkish WordNet is built by The Human Language and Speech Technologies Laboratory at Sabancı University [8] in the scope of BalkaNet Project. English WordNet is constructed by Princeton University [7].

However, incompleteness of knowledge bases which leads to incomplete or wrong disambiguation of texts is still another challenge for CLIR. Insufficient context is also another problematic issue for CLIR, especially during translation of short queries. For example for a query that consist of a single word it is not possible identify intended meaning of the word unless it has one single meaning.

1.1 Motivation

The main goal of thesis is to implement and evaluate CLIR on Turkish and English texts. Main motivation of the thesis is to fulfill the gap in CLIR studies conducted for Turkish, so that monolingual Turkish users can benefit from the content written in English. Also by implementation different CLIR methods, we want to initiate CLIR studies for Turkish. In the thesis, we compare six different CLIR methods based on query translation and sense indexing approaches.

In addition, as a sub goal we evaluate different enrichment processes for knowledge based WSD methods for Turkish texts. Knowledge based methods can be considered as different variations and adaptations of the Lesk algorithm [9]. The main motivation for this extra study is to observe the effects of different enrichment methods on WSD separately for Turkish words. There are many studies conducted for English so that more precise interpretations can be made for CLIR methods that use the Lesk algorithm. Hence, we want to see accuracy of WSD method that is used for disambiguation of Turkish words.

1.2 Thesis Organization

Thesis is organized as follows: In Chapter 2, literature review of studies on CLIR is explained. Then studies on WSD are explained. In Chapter 3, WSD methods on Turkish texts will be explained. In Chapter 4, implemented CLIR methods are mentioned. Then dataset preparation systems are described in the Chapter 5. In Chapter 6, evaluation results of CLIR algorithms are analyzed. WSD methods results are explained in Chapter 7. In Chapter 8 evaluation result and findings will be discussed and promising future research directions based on the thesis work is mentioned.

CHAPTER 2

RELATED WORK

In this chapter, related work in cross-lingual information retrieval (CLIR) and word sense disambiguation (WSD) is discussed. There are two main approaches for CLIR in the literature: translation approach and sense indexing based approach. Details of these approaches are discussed in the CLIR related work section. In this thesis, we achieve CLIR by known adapting knowledge based WSD method. Therefore, WSD studies in the literature are presented in the WSD related work section. In the thesis, we compare WSD methods that use different knowledge bases. Hence, in he last section of this chapter English and Turkish WordNets and Wikipedia are described.

2.1 Cross-lingual Information Retrieval

CLIR is generally achieved by converting query language and document language into a common representation. This conversion is performed by translation of query sentence to the target document collection language or replacing the words in the query and document with their meanings. Therefore, CLIR studies in the literature can be categorized into two groups namely: translation approach and semantic indexing approach. In translation based approaches, there are two different alternatives according to translation method used. Translation can be performed by using dictionaries or bi-lingual parallel corpora. Dictionary based translation methods use machine readable dictionaries like WordNet and Wikipedia and benefit from grammar rules defined for the languages. However, statistical methods are adopted in methods that use bi-lingual parallel corpora. Second approach sense indexing based approach can be regarded as a sub part of translation approach, because WSD is a sub task of language translation. However, translation based approaches still employ information retrieval using

words. However, we prefer to discriminate these two approaches to emphasize on difference between keyword retrieval and sense retrieval.

2.1.1 Translation Based CLIR

CLIR can be provided by translation of the query sentence into the target language or translation of whole document collection into a query language. In document translation approach, whole document collection is translated into the target language then monolingual information retrieval performed on the translated document collection using user query in the target language [10]. However, translation of all document collection which is especially large and unstable can be costly. This performance bottleneck reduces the scalability of CLIR systems that translate the document collection. Translation of all document collection also demands more hardware resources comparing other alternatives. The other translation based CLIR method is query translation which is widely preferred. Query translation method is relatively efficient compared to document translation. However, one of most important drawback of query translation is to resolve ambiguity of short queries during translation since disambiguation of terms in a small context may not be possible [11]. For instance, translation of query with one word is not possible if word has more than one sense. So main issue of query translation methods is controlling query translation ambiguity.

There are several CLIR method evaluations conducted. In his study Oard [5] compares six different CLIR methods which are same language query, dictionary based query translation, machine translation based query translation, lexical conceptual structure based query translation, machine translation based document translation and foreign language. Same language query method is manual translation of the query which is used as a control group in the study. Manual translation of query used in order to determine upper bound for CLIR methods. Dictionary based CLIR is essentially based on replacing each query term with the corresponding words in the dictionary. While performing query translation, Oard does not apply disambiguation for the query terms. In a small context it does not always possible to translate the query so there six dictionary based translation method is used. These translation methods are based on selecting first matching of the word or stem of the word from the dictionary. In the machine translation method, Oard uses Logos machine translation system. In lexical conceptual structure query translation method, preparation of disambiguated query is done according

to event based entries. Firstly syntactic relations of query words are extracted using a text parser. According to these syntactic relations query translation is applied. According to his study, the best results are achieved by manual translation, then machine translation approach is second ranked, especially performance of machine translation approach for long queries is satisfactory.

In the study that is relied on query translation, called WikiTranslate Wikipedia is used as a knowledge base [12]. Wikipedia articles are regarded as senses and keywords that have links to these articles are considered as morphological variations of these articles. At first step the query is mapped to the Wikipedia concepts by retrieving most related articles in Wikipedia. The most related articles are selected by applying classic information retrieval methods. Then each term in the query is searched in the content of the articles or in the internal link structure of the articles. The Wikipedia concept that has the most number of links in the related articles is assigned to the word. For example for query, *new album of Jackson*, the term Jackson is mapped to the Jackson article that has most linked from initially retrieved related documents. Then using cross-lingual links of Wikipedia, query translation is performed. CLIR results of Wikitranslate system are reasonable and Wikipedia is regarded as an alternative to the current translation resources.

2.1.2 Sense Indexing Based CLIR

Sense indexing method is based on transforming documents and queries into intended meanings of the words. Generally in this approach, documents are indexed according to the meanings of the words. Therefore identification of meanings of the words requires word sense disambiguation. And retrieval of related documents from sense index of document collection is also performed according to the query that is transformed into senses. So all words in the query is disambiguated. Sense indexing approach is used for eliminating polysemy and synonymy problem in the information retrieval [13], [14]. Polysemy problem in the information retrieval is that a word can have more than one meaning and document retrieved may not be related to the intended meaning of query. Polysemy problem causes low precision in the retrieval. Synonymy problem is that a word can have more than one synonym and since query does not include all synonyms, related documents may not be retrieved. Synonymy problem decreases the recall performance of the information retrieval system. Thereby

indexing according to senses and performing retrieval according to senses aim to eliminate these problems. Using dictionaries which contain cross-lingual sense matching like WordNet, sense indexing based approach can be used in the CLIR studies because senses are language independent.

In order to eliminate polysemy and synonym problem, semantic indexing based information retrieval systems are developed [13], [14]. Shütze prefers to apply co-occurrence based WSD algorithm based on corpus not hand-built lexical source. In the study of Shütze, retrieval reaches partly success due to the fact of avoided assumption of WSD which are *"one sense for one occurrence"* and *"fixed number of senses per word"*. But in the study, it is assumes that multiple senses of a word can be simultaneously used in the method. Mihalcea uses WordNet base WSD algorithm, however only 55% of the words can be disambiguated. Due to lack of coverage, hybrid retrieval method is used which combines sense and word based indexing results.

2.2 Word Sense Disambiguation

WSD was firstly considered as a different computational linguistic problem in late 1940s [15]. Problem is defined by Weaver [16] as that without looking words in the neighborhood in the context it is impossible to determine intended meaning of the word. This problem is also regarded as a difficult problem, even Bar-Hillel [17] mentioned that it is not possible for computer program to determine meaning of word *"pen"* in the below sentence because disambiguation requires all world knowledge like relative size of objects.

Little John was looking for his toy box. Finally he found it. The box is in the pen. John was very happy.

In 1990s, by construction of WordNet, start of Senseval and application of machine learning algorithms to WSD problem become milestone for WSD studies [15]. Machine learning based WSD plays important role in the research area and these methods result with the highest accuracy in the Senseval competitions. Senseval is a community that establishes a framework for WSD evaluation. There are lots of WSD applications that reports performance results but it is not possible to objectively compare success of these methods. Therefore Senseval bring standardization for evaluation of WSD algorithms.

WSD approaches can be categorized into three groups: supervised, unsupervised and knowledge based approaches [15]. Supervised WSD approaches use machine learning algorithms. Unsupervised approach is based on clustering of words in the context. Knowledge based methods use ontological and lexical resources like WordNet to identify sense of the words.

2.2.1 Supervised Word Sense Disambiguation

Supervised word sense disambiguation are generally performed by applying machine learning algorithms. In this approach, WSD problem is considered as a classification problem and the goal of the classifier is to correctly identify sense of word in the context according formerly learned training data. Supervised WSD algorithms outperform unsupervised algorithm as shown by experimental studies conducted by SENSEVAL [15].

One of the most popular machine learning algorithm is Naive Bayes which is applied to the to the WSD problem [18]. Another classifier is introduced by Gale, Church and Yarowsky [19]. This classifier is adaptation of Bayes classifier and reaches 90% accuracy while disambiguating two senses of small set of words. Mooney [20] mentions that Naive Bayes and neural networks obtains 73% accuracy for the word *line* which has six senses. Also 3-nearest neighbor, perceptron, decision tree algorithms are mentioned in the Mooney's survey.

Main problem of supervised WSD algorithm is the preparation of sense tagged training corpus. There are two large sense tagged corpora available for English which are SemCor [21] and SENSEVAL [22] corpus. SemCor corpus is created by Princeton University which is sub set of English Brown Corpus. SemCor contains 700.000 part of speech tagged words. And there are about 200.000 words that are sense tagged according to WordNet 1.6 in SemCor. The corpus that is created by SENSEVAL is prepared by using small set of manually tagged corpus HECTOR. HECTOR corpus is created by Oxford University and contains 17M words.

In order to overcome preparation problem of large training corpora , there are two methods: *bootstrapping methods* and using collaboratively constructed data sources. *bootstrapping methods* are based on preparation of initial classifier then using this classifier, large training set is extracted [23], [24].

Using Wikipedia as a traning set for WSD explained in the study [25], [26]. One of these studies is Wikification which means the process of extraction keywords in the document and

linking keywords to the related Wikipedia article. In this study, articles in the Wikipedia are considered as a senses and keywords hyper-linked to these articles corresponds to the surface forms. Using nature of Wikipedia hyper-link structure, which surface word linked to which article can be extracted and by using global and context features of keywords, training vector can be prepared. Another study of Mihalcea is Word Sense Disambiguation Using Wikipedia, which proposes automatic creation of sense tagged corpora using Wikipedia. In this study firstly keywords and all co-occurrence of the keyword is extracted and these keywords are labeled in the context. Then these labels are manually linked to the corresponding WordNet senses. This process eases sense tagging process of corpora.

2.2.2 Unsupervised Sense Disambiguation

Unsupervised word sense disambiguation methods aim to overcome knowledge acquisition bottleneck [19]. Unsupervised methods use raw corpora which any sense tagging process is applied to. These approaches are based on the idea that same sense of word will have similar neighbor words. By clustering words and their contexts, sense of words can be induced. Pure unsupervised methods do not rely on any dictionaries nor ontologies. Main problem of this approach is lack of usage of shared sense inventory [27]. In other words, all senses are discriminated according to corpora and senses of a word can be change during the usage of different corpora, so it may not be possible to use and identify senses independently from corpora. While WSD disambiguates sense of terms, unsupervised methods just performs word sense discrimination. So defined senses are by unsupervised WSD depending on corpora and may not match sense in traditional dictionaries. Here we mention three main methods of unsupervised WSD that are context clustering, word clustering and co-occurrence graphs.

In *Context clustering*, occurrence of the word represented as *context vector*. Then applying clustering to these vectors, possible senses of the target word is identified. Shütze [28] proposes the algorithm called *context-group discrimination* that groups context of occurrence of ambiguous words.

Another method for unsupervised WSD is *word clustering* [29]. Lin introduces well-known approach that consists of identification of similar words to the target word. Similarity between words is calculated according to syntactic dependencies like subject-verb, verb-object etc. Feature vector is prepared by using these dependency relations and word clustering algorithm

is applied in order to discriminate sense of the target word.

2.2.3 Knowledge-Based Sense Disambiguation

There are generally two main types of knowledge based algorithms namely word overlap (the Lesk algorithm and variations) and calculation of semantic relatedness. These methods can be used with any knowledge bases that have senses and their relations. However, one of the most significant drawbacks of knowledge based WSD is coverage of dictionaries. Names, contemporary terms are generally not included in dictionaries. Thereby, this coverage problem leads searchers to the usage of collaboratively created dictionaries like Wikipedia [25]. Wikipedia contains more defined terms, defined information about terms and defined relations between terms by intra-link structure of document due community support of millions of people. Now, I will mention about two knowledge base sense disambiguation algorithms namely which are overlap of sense definitions and semantic relatedness.

2.2.3.1 Overlap of Sense Definitions

The Lesk Algorithm uses dictionary glosses to disambiguate word that has more than one meaning in the sentence. In the Lesk algorithm, the actual sense of a word is identified according to nearest words' glossaries. Sense with higher overlap with other glossaries is assigned to the word. Context of the word is determined by window around the word. A word is assigned to the sense whose gloss that shares most common terms of other words" glosses in the window. Visual representation of algorithm shown in the Figure 2.1. The Lesk algorithm using Oxford Advanced Learner's Dictionary has 50-70% precision [9].

However, there are some problems with the Lesk algorithm. One of these problems is that sentences that have lots terms with many senses can cause combinatorial inefficiency. For example, assume that for a sentence that has 10 words with average 10 sense, there are 10^{10} possible sense combinations available for this sentence. So combinatorial inefficiency reduces the scalability of systems that use the algorithm. Cowie proposes the simulated annealing method for the Lesk algorithm in order to eliminate efficiency problem [30]. In this variation of the Lesk algorithm, all definitions are collected from dictionary and terms that occur in these definitions get score equal to the their number of occurrences. Then all of these scores

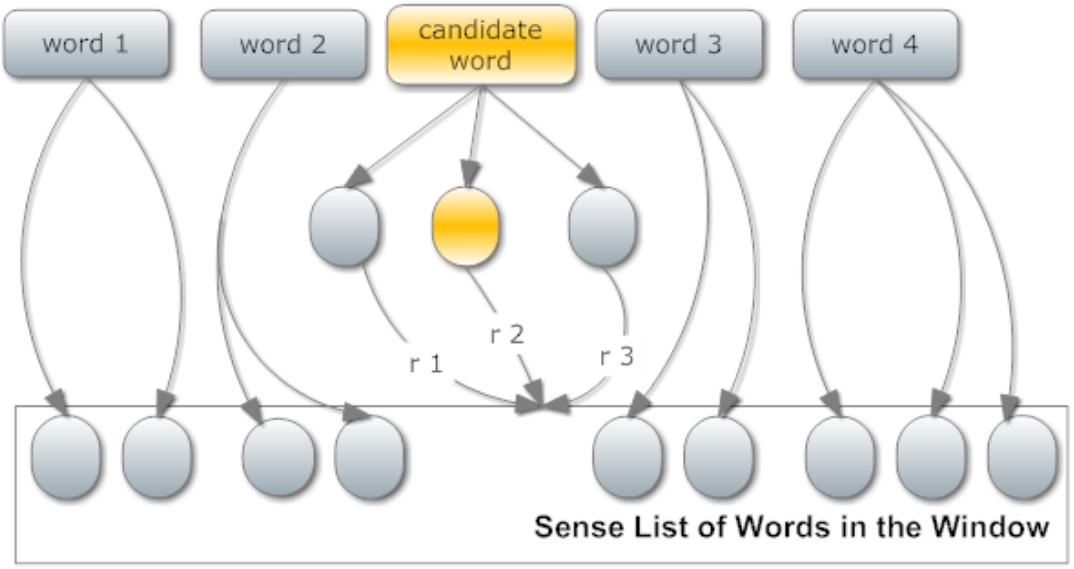

Figure 2.1: Graphic Representation of the Lesk Algorithm

are added and *redundancy* of text is obtained. The main aim of algorithm is to find combination of senses that minimizes the *redundancy* function. Then most frequent senses are selected as an initial combination. Sense of random word is replaced by another sense and *redundancy* is recalculated, if this replication reduces the *redundancy* function score, this replacement is regarded as a valid. And there are several iterations are performed until no change in the function achieved.

Another approach that attacks combinatorial explosion problem of the Lesk algorithm is *the simplified Lesk algorithm*. In the simplified Lesk algorithm, candidate senses are selected according to word overlap of gloss and context itself. In the original algorithm, word overlap calculation is made in terms of glosses of each word in the context. According to comparative study of Vasilescu [31] simplified Lesk algorithm outperforms the original Lesk algorithm in terms of precision and efficiency.

The third version of the Lesk algorithm is called *augmented spaces* method. This method is introduced by Banerjee and Pedersen and called the adapted Lesk algorithm [32]. The adapted Lesk algorithm enlarges the dictionary glosses while calculating semantic relatedness. In order to enrich gloss of senses, glosses of related senses are used. Related senses are selected based on WordNet relations such as hypernym, hyponym etc. In comparative evaluations the

adapted Lesk algorithm doubles the precision of the Lesk algorithm on English noun set of Senseval-2.

2.2.3.2 Semantic Relatedness

Other knowledge based sense disambiguation method is based on calculation of semantic relatedness of senses. Semantic relatedness is calculated according to local context of the given word. However, like the Lesk algorithm, algorithms based on semantic relatedness measures are computationally expensive. One of these similarity measures is introduced by *Leacock* [33]. Leacock similarity measure relies on shortest path between synsets. Then shortest path value is normalized by depth of taxonomy. In the Equation 2.1, $Path(C_1,C_2)$ corresponds length of path between two senses C_1 and C_2 and D corresponds length of taxonomy.

$$Similarity(C_1, C_2) = -log(\frac{Path(C_1, C_2)}{2D}) \tag{2.1}$$

Hirst and St-Onge [34] is another similarity measure which is based on direction of links in the path between two senses. In the Equation 2.2 $Path(C_1, C_2)$ corresponds length of path between senses and d represents the number of changes of direction.

$$Similarity(C_1, C_2) = C - Path(C_1, C_2) - kd \tag{2.2}$$

Other similarity measures are Resnik [35], Jiang and Conrath [36], Mihalcea and Moldovan [37] and Agire and Rigau [38].

2.3 WordNet

WordNet [7] is lexical database, which is one of the most important resources for computational linguistics, text analysis and related research areas. In the WordNet, nouns, verbs, adjectives and adverbs are grouped into cognitive synonyms that state distinct concepts. Also these synonym groups called synset are interlinked with each other according to semantic relations. These relations are shown in the Figures 2.2 and 2.3.

English WordNet has been developed by Princeton University [7]. English WordNet contains 155287 unique strings, 117659 synset and 206941 word-synset pairs. There are many related

Relation	Definition	Example
Hypernym	From concepts to subordinates	breakfast → meal
Hyponym	From concepts to subtypes	meal → lunch
Member Meronym	From groups to their members	faculty → professor
Has-Instance	From concepts to instances of the concepts	composer → Mozart
Instance	From instances to their concepts	Austen → author
Member Holonym	From members to their groups	copilot → crew
Part Meronym	From wholes to parts	table → leg
Part Holonym	From parts to wholes	course → meal
Antonym	Opposites	leader → follower

Figure 2.2: Noun relations in WordNet [1]

Relation	Definition	Example
Hypernym	From events to subordinate events	fly → travel
Troponym	From a verb to specific manner elaboration of that verb	walk → stroll
Entails	From verbs to the verbs they entail	snore → sleep
Antonym	Opposites	increase → decrease

Figure 2.3: Verb relations in WordNet [1]

application developed based English WordNet which are available on the web page of English WordNet [39].

Turkish WordNet is built by The Human Language and Speech Technologies Laboratory at Sabancı University [8] in the scope of BalkaNet Project. Turkish WordNet also linked to the English WordNet thereby all other wordnets that has linked to English WordNet. Current Turkish WordNet has cross-lingual references to the 2.0 version of English WordNet. Turkish WordNet is in the XML format shown in the Figure 2.5 and it can be used by permission of Sabancı University. There are currently 20420 words, 14795 synset and 6717 definitions in Turkish WordNet. Therefore there are some synsets without any definition. Structure of Turkish and English WordNet is shown in the Figure 2.4.

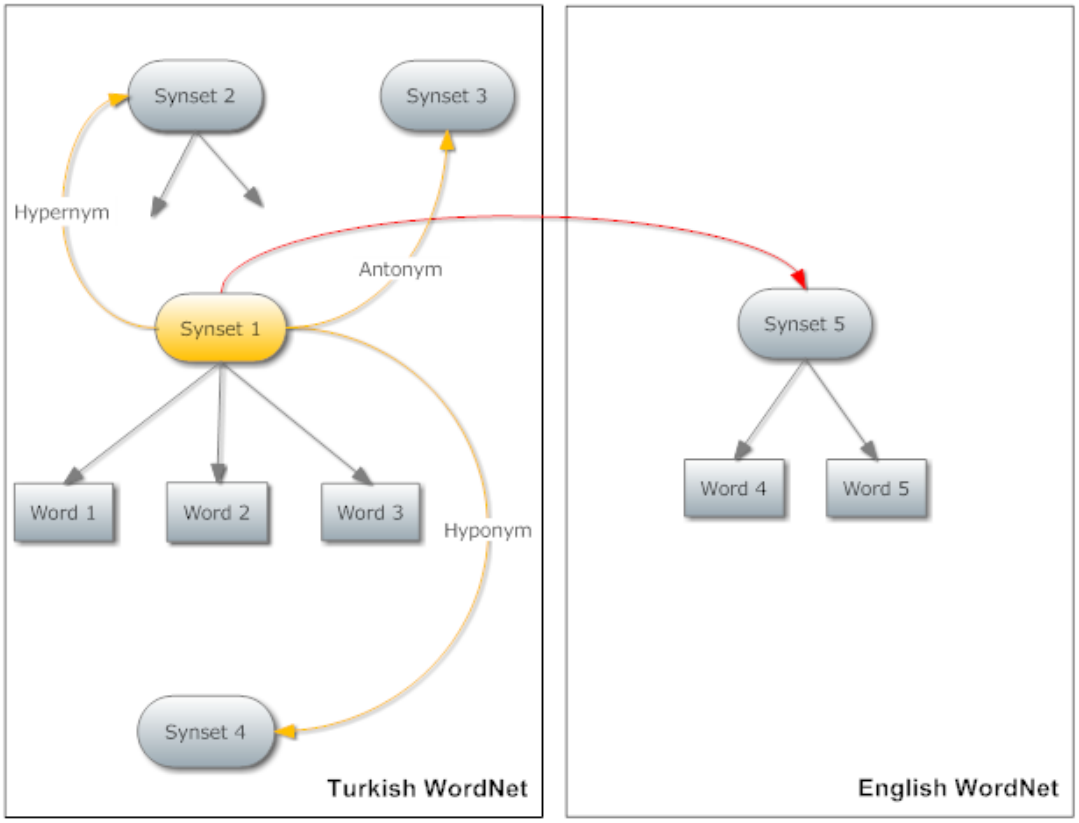

Figure 2.4: Structure of Turkish WordNet

2.4 Wikipedia

Wikipedia is multilingual, free content encyclopedia project. Wikipedia is collaboratively written by volunteers from all around the world. Wikipedia is one of largest reference Website on the Internet. Any person can contribute articles in Wikipedia, by just editing the article. Although convenience of changing content of articles makes Wikipedia grow rapidly, this leads to doubt about quality and correctness of articles. But auto control mechanism works well and Wikipedia became indispensable resource for researchers from diversity of areas. Since Wikipedia is the world's largest collaboratively edited source of encyclopedic knowledge and this manually edited content and structured knowledge is interpreted by computers, Wikipedia starts to be used as a knowledge base by the wide range of external applications.

Wikipedia is becoming one of the most important computational linguistic resources due to structured and machine readable information, completeness and up to dateness. WordNet is

```
<SYNSET>
<ID>ENG20-09901016-n</ID>
<POS>n</POS>
<SYNONYM>
<LITERAL>tezgahtar</LITERAL>
</SYNONYM><ILR>ENG20-09410258-n<TYPE>hypernym</TYPE></ILR>
<DEF>Kahve, gazino ve mağaza gibi yerlerde tezgahta duran, satış yapan kimse:</DEF>
<STAMP>ozlemc 2003/11/13</STAMP></SYNSET>
```

Figure 2.5: XML Format of Turkish WordNet

constructed by an academic group of people by manually or automatically and it does not have any community support during the creation phase. Wikipedia starts to become hot topic for the computational linguistic related areas. Wikipedia and its link structure are considered as a tagged corpora and machine learning algorithms can be applied using this corpora. Using Wikipedia for automatic sense disambiguation [25] and WikiTranslation can be shown as two examples of these studies. In our study Wikipedia is used as enrichment source due to its completeness and accessibility.

CHAPTER 3

WORD SENSE DISAMBIGUATION

In this chapter we elaborately explain word sense disambiguation (WSD) studies on Turkish text. WSD is one the most crucial parts of the CLIR studies, because accuracy of CLIR relies on correctness of WSD. Therefore, using better WSD algorithm will increase the quality of CLIR. In this thesis, we separately analyze and evaluate WSD algorithms for Turkish texts instead of observing effect of different WSD algorithms on CLIR methods.

In WSD studies, the Lesk algorithm and its variations are implemented and explained. The Lesk algorithm is one of the widely used knowledge based WSD algorithms as it is mentioned in the Chapter 2. We conduct comparative WSD study for Turkish words using different knowledge bases, different text enrichment methods and different sense relatedness metrics . Moreover, effects of different context window sizes for Turkish WSD are observed in the study.

Overview and flow of WSD process is shown in the Figure 3.1. The word and disambiguation context of the word is accepted as an input for WSD process. After stop-word elimination, dictionary based stemming is applied in order to find real morphological stem of the words. Then candidate senses for the word and senses of context words are fetched from Turkish WordNet which is used as a primary knowledge base in this study. All senses in the context window is put into the sense bag. Afterwards, gloss enrichment variations are applied using English WordNet and Turkish Wikipedia. Score of candidate senses for the word is calculated by total semantic relatedness score between all senses in the context sense bag. Then the candidate sense with highest score is assigned to the word.

In this chapter, we firstly describe common steps for WSD. And then details of alternative enrichment methods and different sense similarity metrics are explained. At last performance

consideration of the WSD process are discussed.

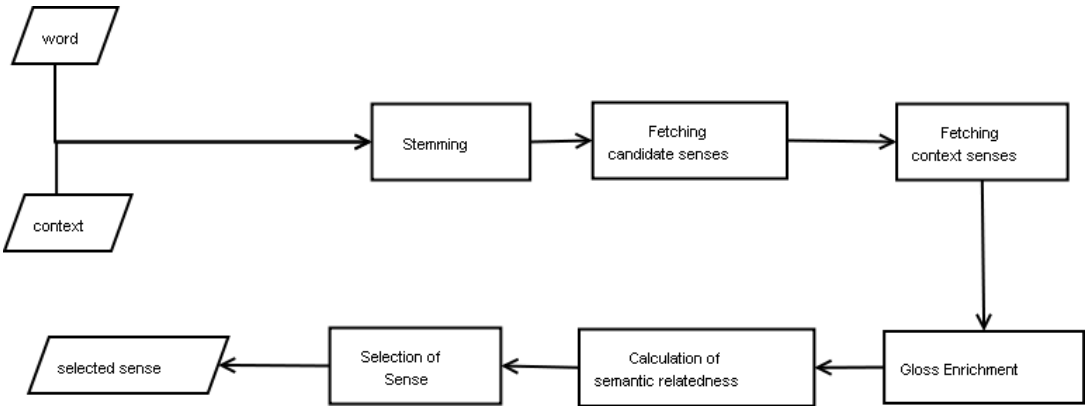

Figure 3.1: Overview of WSD Process

3.1 Stop Word Elimination

Stop-word elimination is applied as a first step for WSD. We use two stop-word list for both English and Turkish. Turkish stop-word list contains 147 words and the list is used in Information Retrieval on Turkish Texts [40]. English stop-word list contains 422 words and created by [41].

Stop-word elimination is commonly used task especially in information retrieval studies. Stop-words are repeated almost all documents and do not carry any document specific information. Stop-word list is generally extracted from large corpus by identifying words that are common for all documents. Also elimination of stop-words increase the computational performance of WSD because stop-words are contained by texts with high frequency so we do have to deal with these words fro WSD.

It is expected that removing stop-words improves calculation of semantic relatedness sense, because they are not specific for any gloss, almost all senses include these words. While computing relatedness of senses, not removing stop words will lead us to get irrelevant results and shadow the importance of other words in the glosses.

3.2 Dictionary Based Stemming

After elimination of stop-words, stemming is applied to the words in order obtain morphological stem of the words. Corresponding definition is fetched according to stem of the words. In this study we use two dictionary based stemmers for English and Turkish. As an English stemmer, stemmer implemented in JWI library [42] is used. This stemmer performs WordNet based stemming. For Turkish, Zemberek library [43] which is open-source linguistic analysis library and used by other open source applications like Open Office as Turkish spell checker. There can be many stems that are extracted as a result of linguistic analysis of used stemmers, but we use the first stem of given word in our study.

Stemming procedure is generally based on removal of suffixes of the words. Main goal of the stemming is reducing morphological variations of words and exposing relations of these variations. In this phase of WSD, stemmed word list is generated from context to be able to locate words in order to fetch the senses, disambiguation word is also stemmed. For example, the word *cars* cannot be found in the dictionary whereas; *car* is included by dictionary. Since the goal of this stemming process is to find the real morphological stem of the word, dictionary based stemmers are used although performance of dictionary based algorithm is worse than non dictionary based stemmers.

3.3 Sense Disambiguation Using Different Knowledge Bases

After preprocessing phase is performed, alternative WSD algorithms are applied to the word. During sense disambiguation of Turkish words, Turkish WordNet, English WordNet and Turkish Wikipedia are used as knowledge bases. There are nine alternative methods that are implemented using different knowledge bases. These methods are explained in the rest of this section.

3.3.1 Using Turkish WordNet Gloss

First method is the Lesk algorithm [9] using Turkish WordNet. This method uses Turkish gloss of senses in order to identify semantic relatedness between two senses. Candidate senses for disambiguation are fetched from Turkish WordNet. Sense bag for context words is also

extracted from Turkish WordNet. For each sense Turkish gloessed are assigned.

As an observation, in the Turkish WordNet there are some senses which do not have any gloss or have glosses which are written in English. This situation most probably reduces accuracy of the Lesk algorithm, since the Lesk algorithm is based on the count of common words of glosses of senses. So senses which do not have a gloss will not be chosen by the algorithm unless there is only a one candidate sense.

3.3.2 Using English WordNet Gloss

Second method is the Lesk algorithm using Turkish and English WordNet. In this method, senses for given Turkish words are selected from Turkish WordNet and glosses that are extracted from English WordNet is assigned to these senses. While fetching English gloss, cross-lingual references of senses in Turkish WordNet to English WordNet is used. Therefore, semantic relatedness of Turkish words is calculated using English glosses. So we carry sense disambiguation problem from Turkish WordNet to more settled English WordNet. Semantic relatedness of Turkish words is computed using English gloss words. Furthermore, although in this study we do not perform multi-lingual text disambiguation, using English gloss provides us to perform cross-lingual WSD for multi lingual texts.

3.3.3 Using English WordNet and Semantic Relations

This method for the WSD is the adapted Lesk algorithm which is based on enrichment of gloss content of senses with WordNet relations [32]. Candidate sense are selected using Turkish WordNet and gloss of senses and semantic relations of senses are employed from English WordNet. Since semantic relations in the WordNet are based on senses, we can use these relations for Turkish words on English senses. For example, one sense of *bicycle* which has *a wheeled vehicle that has two wheels and is moved by foot pedals* definition in the English WordNet 2.0, has hypernym relation with *wheeled vehicle* sense, and in every language bicycle with given sense is wheeled vehicle. Overview of WSD procedure using English WordNet relation is shown in the Figure 3.2.

In the study proposed by Banerjee [32], enrichment is performed by gloss of related words in order to clearly identify semantic relation between two senses. In this study, in addition

to gloss enrichment, we perform enrichment using synonymous words of the related senses. The idea behind is that most related words for the given sense are synonymous words of this sense. Therefore, we implement six different enrichment methods using English WordNet.

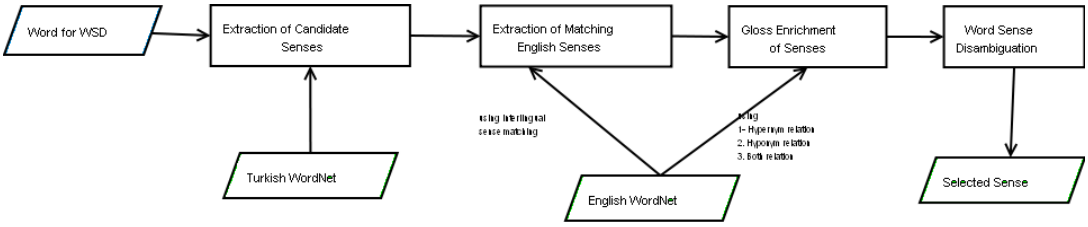

Figure 3.2: Overview of WSD Using English WordNet Semantic Relations

We use six variations of gloss enrichment using English WordNet which is provided by:

Hypernym Gloss Enrichment is performed by using gloss of hypernym senses.

Hyponym Gloss Enrichment is performed by using gloss of hyponym senses.

Hypernym and Hyponym Gloss Enrichment is performed by using gloss of hyponym and hypernym senses.

Hypernym Word Enrichment is performed by using words linked to hypernym senses with synonymy relation.

Hyponym Word Enrichment is performed by using words linked to hyponym senses with synonymy relation.

Hypernym and Hyponym Word Enrichment is performed by using words linked to hypernym and hyponym senses with synonymy relation.

3.3.4 Using Turkish Wikipedia

This method is based on enrichment of gloss of senses using Turkish Wikipedia. Wikipedia is used as a supplementary resource in order to explicitly identify related words during the clustering of short texts [44]. In this study it is mentioned that clustering of text is performed according to common terms in the documents. However, for the short text it is not always

possible obtain relatedness of the text due to lack of terms. So Banerjee enriches short text using Wikipedia and as a result of the study clustering accuracy of short text has improved 8%-20%.

Same notion is adopted in our study. Glosses of senses are considered as a short text and Turkish Wikipedia is used in order to clearly identify semantic relatedness of these glosses. Enrichment process is regarded as an information retrieval task. Firstly Turkish Wikipedia is indexed using the Lucene indexer [45] which is adapted for Turkish. Details of this modification is mentioned in the Chapter 4. Then after extracting candidate senses from Turkish Wikipedia, queries are prepared using Turkish glosses of the senses. Then ten most related documents are retrieved from Turkish Wikipedia and titles of these related articles are appended to the gloss of sense. After enrichment process WSD algorithm is applied to these senses. Overview of the method is shown in the Figure 3.3.

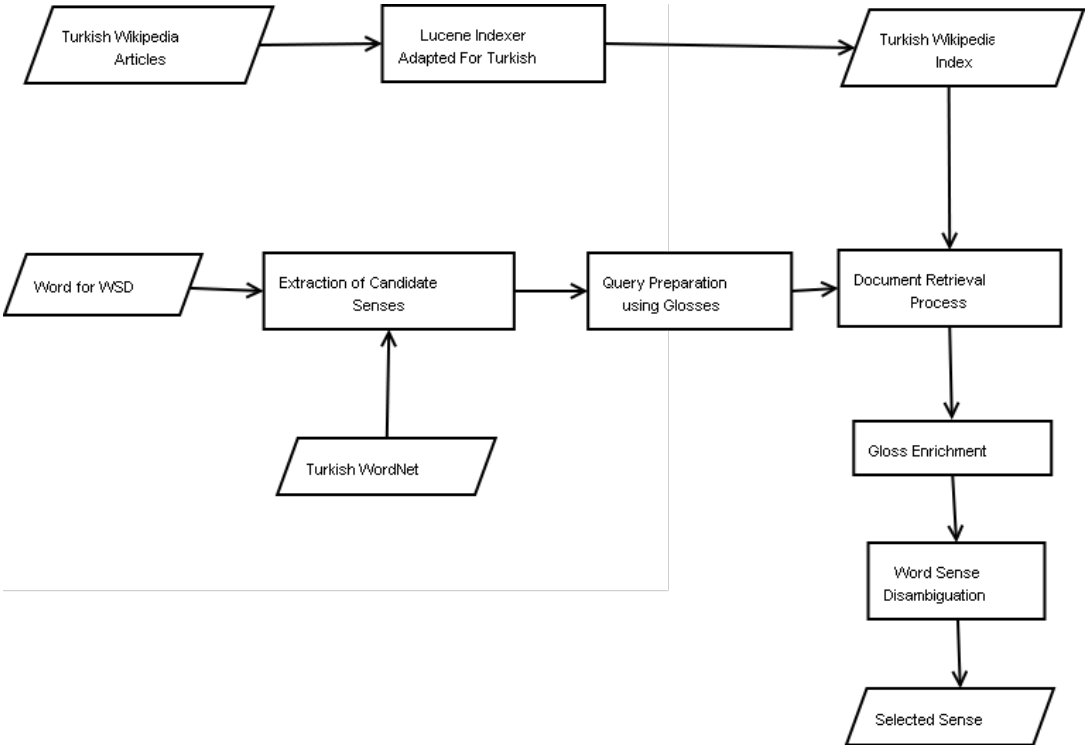

Figure 3.3: Overview of WSD Using Turkish Wikipedia

3.4 Scoring Candidate Senses

Score of candidate senses for the word is calculated in terms of textual relatedness of gloss of candidate sense and context word glosses. Before computing relatedness score of each sense, words in the glosses are stemmed. However, in this step, non dictionary based stemmer is used due to performance considerations unlike first step of WSD process. Then by using different similarity metrics, score of each senses are calculated. Relatedness score of candidate sense is calculated by total similarity score of candidate sense with all context words' senses.

3.4.1 Non Dictionary Based Stemming

Calculation of relatedness between two glosses is performed on stem of the words. Stemming is the process of reducing words to their stem or base form [46]. By stemming, all morphological variation of term is represented by one term so logical relations of these term variations are exposed. For example without stemming *car* and *cars* terms are considered as different terms however they refer to the same term and relatedness between two senses should be incremented due to this term.

Before computing relatedness of senses, words in the gloss of each senses are stemmed and stemmed word list is obtained. For stemming process non-dictionary based stemmers are used, these stemmers generally remove suffixes of words using defined rules. Stem produced by non-dictionary based stemming algorithms does not have to be identical to morphological stem, generally suffixes are removed and terms are reduced to same stem. In other words stem represents related words in same surface representation, but this representation does not have to be meaningful word. Despite the fact that they are faster than dictionary based algorithms, since all stemming process is performed on the memory. Computing semantic relatedness taking into account for performance of system is one the main considerations and there is not any dictionary dependent operation, non dictionary based stemming algorithms are used for Turkish and English for WSD.

In this study, for English words Porter stemming algorithm [47] and for Turkish words algorithm proposed by Eryigit [48] are used and during the stemming process we use Snowball Java API [49] that includes stemming algorithms for almost all European languages including English and Turkish. Snowball is a rule based String manipulation language. Codes written

in Snowball can be converted into Java or C++ programming languages.

3.4.2 Sense Disambiguation Using Different Similarity Metrics

In the traditional Lesk algorithm semantic relatedness of two senses are calculated according to word overlap count of glosses. However, there are problems with word overlap count because WSD algorithms based on this metric tend to select senses which have more words in their glosses. In this study, cosine similarity metric is also used in order to identify relatedness of two senses. Each gloss is converted to the vector with n dimensions in vector space model. Then cosine similarity of two vectors is used for scoring [50]. Representation of glossaries as vectors is shown in the Figure 3.4.

Terms / Docs	t1	t2	t3	.....	tn
d1	1	0	1	.....	2
d2	3	1	1	.....	0
d3	1	2	0	.....	1
...	...	...	...	.....	...
dn	2	0	1	.....	0

Figure 3.4: Representation of Text With Vector Space Model

In the Equation 3.1, A and B represent term frequency vectors. After dot product of these vectors, result is divided by length of each vectors.

$$Similarity(A, B) = \frac{A.B}{||A||.||B||} \tag{3.1}$$

Using cosine similarity while determining relatedness aims to eliminate tendency of the Lesk algorithm of selecting senses with the longest gloss by length normalization of vectors. Also

in order to clearly identify role of normalization, metric based on cosine similarity without normalization is experimented.

In this study we use two different similarity metrics which are word overlap count and cosine similarity. And we evaluate the performances of these measurement metrics.

3.5 Performance Considerations

The Lesk algorithm is computationally costly method for WSD as it is mentioned in Chapter 2. There are some studies made like *simulated annealing* [30] which aims to eliminate efficiency problem of the Lesk algorithm. Since in our study, disambiguation of large document collection is one the main tasks, performance of WSD algorithm become more crucial. In order to improve efficiency of the system, we apply some enhancements and get rid of system related bottlenecks. These enhancements are mentioned below.

3.5.1 Format of Turkish WordNet

Turkish WordNet is defined in XML format which is explained in previous chapter. However accessing XML content lots of time and applying filters according to sense id and lexicons reduces the efficiency of the system. In order to apply filtering to XML language using XPath which is XML query language and included in Java SDK [51] is one solution for accessing and filtering WordNet data. However, Xpath parses XML document using DOM parser which firstly fetches all XML content to memory than filtering applied. Fetching all WordNet XML content reduces the efficiency of the system. Also since it is not possible for XML content to define index like in database systems, access performance of the system is not enough for disambiguation of large system. So in order to eliminate performance problems derived from XML format, all Turkish WordNet content and keyword-sense relations are transferred to MySQL database system [52].

3.5.2 Caching Mechanism

In order to improve efficiency of the system, caching is used in processes that require disk access. Caching is applied to sense definition retrieval process for English and Turkish Word-

Net. Due to nature of the algorithm, there is lots of disk access made in order to retrieve gloss of sense, so applying caching to this process aims to improve performance of the system.

Caching is also applied to calculated semantic relatedness of senses. As it is mentioned in Chapter 2, for a sentence that has 10 words and if each word has 10 senses, there are 10^{10} possible meanings available for this sentence. However, situation is not that dramatic if caching is applied. Because, while disambiguating senses of the words in a sentence, calculation of sense pairs is done many times. So by applying caching, calculation of relatedness of sense pairs is assured. For example, context window defined below, words to be disambiguated is given and sense count of these words are given under the words in the Figure 3.5. During disambiguation of $word_3$, calculation of two candidate senses of $word_3$ with senses of $word_1$ and $word_2$ is performed and stored in a map. So sense relatedness computation is performed for just senses of $word_3$, $word_4$ and $word_5$. Computational cost of sentence sense disambiguation is not 240 (3x2x2x4x5), computation cost of the disambiguation of this sentence with caching is 99 (13x3 + 11x2+ 9x2 + 4x5).

$word_1$ (3 senses)	$word_2$ (2 senses)	$word_3$ (2 senses)	$word_4$ (4 senses)	$word_5$ (5 senses)

Figure 3.5: Sample Context Window of Sense Disambiguation

3.5.3 Possible Memory Leak Problems

In order to maintain scalability of the system, memory management and prevention of possible memory leaks become significant, so Java weak references are used for the caching. Weak references can be garbage collected in case of memory needs. Thereby by using weak reference for caching, we improve performance of system and handle possible memory leak problem. However, system can still contain unrevealed memory leak problems.

3.5.4 Efficiency of Sense Similarity Algorithms

To calculate common term count of two text in worst case can be computed in O(n^2). This worst case algorithm can be seen in the Figure 3.6.

```
(1) for each word wi in the first gloss
(2) for each word wj in the second gloss
(3)     if wi equals wj and wj is not marked
(4)         increment similairy by 1
(5)         mark wj
```

Figure 3.6: Inefficient Word Overlap Algorithm

In order to calculate word overlap count between two glosses during implementation of the system, the algorithm mentioned in the Figure 3.7 is employed. In this algorithm, trie data structure is constructed for each gloss. Trie is an ordered tree data structure used to store data with string key values [53]. Trie is generally used in dictionary applications and provides to reach data with string keys in a constant time. Sample trie data structure is showed in the Figure 3.8. We use open source Java trie implementation presented in [54]. This trie implementation is adapted to support Turkish specific characters like ş, ü etc.

While constructing trie for gloss frequency of each term is also stored. Frequency of term is how many times term is repeated in the gloss. Then each unique term in the first trie is searched in the second trie. If term is found in the second trie, increment relatedness value by minimum of term frequency in the first trie and term frequency in the second trie. For instance, if there are two *paper* term in first gloss and three in the second gloss, word overlap count for the *paper* term is two so relatedness is incremented by two. Construction of trie is performed using stem of the gloss words.

```
(1) prepare trie for the first gloss
(2) prepare trie for the second gloss

(3) for each unique word wi in the first trie
(4)     search the second trie as wj
(5)     if wj is found in the second trie
(6)         increment similarity by min(frequency of wi, frequency of wj)
```

Figure 3.7: Proposed Word Overlap Algorithm

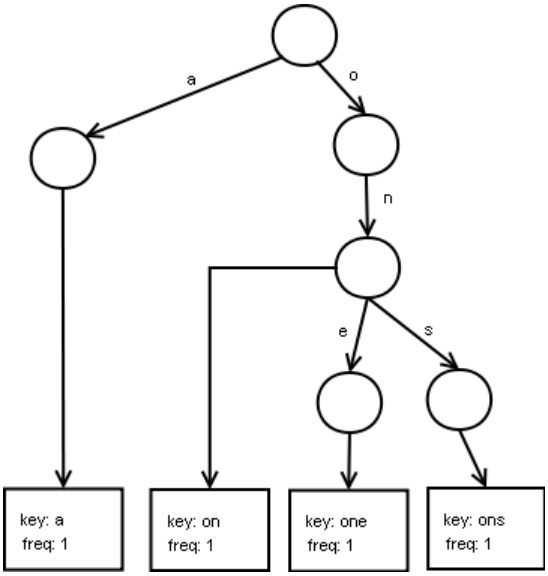

Figure 3.8: Sample Trie Structure For Words: ”a”, ”on”, ”one”, ”ons”

CHAPTER 4

CROSS-LINGUAL INFORMATION RETRIEVAL

Information retrieval (IR) systems are generally composed by three phases which are crawling, indexing and document retrieval. Crawling systems are considered as an important part of search engines, however in this study crawling phase is out scope. Indexing is a batch process, existing document set is indexed by applying classic IR techniques such as cleaning irrelevant character, stemming and stop-word elimination. Indexing is preprocessing phase of the IR system and it is a batch process which does not have any interaction with user. However, retrieval phase involves user interaction. In the retrieval phase, relevant documents are listed as a result of the query using prepared index. Index preparation enables the system to respond back to the user request quickly.

CLIR system aims to provide query-document language independence, so regardless of query and document language; related documents are retrieved as a result of the given query. In this study we focus on Turkish and English text as a demonstration of CLIR system. In this study we develop six different CLIR methods and compare their performance.

In this chapter, firstly overview of the system is given. Then indexing phase for different methods is explained. And finally alternative CLIR methods implemented are described in detail.

4.1 Overview of the System

The developed system consists of two phases which are indexing and retrieval. And English and Turkish Wikinews are used as document collections. In the indexing phase, keyword and sense index for each document collection is created. In the retrieval phase, six alternative

CLIR methods are used in order to retrieve relevant documents from the indices.. For indexing and retrieval, a high performance open source text search engine, Lucene [45], is used. In order to adapt Lucene for this study we make several adaptations on it.

4.2 Indexing

There are four indexes created in the study which are sense and word indexes for Turkish and English Wikinews. Firstly, XML dump of Wikinews is parsed and article content which is in Wikipedia markup language format, is extracted for each document. Then these Wikinews content is converted to the plain text from Wikipedia markup format. In other words Wikinews specific terms and expressions are removed or converted to the plain text correspondence. After, some characters, such as punctuation marks, are removed from the content. After that, word sense disambiguation is applied to the whole content and words are converted to their selected sense ids in English WordNet. Lastly, using Lucene Standart Analyzer, sense id content is indexed. Overview of Turkish sense index is shown in the Figure 4.1.

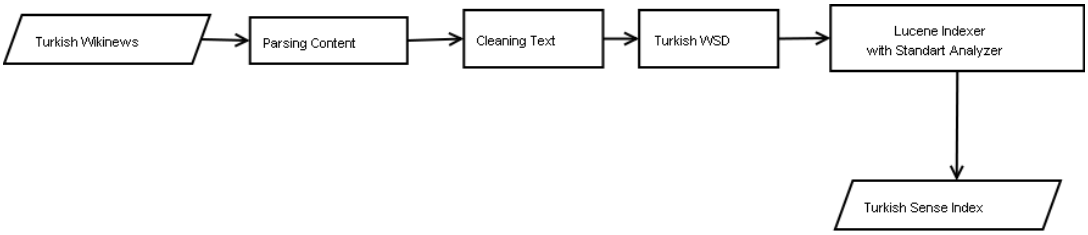

Figure 4.1: Overview of Turkish Sense Indexing

For English sense index English Wikinews is used. Similar steps are applied to the English Wikinews content as Turkish Wikinews. There are some differences in the word sense disambiguation stages. For English content, English WSD is applied using English WordNet. Then by using Lucene Standard Analyzer English sense index is created. Overview of English sense indexing is shown in the Figure 4.2.

After parsing XML contents and cleaning irrelevant characters, for indexing customized Lucene analyzer is used during the creation of Turkish word index for Turkish Wikinews. In Turkish Lucene analyzer, Turkish stop word list is used for stop word elimination. Also Turkish stemming is applied to the content word, after all Turkish word index is created. Turk-

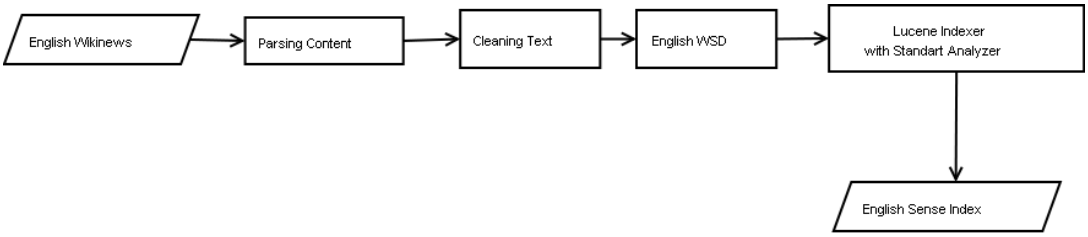

Figure 4.2: Overview of English Sense Indexing

ish word indexing processes is shown in the Figure 4.3. And lastly for English word index, standard analyzer, which is the one of the default analyzer of Lucene library, is used.

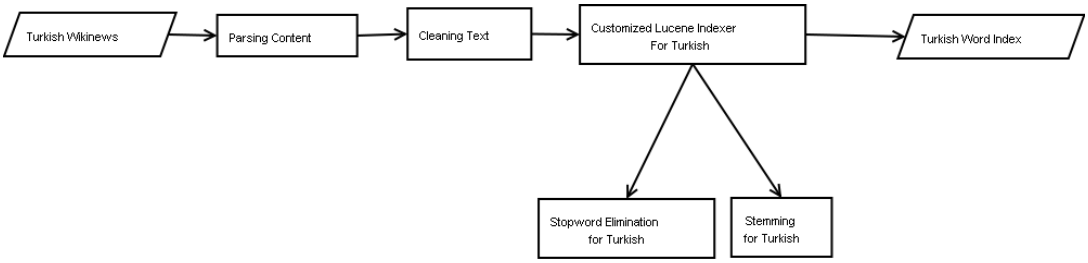

Figure 4.3: Overview of Turkish Word Indexing

4.2.1 Processing Document Collection

XML dumps of Turkish and English Wikinews are used as document collection. In order to use this large XML content, parsing the XML is reassured. The next step is the conversion of extracted Wikipedia content to plain text.

4.2.1.1 Parsing XML Content

In order to parse XML content, event based SAX parser library, which is the standard library in the JDK 6 is used. For large XML processing SAX, parser is preferred instead of DOM parser. This is due to the reason DOM parser, unlike SAX parser, initially loads the whole XML content to the memory which is not feasible for large XML documents. For each article residing in the Wikinews page; id, title, content and time-stamp values are extracted.

4.2.1.2 Eliminating Unnecessary Articles

There are some articles in the Wikinews, which are used to maintain internal structure of Wikinews. These articles are identified by using the start words of the title. Template, category, comment, user, history, redirect and media-wiki pages are eliminated and they are not indexed.

4.2.1.3 Converting Wikinews to Plain Text

Wikipedia articles are constructed with a specific mark-up language. This language includes links, images, bulleted lists and layout. Sample wiki mark-up languge is shown in the Figure 4.4. Indexing articles in this format requires conversion of content to the plain texts. Mark-up language specific expressions are removed from content during the plain text conversion. In order to achieve this conversion, Bliki engine, which is Java Wikipedia engine [55] is used. Bliki engine can make cross conversions between plain texts, HTML syntax and Wikipedia syntax.

# Numbered list # Second item ## Sub item	→	1. Numbered list 2. Second item 2.1 Sub item
Link to [[wikipage]]	→	Link to wikipage
[[URL|linkname]]	→	linkname
== Large heading === Medium heading ==== Small heading	→	**Large heading** **Medium heading** **Small heading**
No linebreak! Use empty row	→	No linebreak! Use empty row

Figure 4.4: Small Set of Wikipedia Markup Language [2]

4.2.1.4 Cleaning Wikinews Specific Terms

Although Wikinews content is converted to plain text, there are still Wikinews specific terms included in the content. These words are eliminated, since they do not carry any meaning and value for retrieval. These eliminated Wikinews specific terms are {*date*}, {*Date*} and {*source*}.

If these terms are not eliminated, they will be the most frequent words in the indices which can lead problem of unrelated document retrieval.

4.2.1.5 Cleaning Irrelevant Characters

Characters that are not included the Turkish and English alphabet and digits are eliminated and words that include the eliminated characters are divided. If there is a word with one letter these words are also eliminated before indexing. For example, word group *John's,balloon* is converted to *john balloon*.

4.2.2 Adaptation of Lucene Indexer for Turkish

Lucene is high performance open source text search engine. And Lucene enables users to customize the indexing process for their special needs by using custom analyzer. In the study, we develop custom analyzers for Turkish and English word indexing. English word indexing analyzer, which is called *StandardAnalyzer* is already implemented in Lucene but we do not used for creation of English word indexing in order to use different stop-word list. Stop-word elimination and stemming processing are conducted in the analyzing step. So in order to index Turkish content, a new *Turkish Analyzer* is implemented. For analyzing Turkish content Turkish stop-word list and Turkish stemmer are used. In addition, *StandardAnalyzer* is employed during the creation of Turkish and English sense indices.

4.2.2.1 Stop-word Elimination

During stop-word elimination process we used stop-word list suggested in the studies [40] and [41].

We previously mentoined in 3, mentioned arguments are still applicable for document indexing. Stop-words are words that are repeated in almost all documents, they do not contain any document specific information. Stop-word list is generated from large corpus by identifying words that is common for all documents. By removing stop-words, we can improve efficiency of data storage and indexing time performance, because stop-words are used in almost all documents in a very high frequency.

Removing stop-words improves retrieval quality of documents, because they are not specific for any page. While retrieving documents, not removing stop words will lead us to get irrelevant documents and will shadow the importance of other words in the query.

4.2.2.2 Stemming

In this study, stemming is conducted for English and Turkish words. Stemming for Turkish words is performed using Snowball stemming library. Snowball is Java based stemming library that contains 15 stemmers for various languages. Turkish stemming algorithm in the library is developed according to the algorithm proposed in [48]. In order to stem English words, we use Porter stemming algorithm which is also included in the Snowball library. Stemming process of Turkish and English words is elaborately described in the Non Dictionary Based Stemming section of the Chapter 3.

Stemming is process of narrowing the given word into its stem. Stem of the word does not have to be identical to morphological stem. In general, suffixes are removed and terms are reduced to the same stem. In other words, stem represents related words in the same surface representation, but this representation does not have to be a meaningful word.

Stemming improves overall performance of IR process in terms of space needs [56]. By removing suffixes, all morphological variations of a term are represented by a single term. Stemming also exposes logical relation of these term variations.

4.2.3 Word Sense Disambiguation

The Lesk algorithm is applied in order to disambiguate senses of English and Turkish words in the document collections. As a complementary study, we comparatively evaluate various alternatives of the Lesk algorithm for Turkish words. There are nine knowledge base alternatives and two similarity metric is used in this study. Details of the word sense disambiguation algorithms are explained in the Chapter 3.

English Wikinews articles are indexed using the Lesk algorithm with cosine similarity. In the original algorithm, calculation of sense relatedness is according to word overlap count of sense glosses. But we prefer to apply cosine similarity metric instead of word overlap

count metric due to its better accuracy on Turkish words. In this thesis, effects of using different similarity metrics on English WSD is not experimented however, better results of cosine similarity metric leads us to use this metric on English texts.

Turkish Wikinews articles are indexed using the English WordNet based Lesk algorithm with cosine similarity metric. This variation of the Lesk algorithm benefits from both Turkish and English WordNet. Senses of Turkish words are extracted from Turkish WordNet but English glosses are used from English WordNet for these senses. Glosses of senses are more established than Turkish WordNet, so that disambiguation relying on English glosses probably has better accuracy.

4.3 CLIR Methods

In the retrieval phase, again Lucene document scorer is used. We benefit from well established with high performance infrastructure. In order to adapt retrieval process fro different CLIR methods, same analyzers are used as in during the indexing. For example, during retrieval of Turkish documents from Turkish word index, customized Turkish analyzer is used.

We implemented six CLIR methods categorized using two approaches which are query translation approach and sense indexing approach. Query translation based methods uses word index of the document collections, sense indexing based CLIR methods use sense index of the document collection. During evaluation of system we prefer to apply Turkish queries to the both English and Turkish document collection since English content is much larger than Turkish document collection,

These alternative methods shares mostly similar steps involving query preparation, index selection and retrieving relevant documents. In query preparation step, users formulate the query and different CLIR methods extract necessary information from this formulated query. Then retrieval and scoring is performed using the Lucene indexer. Main steps of retrieval is shown in Figure 4.5.

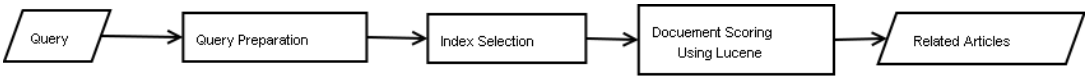

Figure 4.5: Overview of the Retrieval Phase

4.3.1 Query Translation Based CLIR

Query translation based CLIR methods are performed using word index of the document collections. In this thesis four variations of query translation based CLIR is implemented. These methods are synonymy enrichment with manual disambiguation, synonymy enrichment with automatic disambiguation, manually enriched query using Turkish and English WordNets and Google translation. These alternative methods will explained in the rest of this chapter.

In this approach, firstly user query is translated to the target languages then monolingual information retrieval techniques are applied in order to fetch relative documents. This method does not require translation of the all document collection; translation is performed just for the query. Thus query translation approach is more scalable than sense indexing based approach.

4.3.1.1 Synonym Enrichment

In this method, during the query preparation, Turkish and English synonyms of the selected sense of the query words are appended to the query. Since queries are in Turkish, the Lesk algorithm with English WordNet and cosine similarity metric is applied. Then, after sense selection, synonym words of these senses in Turkish and English WordNet are appended to the query. Then document retrieval and scoring process are applied to both English word index and Turkish word index. Since all synonyms of a word in Turkish and English is appended to the query, it is expected to retrieve all related documents in the both languages.

Although, query enrichment process generally improves recall of the system, it reduces precision. Possibility of retrieving irrelevant documents is increased, since all enriched words can have more than one meaning and retrieved documents can be related to the other meanings of the words.

This method requires sense disambiguation for queries. Manual and automatic WSD are

applied to in order to compare effect of WSD on the short queries.

Automatic Disambiguation of Query is performed using the Lesk algorithm once again. However, due to limited context, success expectation of automatic WSD is not high. In case of wrong disambiguation, completely irrelevant documents can be retrieved.

Manual Disambiguation of Query is expected to perform better accuracy, but this method requires more information than automatic WSD. After text query is given by user, each word in the query is stemmed and possible meanings of these words are listed to the user. User selects intended meaning of the terms for the query. Then by using these selected senses, query enrichment process and retrieval processes are performed.

4.3.1.2 Manual Enrichment with WordNet

In this method, system enables users to enrich the query with related words of the query terms. After query is entered, system asks user to manually disambiguate terms by listing senses of the query terms from Turkish WordNet. After performing disambiguation, using English WordNet hypernym, hyponym and synonym relations, related words are presented to user. English WordNet is preferred in order to expose semantic relatedness, since coverage of Turkish WordNet is not as good as that of English WordNet. Hence by using semantic relations of English WordNet, hyponymous, hypernymous and synonymous senses are identified. For each selected sense all words with synonym relation with these senses are manually appended to the query in order to gain cross-lingual nature to the query. It is users responsibility to transform monolingual Turkish query to cross-lingual English and Turkish query by using English and Turkish WordNets.

4.3.1.3 Google Translation

The last query translation method is using Google translation [57] which is provided as a translation service. Google translation is based on statistical translation approach which does not use grammar rules for each language. Translation is made by extracting statistical translation model using two large parallel corpus. [58]. We cannot find any evaluation result for Turkish-English translation results, but according to performance evaluation tests of con-

ducted on Arabic and Chinese in National Institute of Standards and Technology (NIST) in 2005, Google translation achieves to get the best result almost all datasets [59].

Google translation is used, as machine translation alternative for CLIR. So we want to consider the effect of statistical translation approach developed by Google on CLIR for Turkish. In order to obtain cross lingual query sentence, Turkish text query is translated using Google translation and then this translated text is appended to the original query. Retrieval of related documents are conducted using Turkish and English word index.

4.3.2 Sense Indexing Based CLIR

Sense indexing based CLIR retrieves the articles using senses instead of words. In this method, sense indices of the document collections are used. Queries are also converted to the sense ids. Then, retrieval is performed by using query consisting of sense ids.

This method requires disambiguation of senses for given query. Two disambiguation is applied to the queries which are *manual disambiguation* and *automatic disambiguation*. As it is mentioned in the Synonymy Enrichment method, the Lesk algorithm with English WordNet and cosine similarity is used for Turkish queries for disambiguation of senses.

4.3.3 Hybrid CLIR

Hybrid CLIR method is employed that uses both keyword and sense retrieval due to low coverage and completeness problems of the WordNets. These problems yield to low accuracy while identifying sense of the words. Also because of coverage problem, words which are not in the WordNets, cannot be indexed. In order to overcome this completeness problem, a hybrid method which is simply a combination of sense indexing and word indexing methods are used. All documents are retrieved by using manually prepared sense query and the query text itself. Then retrieved documents are ranked according to the retrieval score calculated by the Lucene scorer and first n documents are presented as a result of this process, where n is retrieval evaluation constant. During the evaluation, result of first n documents are presented to the users as a result of retrieval process. Details of evaluation system is explained in the Chapter 5.

CHAPTER 5

DATASET PREPARATION

In this chapter, dataset preparation methods for word sense disambiguation and cross-lingual information retrieval methods are explained. Also evaluation system for CLIR is described. In order to conduct evaluation for CLIR and WSD algorithms, two dataset preparation software are developed. The aim of these software is ease the manual evaluation process for the users.

5.1 Dataset Preparation for Cross-lingual Information Retrieval

In order to evaluate different CLIR methods, we develop a information retrieval evaluation system that eases manual query evaluation process. First step is query creation. In the query creation phase, information is gathered which is necessary for six different CLIR methods. Then system retrieves documents using different query results and presents. Finally users manually define query-document relatedness for given document result. In the rest of the chapter detail of each section will be explained.

5.1.1 Document Collections

Turkish and English Wikinews XML dumps are used as document collection. Wikinews is collaboratively created news source by volunteers. Enghlish Wikinews has started in late 2004 and Turkish Wikinews has started in June 2007. Wikinews dumps contain all articles in XML structures. These dumps are periodically prepared by MediaWiki community for all Wikipedia related projects like Wikinews. In the thesis, we used pages-articles version of Wikinews which do not contain any history of pages, information about English and Turkish Wikinews can be seen in the Figures 5.1 and 5.2. As it is seen, English Wikinews can be

considered as large size document collection comparing Turkish Wikinews.

Start Date	October 26, 2004
Article Count	131000~
Dump Date	February 08, 2010
Type	pages-articles

Figure 5.1: Turkish Wikinews Properties [3]

Start Date	February 03, 2008
Article Count	2800~
Dump Date	March 01, 2010
Type	pages-articles

Figure 5.2: English Wikinews Properties [3]

5.1.2 Query Preparation for Different CLIR Methods

Query preparation is one of the most important parts of the evaluation system. There are six CLIR alternative methods which is mentioned in the Chapter 4. Query creation process, namely collecting necessary information for different CLIR methods from users, there are three steps in query creation.

Query Input: First step of query creation, users are asked to enter the query.

Sense Selection: Second step is sense selection. System parses the given query, and identifies candidate senses for each query terms. In this step users select indented sense of the query terms. If the term is not in Turkish WordNet, word is not listed. If intended sense of the word is not in the candidate sense list, users do not choose any of them. Sense selection window for the sample query, *"Jackson ölümü"*, is shown in the Figure 5.3.

Query Enrichment: The last step is query enrichment which enables users to enrich their queries by using Turkish and English WordNet related words. In this step, system presents related terms in tabular and in graphical form. In the graphical representation, query and related terms are graphically showed to the user so that WordNet relations are explicitly given to the users. Visual query and related word representation is shown

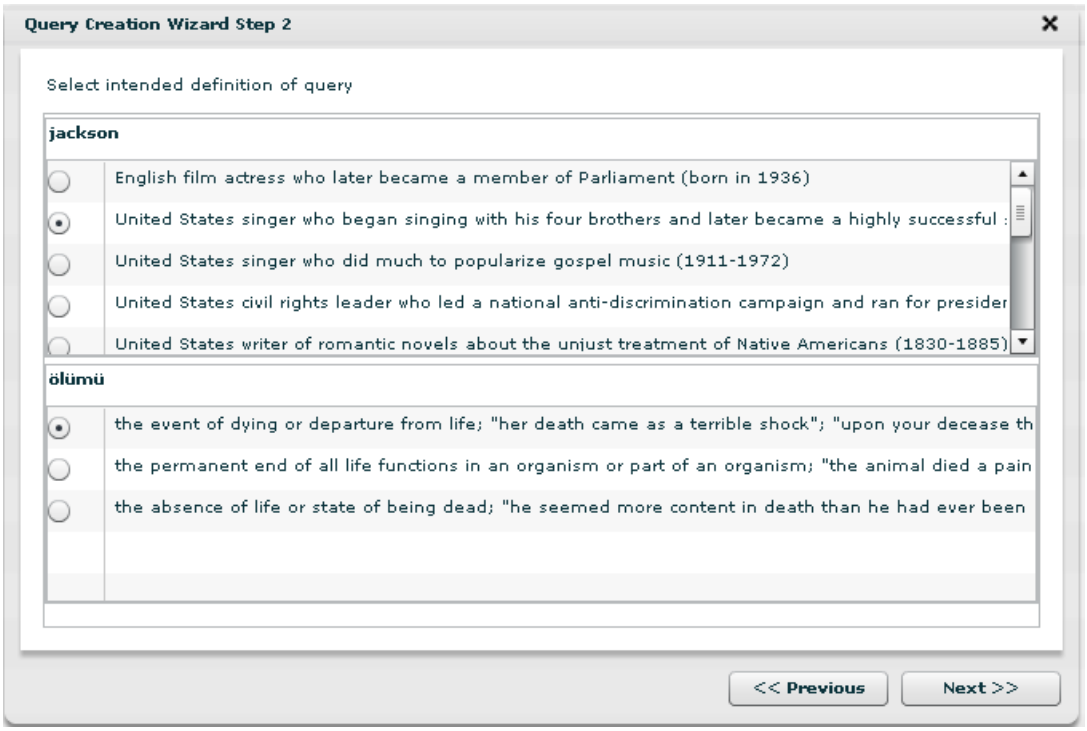

Figure 5.3: Sense Selection during Query Creation

in the Figure 5.4. While visualizing queries, Visualizer Flex library [60] is used. Users add the selected words, by dragging the selected word from graph to the selected word list or double clicking the selected word.

Automatic sense disambiguation based synonym enrichment and sense indexing methods and Google translation based methods uses only the input query. In the manual sense selection based sense indexing and synonym enrichment methods, selected senses of the query terms are used. In manual enrichment method, related word list which is created in the third step and query text is used.

5.1.3 Showing Retrieved Document Results

In the study, first 50 documents that get highest score for each CLIR method is retrieved. If there is not any common document as a result of the CLIR methods, total 300 documents are presented to the users. However, there are some documents are common, so these common documents are merged and there is only one instance of common documents in the result list.

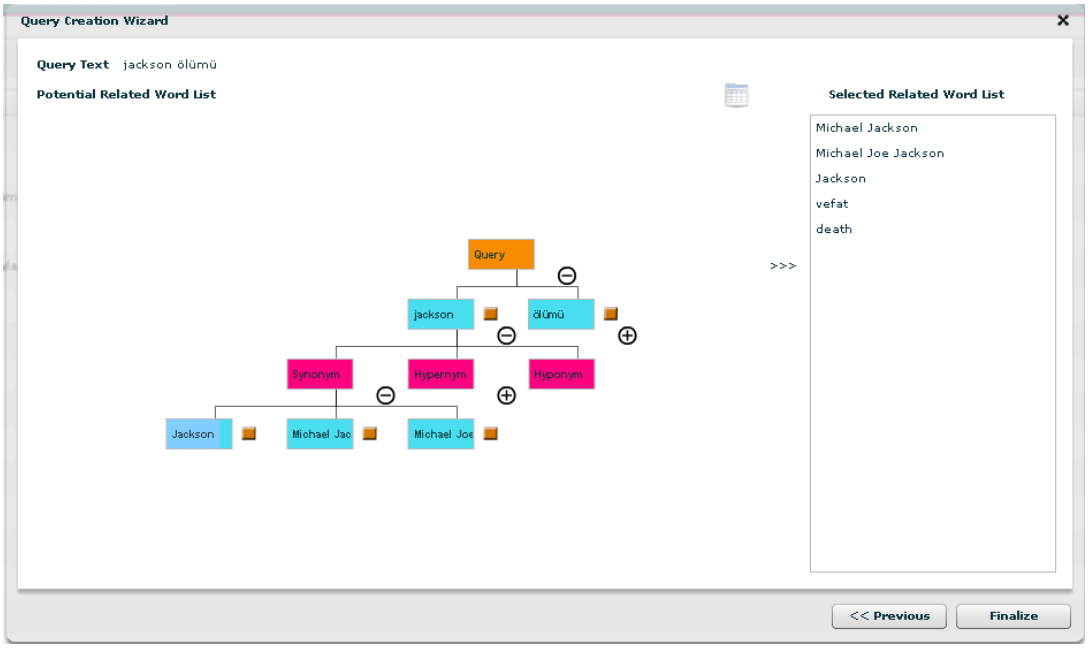

Figure 5.4: Visual Representation of Query

If user marks a document as related to the given query, this evaluation reflected to all CLIR methods that retrieves this document. So that users only evaluate the common documents that are retrieved as a result of more than one CLIR methods.

5.1.4 Defining Query-Document Relatedness

In this process, the system presents all documents retrieved as a result of the each CLIR methods. Then users manually evaluate whether given documents are related to the query or not. There are three option is presented to the users while choosing relatedness, which are relate, unrelated and unknown. Unknown option is used for the documents that do not represent any meaningful content like Wikinews related template or user pages. These documents are actually eliminated during the indexing phase but there can still exist. So these unknown documents are not included while comparing CLIR methods. Query evaluation screen of the system can be seen in the Figure 5.5.

Moreover, the system enables users to have more than one query creation. In the main screen of the user these queries are listed to the users.

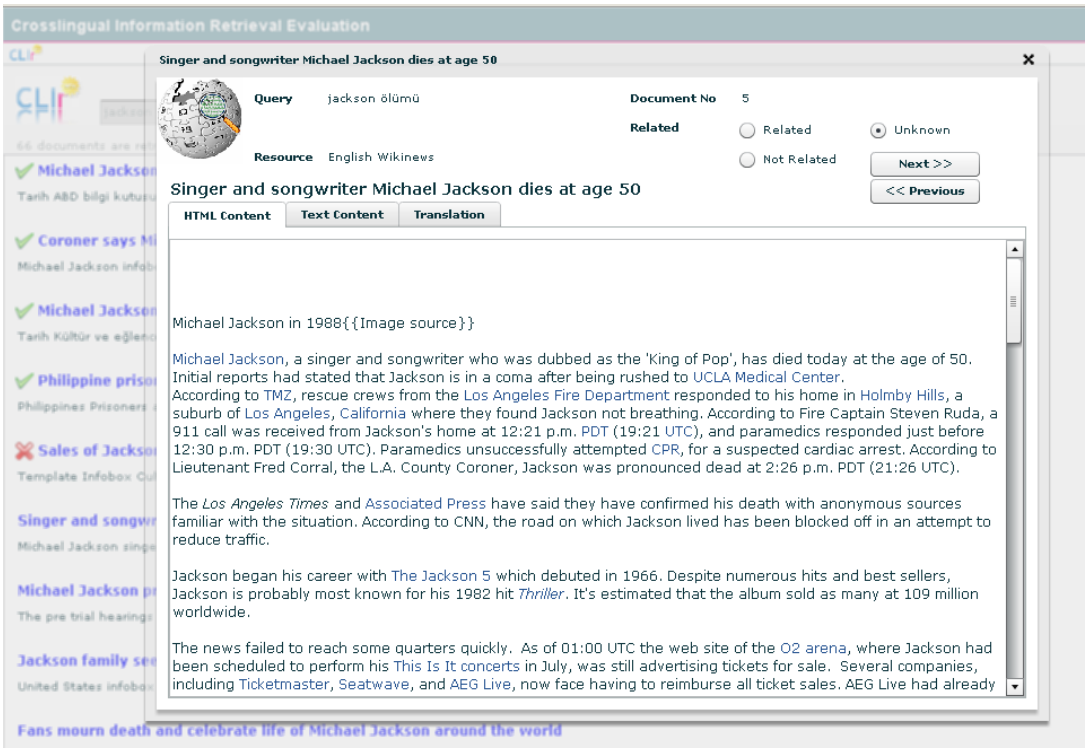

Figure 5.5: Document Evaluation

5.1.5 CLIR Method Evaluation

System presents precision results of first 50 documents of six CLIR methods for each query. Precision values are calculated using method mentioned in the Chapter 6. Users can immediately see comparative result of each query's precision result. Sample screen of the query precision result screen is shown in the Figure 5.6.

5.2 Dataset Preparation for Word Sense Disambiguation

We divided dataset preparation into two parts. In the first parts sentences and candidate words are determined. In the second part senses are manually tagged using senses from Turkish WordNet. In order to be able tagged Turkish sense, it is assumed that users knew English and Turkish, so that users can understand English definition of Turkish words.

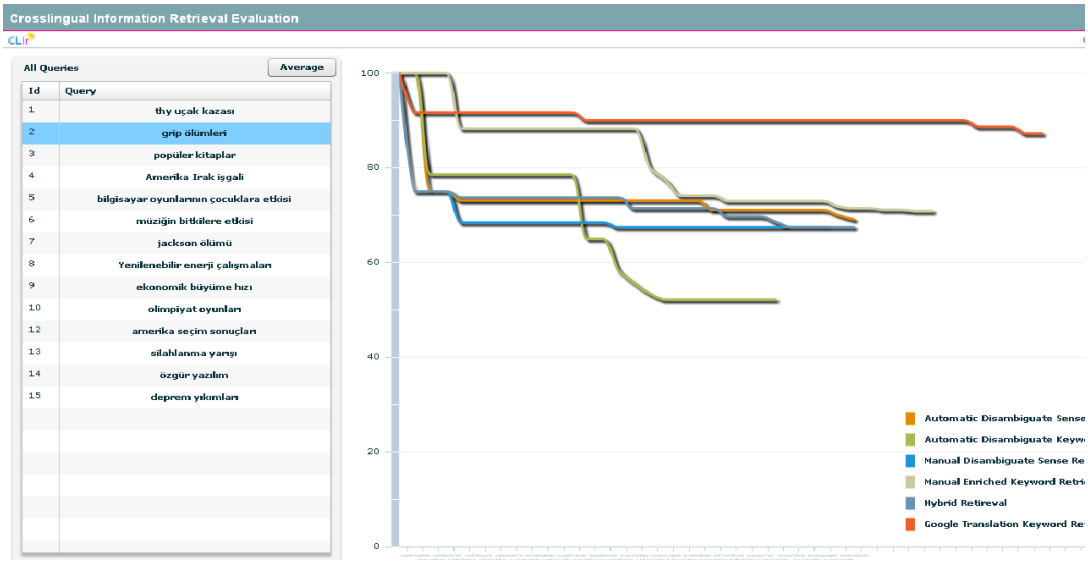

Figure 5.6: Precision Recall Curve of a Query

5.2.1 Preparation of Dataset

Dataset preparation is made by manually tagging senses of the words in the sentences. Firstly, sentences and words are collected from the Internet randomly. Then sense tagging operation is performed. We firstly identify word that is applied WSD then candidate sentences are collected that includes these words.

Candidate words are categorized into two groups according to sense count in the Turkish WordNet. These groups are created by ambiguity of words; it is assumed that more ambiguous words have more senses. First group contains words with two senses, and second group contains words with three senses. There are 200 sentences totally, 100 sentences for each word group.

Developed system accepts the candidate word and context sentence separately which is shown in the Firgure 5.7. During the preparation of dataset, some controls are applied in order to maintain cohesion of dataset. If given words is not included in the sentence, system does not allow saving the sentence. If stem of the word does not in the WordNet or word has only one sense, system does not allow to save the sentence.

Figure 5.7: Dataset Preparation

5.2.2 Manually Tagging Senses

Figure 5.8: Sentence List

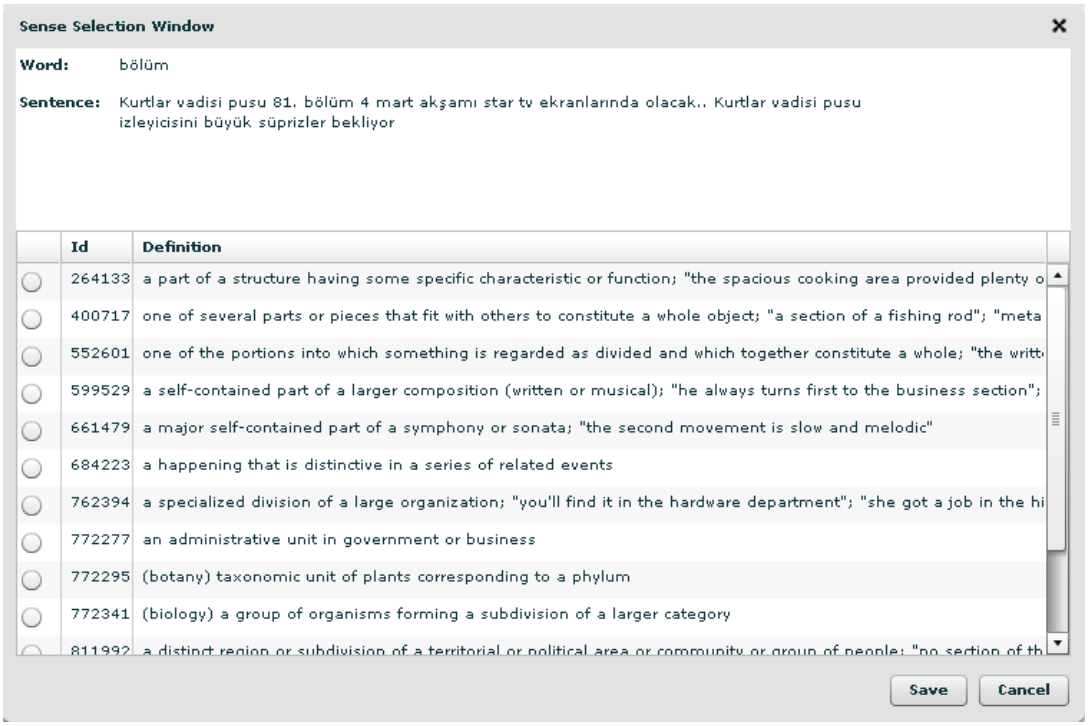

Figure 5.9: Sense Selection Window

Second step is manually tagging the actual meaning of the given words. System presents the sentence pool to the user identifying which sentences are processed which is shown in the Figure 5.8. Then system shows candidate senses for this word during sense selection. Due to missing definitions of some Turkish senses, all senses are presented users with English definitions. Sample sense selection window is shown in the Figure 5.9. Then user identifies the actual sense by applying selection.

CHAPTER 6

CLIR EXPERIMENTAL RESULTS AND EVALUATIONS

In this chapter, details of experiments for comparison of different methods for CLIR will be explained. Firstly evaluation metrics will be explained then experiment results are analyzed in terms of these metrics. For evaluating different CLIR methods, we employ classic information retrieval metrics, which will be detailed in the rest of the chapter.

6.1 Retrieval Performance Evaluation

In order to evaluate CLIR methods applied, query results collected on English and Turkish Wikinews dataset will be used. Information retrieval approaches can be examined in several ways which are retrieval performance, indexing time and space. We firstly focus on *retrieval performance* of the different CLIR methods, which leads us to know accuracy of these methods. During the evaluation steps and metrics are used that is explained in the book *Modern Information Retrieval* [56].

6.1.1 Recall and Precision

Let R be the set of relevant documents in the whole collection, and $|R|$ be the number of documents in this set. Assume that document set A is generated by the retrieval method as a result of information request. And $|A|$ is the number of documents in the set A. Let $|Ra|$ be the number of documents in the intersection of sets R and A. Set A and R is shown in the Figure 6.1.

Precision is the ratio of number relevant documents in the set A over $|A|$. Precision value

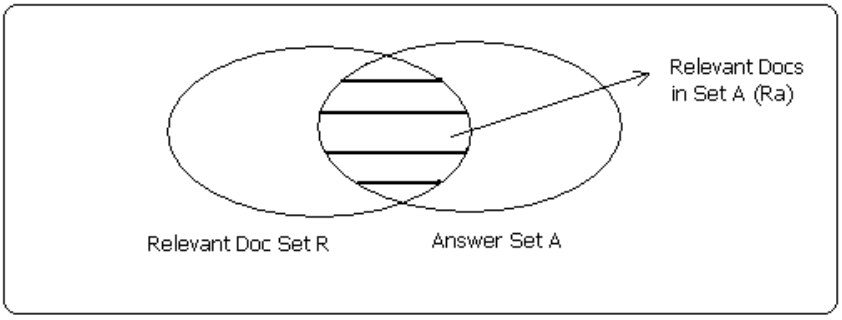

Figure 6.1: Illustration of Document Sets

reveals the accuracy of the retrieval methods.

$$Precision = \frac{|Ra|}{|A|} \tag{6.1}$$

Recall is the ratio of number of relevant documents in the set A over $|R|$. Recall reveals the completeness of the retrieval methods.

$$Recall = \frac{|Ra|}{|R|} \tag{6.2}$$

While comparing different information retrieval approaches, calculating single precision and recall is not always enough. Because, the order of presented document set A is also important and evaluation of all documents is not always possible. Instead of evaluation of all document set, all documents in the set A presented to the user in order according to relevance value. Then users examine top most relevant documents in the list. As users proceeds evaluation of the list, relative precision and recall values is changed. So by plotting precision and recall curve, behavior of different methods can be observed. If more documents are retrieved, recall of the method is increased but precision is decreased. Thus, precision and recall curve is generated as result list evaluated, this precision and recall curve reveals the behavior of different information retrieval algorithms. Some algorithms can have better precision values in lower recall levels, some algorithms in higher recall levels.

As an illustration how precision and recall curve is calculated, we explain an example. Assume Rq is the relevant document set with size 10 for the query q in the document collection.

Aq is the candidate relevant ranked document set retrieved as a result of a CLIR method. Each document is processed from most relevant document, if first document is related, precision value for this step becomes 100% and recall is 10%, then if second document is not relevant to the query, precision is 50% and recall is still 10%.

We calculate precision and recall curve for single query, however, during the evaluation of the systems several distinct queries are used. In order to evaluate retrieval performance of the algorithm over all test queries, average precision values are used in each recall level as shown in the Equation 6.3. In this thesis, average precision and recall curve is calculated according to 11 standard recall level, which are 0%, 10%, 20%.....100%. For all queries, precision values are averaged for specific recall level.

$$\bar{P}(r) = \sum_{i=1}^{N_q} \frac{P_i(r)}{N_q} \tag{6.3}$$

In the equation $\bar{P}(r)$ corresponds to average precision at the recall level r, N_q is a number of queries evaluated, $P_i(r)$ is the precision at recall level r for the i-th query. However, it is not always possible to obtain precision value for specific recall level, in other words it is not always possible to get precision value for, for instance, 10% recall level.. Thus in order to achieve to obtain precision for 11 standard recall levels, interpolation on precision recall curve is applied. *Interpolated precision* at a certain recall level is highest precision calculated in any recall level that is greater than this recall level 6.4. For instance, if precisions calculated in the recall levels are 10 for 60%, 15 for 70%, then if we apply interpolation both recall levels have 15 precision.

$$P(r_j) = max_{r_j<r<r_{j+1}} P(r) \tag{6.4}$$

Interpolation is not only used for calculating average precision value but also it can be used to get more smooth precision and recall curve. The evaluation system developed in the scope of this thesis, average precision - recall curve is calculated by 11 recall levels interpolation, but for the curve of single query just interpolation is applied in order to get more smooth graph.

Furthermore, in this study, since it is not possible to know number of all relevant documents in the collection, all relevant documents retrieved from six different CLIR methods are con-

sidered as relevant document set. This assumption can cause us to get different recall values, but it is still enough to compare these CLIR methods.

6.1.2 Single Value Summaries

Precision and recall curve gives us continuous behavior of different CLIR approaches but information of which algorithm outperforms is still not very clear. In order to reveal this information single value summary is used for each query. In this thesis we employ the *average precision at seen relevant documents* measure. This value is calculated by averaging precision values at the recall level after new related document is retrieved. For instance, assume that set A $\{d_1, d_2, d_3, d_4\}$ is result document set of a retrieval method, and $d_1 and d_4$ is relevant documents to the query, then precision level is calculated when d_1 is processed is 100%, and when d_4 is processed is 50%, so the average precision at this level is 75%. This single value mostly related with quick reply of the system, it is not good in recall value.

6.1.3 Evaluation of Retrieval Performance Results

6.1.3.1 Precision - Recall Curve

We applied interpolation for 11 recall levels in order to calculate average precision recal curve for 14 queries for six different CLIR methods. Precision and recall curve of these methods are shown in the Figure 6.2. According to results almost all recall levels Google translation based CLIR gives the best average precision. Second the most successful method is manually enriched keyword retrieval method until 50% recall level, after this level automatically enriched keyword retrieval leads better results. Experimental result in tabular form is shown in Appendix B.

Surprisingly, the worst method is manually disambiguated sense retrieval method. Automatic disambiguated sense retrieval gives better results than manually disambiguated sense retrieval. In our expectations, manual disambiguation should be the highest limit for the automatic disambiguated sense retrieval. Since words in the documents are also automatically indexed, algorithm have tendency to select same senses. So since document and query is automatically disambiguated, they can share same sense which is not intended sense of the word.

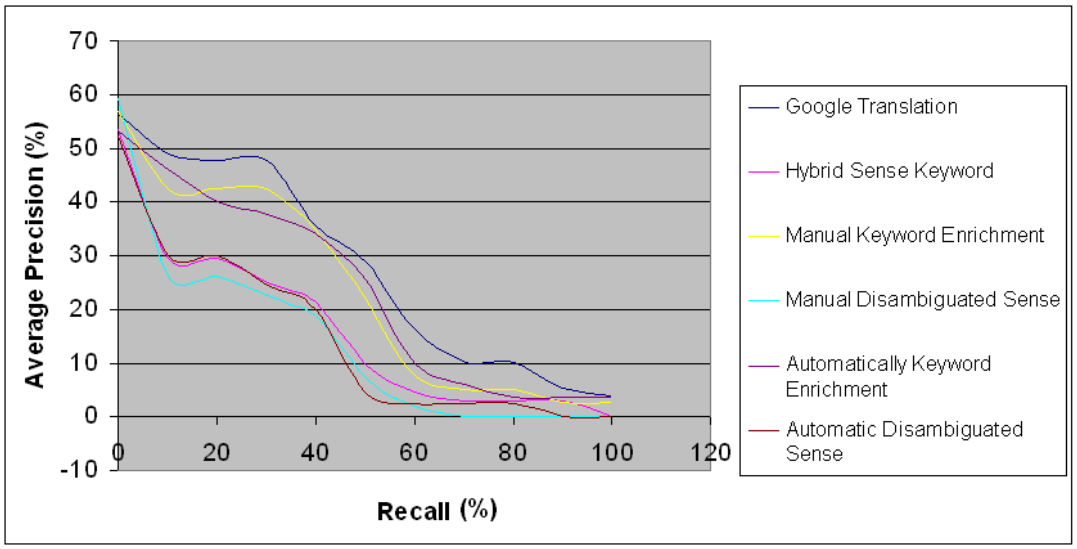

Figure 6.2: Precision - Recall Curve of Six CLIR Methods

Keyword based CLIR methods work better than sense based CLIR methods because all documents are indexed according to senses by automatic sense disambiguation. The accuracy of sense based retrieval methods is limited to accuracy of WSD algorithms.

6.1.3.2 Single Precision Value

Evaluation results using *average precision at seen relevant documents* measure in the Table 6.1, Google translated CLIR method has the best performance among all other methods. Translation quality of Google translation is better than any applied translation and enrichment approach in this study.

Another observation is that again keyword based CLIR methods better results than sense based methods. Among keyword based methods, as expected manual enriched keyword approach has better results than automatically enriched keyword retrieval. The worst method is hybrid (sense-keyword) retrieval, which shows that more intelligent hybrid approach should be applied in order to get better results. In this study, hybrid method is obtained by merging document list of sense and keyword retrieval methods which have the highest relevance.

Comparison of sense based retrieval methods has again same interesting result as it is mentioned before. Automatic sense disambiguation has better average precision value than man-

ually disambiguated sense retrieval. This situation can depend on sense selection tendency of automatic disambiguation algorithm.

Table 6.1: Precision at Seen Relevant Documents Measure

Method	Average Precision
Google Translation	**54.95**
Hybrid Sense Keyword	30.38
Manual Keyword Enrichment	50.23
Manual Disambiguated Sense	30.80
Automatically Keyword Enrichment	47.35
Automatic Disambiguated Sense	37.17

6.2 Performance Evaluations

Indexing performance of the different methods is measure in terms of time and space. Elapsed time for creation of sense and keyword indexing is shown in the Table 6.2. As it can be seen, creation of sense index is 850 times costly than word indexing. Although we try to consider performance of sense indexing system during the implementation, it may not be suitable for large scale system. During the word index, non-dictionary based stemming and stop-word elimination processes are applied. All procedure is completed in a memory, no disc access is required. However, during the creation of sense indexing, for all words dictionary based and non dictionary based stemming are applied, also all sense are fetched many times from both Turkish and English WordNets. If we do not apply caching and algorithmic enhancements results would be worse than this.

Table 6.2: Elapsed Time for Creation of Indices

Indexing Method	Elapsed Time (seconds)
English Word Index	76.5
English Sense Index	64301
Turkish Word Index	3.3
Turkish Sense Index	2757.8

CHAPTER 7

WSD EXPERIMENTAL RESULTS AND EVALUATIONS

In this chapter, we will examine performance of different WSD methods. Firstly, evaluation metrics will be explained and then analysis of WSD result according to these metrics will be stated.

7.1 Evaluation Metrics

For the evaluation of WSD problem, we employ two commonly used statistical metrics during WSD studies. These metrics reveals the accuracy and completeness of different WSD algorithms. Evaluation of different WSD approaches requires hand annotated corpus.

7.1.1 Precision

Precision is a metric that shows accuracy of the system. For WSD study, it corresponds to ratio of correctly disambiguated words over all attempts [61].

$$Precision = \frac{|W_{correct}|}{|W_{allattempts}|} \tag{7.1}$$

where $W_{correct}$ is correctly disambiguated word set and $W_{allattempts}$ is set of all attempts made by the system.

7.1.2 Recall

Recall is a metric that shows completeness of the results. For WSD studies, recall is calculated according to ratio of correctly disambiguated words over all words. If systems achieve to attempt to disambiguate all words then precision and recall of the system will be same. Recall shows completeness of the WSD system.

$$Recall = \frac{|W_{correct}|}{|W_{allwords}|} \tag{7.2}$$

where $W_{correct}$ is correctly disambiguated word set and $W_{allwords}$ is set of all annotated words in the corpus.

7.2 Evaluation of WSD Result

In this study, we compare nine different WSD methods, which uses "Turkish WordNet Gloss" (TRG), "English WordNet Gloss" (ENG), "Hypernym Enriched English Gloss" (HyperG), "Hyponym Enriched English Gloss" (HypoG), "Hypernym and Hyponym Enriched English Gloss" (HHG), "Hypernym Words Enriched English Gloss" (HyperW), "Hyponym Words Enriched English Gloss" (HypoW), "Hypernym and Hyponym Words Enriched English Gloss" (HHW) and "Turkish Wikipedia" (TRWI). Details of these different methods is mentioned in the Chapter 3. While showing evaluation results, abbreviations of methods will be used.

Besides using different knowledge bases and enrichment methods, two different text relatedness measures are used. These measures are word overlap count and cosine similarity. For all different methods that uses different knowledge bases, performance of cosine similarity and word overlap count metrics are evaluated.

Furthermore, in this thesis we employ different derivations of the Lesk algorithm Lesk. And words are disambiguated in a given context. This context is defined by a context window that contains neighbor words. So in order to show effect of window size on different methods, we measure effects of different window sizes on the methods. 9 different window sizes are evaluated and these sizes vary from 2 to 18. For instance, window size 2 means, context consists of only right and left word of candidate word.

In this study we created two annotated corpus that is differentiated according to sense count of the words. Two corpuses contain only words with two and three senses. The corpus that contains words with three candidate senses can be regarded as a more ambiguous set.

7.2.1 Precision Results

Tabular comparison of different WSD approaches in terms of the best precision according to different window sizes is shown in Table 7.1. In the table the best results of different WSD methods are selected for cosine similarity and word overlap count metrics. In order to quickly grasp the performance of algorithms, graphical representation of precision results for WSD methods are show in Figures 7.1 and 7.2. Details of experimental results are shown in Appendix A.

Table 7.1: Precisions of WSD Methods for Different Window Sizes

Method	Cosine Similarity (window size)	Word Overlap Count (window size)
TRG	45.5 (4)	46.5 (4)
ENG	47.5 (12)	44.5 (18)
HyperG	44 (4)	44 (2)
HypoG	44.5 (4)	46 (4)
HHG	46 (16)	46.5 (2)
HyperW	**48 (14)**	44 (4)
HypoW	47.5 (18)	45.5 (6)
HHW	47 (2)	44 (6)
TRWI	46.5 (4)	47.5 (6)

As a control group in the experiment we select first sense in the WordNet and calculate the accuracy. For the corpus that includes words with two sense, first corpus, precision is 50% which is better than most of the algorithms. For the second corpus that has words with three senses, precision of selecting first sense is 32%. When we analyze the results of first corpus, difference and effects of different methods is not very clear. However, evaluations on the second corpus lead us to reach more clear interpretations.

Turkish Wikipedia enrichment and WordNet based methods have better result on first corpus than second corpus. Since almost half of the senses do not have gloss in the Turkish WordNet, accuracy of methods that use these knowledge bases are unpredictable. Words without gloss it is not possible to calculate semantic relatedness between senses. Existence senses without

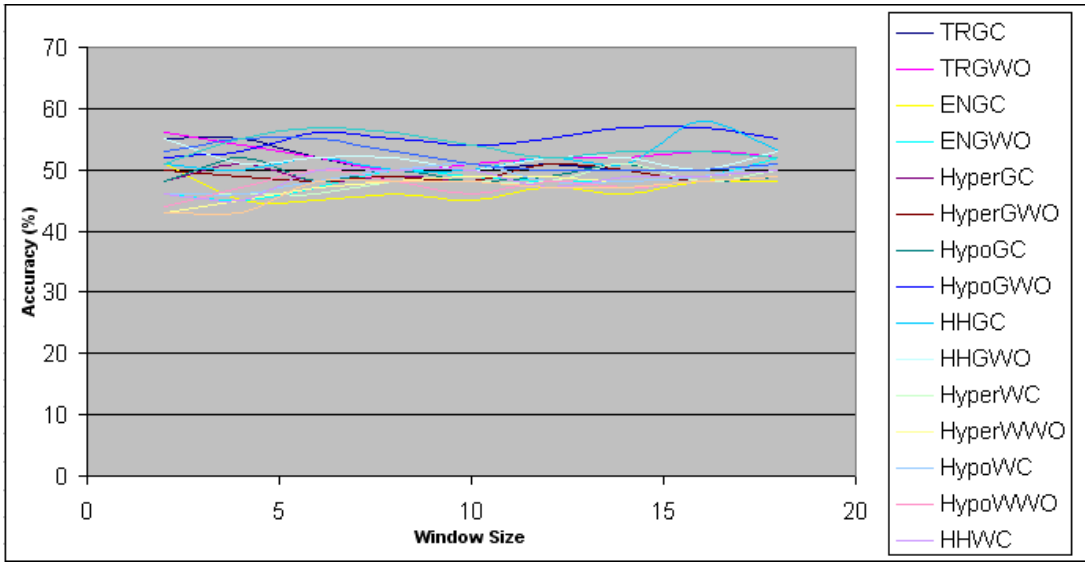

Figure 7.1: Precision By Window Size for Corpus with 2 Senses

gloss affects Turkish Wikipedia based approach since enrichment of gloss is applied relying on this gloss. We can say that there are more senses that have gloss in the first corpus than second, so that Turkish Wikipedia and WordNet based approaches achieve satisfactory results.

Results show that cosine similarity metric outperforms the word overlap count metric. In the second corpus, for almost all English WordNet based methods, cosine similarity metric has better results than word overlap count metric. WSD with word overlap count metric tend to select senses with more gloss words, whereas, cosine similarity metric is normalize the score by gloss length. Hence we can say that cosine similarity metric more accurately exposes the semantic relatedness of senses.

In order to see effect of window sizes, average precision is calculated for all approaches. Average precision values for all window sizes are shown in the Table 7.2. According to average precision, context window with two words has better results. Also the best result in the second corpus is achieved with window size two using English glossary. Hence it can interpreted that the more word close to the candidate word, the more it contains more information on intended sense of the word. However, we can not achieve clear interpretation from all window results mentioned in Chapter chp:wsdexperiments. We have expected to get polynomial precision curve respect to window size but window size results do not show regular behavior.

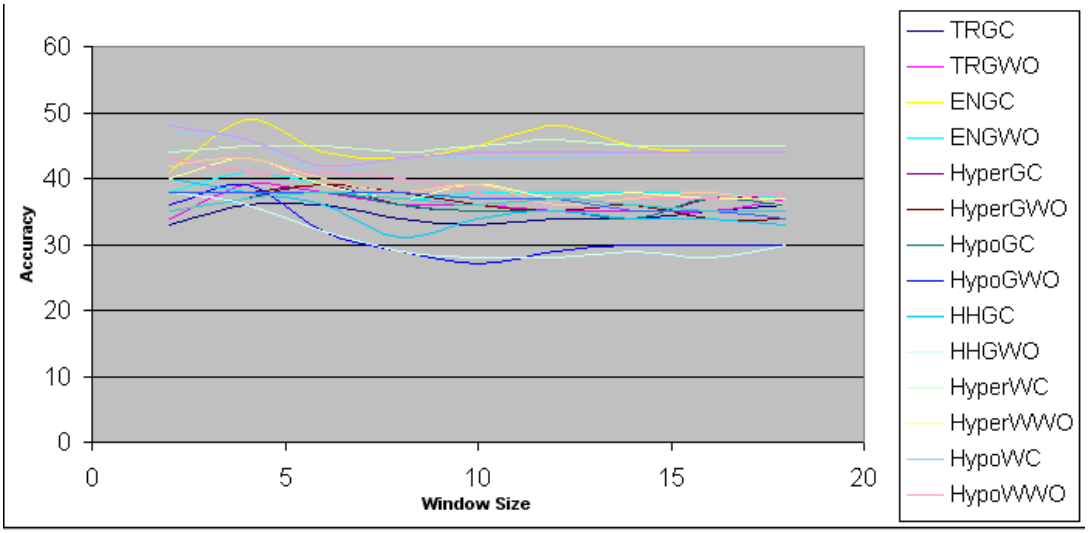

Figure 7.2: Precision By Window Size for Corpus with 3 Senses

We do not need to mention recall results of the methods because there is not any words that can not be applied WSD. WSD dataset preparation software do not allow to save words that do not reside in the WordNet. Therefore, recall of the different is as same as precision values.

Table 7.2: Average Precisions for Different Window Sizes

Method	Window Sizes								
	2	4	6	8	10	12	14	16	18
Average Precision for Corpus 1	49.1	49.3	50.1	50.1	49.4	49.7	50.1	50.2	**50.7**
Average Precision for Corpus 2	39.2	**40.5**	38.7	37.2	37.1	37.1	36.9	37.1	37.1
Average Precision (Overall)	44.1	**44.9**	44.4	43.6	43.3	43.4	43.5	43.6	43.9

CHAPTER 8

CONCLUSIONS AND FUTURE WORK

In this thesis, we compare six different CLIR methods on Turkish and English texts. These methods can be categorized into two main approaches, which are query translation based and sense based approaches. CLIR methods based on query translation based approach is achieved by translation of the user queries into target languages and retrieving relevant documents in different languages for each translated query. In sense base approach, words in the query and whole document collections are transformed into their corresponding meanings and retrieval process is performed using meanings. In different languages, meanings are common although their word representations are different so that performing on senses provides us to retrieve documents independent from query and document languages. In the literature, there are lots of studies conducted on these two CLIR approaches but we cannot find any study for Turkish texts. With this study, we aim to to fulfill the gab in CLIR study conducted for Turkish texts.

During the implementation of this thesis, most of linguistic process is made for both English and Turkish texts. For both language, sense based and keyword based totally four indices are created. While creating these indices, dictionary based and non-dictionary based four different stemming algorithms and two different stop word lists are used. Dictionary based stemming algorithms are used for WSD in the sense based index creation process. For classic information retrieval process it is preferred to use non-dictionary based stemming algorithms due to its performance in terms of time. Performance of the system is a important consideration, because of high computational cost of WSD process and working with 30.000 documents. Furthermore, performing manually query evaluation in order to compare different CLIR methods is another challenge. Hence the evaluation system is developed in order to created manually evaluated query set over document collection. System compares six different CLIR methods and shows graphical comparison results for each query.

In this thesis, six different CLIR methods, which are based on mentioned approaches are performed. First method that is based on query translation is enrichment of the query sentence with English and Turkish synonyms that are identified by automatic sense disambiguation. Second method is enrichment of query sentence with users using related words in the English and Turkish WordNets. In this method, it is users responsibility to gain cross-lingual structure to their queries. Third method is translation of query sentence using Google translation. There are two sense based CLIR methods which are discriminated by manual and automatic sense disambiguation. Sixth and the last method is the hybrid CLIR method, which is based on both senses and keywords.

Sense based information retrieval is a solution for not only CLIR challenge but also synonym and polysemy which are two important information retrieval problem. However, in this study we showed that query translation based CLIR methods have better results than sense based methods. One of the main reasons for result is the low accuracy of automatic sense disambiguation algorithms. Performance of sense based CLIR methods is directly proportional to the accuracy of WSD algorithms. Another reason is coverage of knowledge based which yields to index words that is not included in the knowledge base. Google translation shows the best results among all sense and query translation based methods. Google translation is used as a machine translation system which aims to translate text to a target language without meaning loss with considering grammatical structure of both languages. However, other query translation based approaches naively just deal with query enrichment with words from both languages. So It is the natural result for Google translation method to have better results comparing the other query translation methods.

In the scope of this thesis, we also compare different WSD methods on Turkish words as a complementary study. The main motivation of this study is to observe accuracy of WSD algorithms separately so that it is more convenient to interpret CLIR studies. In this study mainly the Lesk algorithm and its variations are evaluated. In the Lesk algorithm, word sense disambiguation is achieved by calculating relatedness of candidate sense with senses of neighbor words. This semantic relatedness between two senses is calculated by common word count in the glosses of the senses. Using various text enrichment methods, glosses of sense is enriched and effects are observed. During the enrichment process, Turkish and English WordNet and Turkish Wikipedia are used as knowledge bases. The main purpose of enrichment process is to clearly expose semantic relatedness of glosses senses which can

be considered as short texts. Besides word overlap count, in order to calculate semantic relatedness of two senses, cosine similarity metric is used. Each gloss is considered as a vector and cosine similarity between these two vectors is used as similarity of two senses. In evaluation result, we saw that, WSD methods that use cosine similarity have mostly better precision. The main reason is word overlap count metric methods tend to select glosses with more words; whereas, in cosine similarity metric similarity results are normalized by vector lengths so that semantic relations between each sense become more clear. Furthermore, enrichment methods improve average precision of the WSD algorithms and the best results are achieved by using English WordNet.

We aim to make contribution to CLIR studies on Turkish with this study. For the next steps of Turkish information retrieval studies, it would be better to insist on improving sense based retrieval approach. This approach also attacks the polysemy and synonymy problems which can result better retrieval quality. In the future, sense based indexing can improve the accuracy of classic information retrieval methods by employing smart hybrid method. Also improving accuracy of WSD algorithms in the future, the better CLIR results can be achieved.

Another future work is to use better knowledge base in order to overcome coverage problem in sense disambiguation process. The first alternative knowledge base is Wikipedia. There are studies that use Wikipedia as a knowledge base for WSD [25]. In that study, Wikipedia is used as a primary knowledge base and better accuracy in WSD is achieved. Hence similar WSD approach can be employed for CLIR by using both English and Turkish Wikipedia.

The last future work for this thesis is using English Wikipedia while identifying correct senses of Turkish words. In this thesis we use Turkish Wikipedia in order to enrich Turkish gloss of senses, however, this method do not have satisfactory results due to existence of senses without gloss. By using Turkish and English WordNet and English Wikipedia for enrichment process, it is expected to achieve better results. Because every sense in the English WordNet have gloss and completeness of English Wikipedia is more than Turkish Wikipedia.

REFERENCES

[1] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2000.

[2] "Wiki markup language." `http://www.wikicreole.org/`, Mar. 2010.

[3] "Meta wiki." `http://meta.wikimedia.org`, Mar. 2010.

[4] D. W. Oard, "Alternative approaches for cross-language text retrieval," in *AAAI Symposium on Cross-Language Text and Speech Retrieval. American Association for Artificial Intelligence*, 1997.

[5] D. W. Oard, "A comparative study of query and document translation for cross-language information retrieval," in *AMTA '98: Proceedings of the Third Conference of the Association for Machine Translation in the Americas on Machine Translation and the Information Soup*, (London, UK), pp. 472–483, Springer-Verlag, 1998.

[6] A. Pirkola, T. Hedlund, H. Keskustalo, and K. Järvelin, "Dictionary-based cross-language information retrieval: Problems, methods, and research findings," *Inf. Retr.*, vol. 4, no. 3-4, pp. 209–230, 2001.

[7] Fellbaum, *WordNet: An Electronic Lexical Database*. MIT Press, 1998.

[8] S. Stamou, K. Oflazer, K. Pala, D. Christoudoulakis, D. Cristea, D. Tufis, S. Koeva, G. Totkov, D. Dutoit, and M. Grigoriadou, "Balkanet: A multilingual semantic network for balkan languages," in *Proceedings of the First International WordNet Conference*.

[9] M. Lesk, "Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone," in *SIGDOC '86: Proceedings of the 5th annual international conference on Systems documentation*, (New York, NY, USA), pp. 24–26, ACM, 1986.

[10] J.-H. L. Oh-Wook Kwon, I.S. Kang and G. Lee, "Cross-language text retrieval based on document translation using japanese-to-korean mt system," in *in NLPRS*, pp. 101–106, 1997.

[11] M. Sanderson, "Word sense disambiguation and information retrieval," in *SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, (New York, NY, USA), pp. 142–151, Springer-Verlag New York, Inc., 1994.

[12] D. Nguyen, A. Overwijk, C. Hauff, R. Trieschnigg, D. Hiemstra, and F. J. de, "Wikitranslate: Query translation for cross-lingual information retrieval using only wikipedia," in *Evaluating Systems for Multilingual and Multimodal Information Access* (C. Peters, T. Deselaers, N. Ferro, and J. Gonzalo, eds.), Lecture Notes in Computer Science 5706, pp. 58–65, 2009.

[13] R. Mihalcea and D. Moldovan, "Semantic indexing using wordnet senses," in *Proceedings of the ACL-2000 workshop on Recent advances in natural language processing and information retrieval*, (Morristown, NJ, USA), pp. 35–45, Association for Computational Linguistics, 2000.

[14] H. Schütze and J. O. Pedersen, "Information retrieval based on word senses," in *Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval*, pp. 161–175, 1995.

[15] E. Agirre and P. Edmonds, *Word Sense Disambiguation: Algorithms and Applications (Text, Speech and Language Technology)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[16] W. Weaver, "Translation," in *Mimeographed*, pp. 15–23, MIT Press, 1949.

[17] Y. Bar-Hillel, "The present status of automatic translation of languages," in *Advances in Computers*, (New York), Academic Press, 1960.

[18] T. Mitchell, *Machine Learning*. McGraw-Hill, 1997.

[19] W. A. Gale, K. W. Church, and D. Yarowsky, "A method for disambiguating word senses in a large corpus," *Computers and the Humanities*, vol. 26, no. 5, pp. 415–439, 1993.

[20] R. J. Mooney, "Comparative experiments on disambiguating word senses: An illustration of the role of bias in machine learning," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 82–91, 1996.

[21] S. Landes, C. Leacock, and R. I. Tengi, "Building semantic concordances," in *WordNet: An Electronic Lexical Database* (C. Fellbaum, ed.), pp. 199–216, Cambridge, Massachusetts: The MIT Press, 1998.

[22] E. Senseval, A. Kilgarriff, J. Rosenzweig, R. F. E. Senseval, and A. Kilgarriff, "Framework and results for english SENSEVAL," tech. rep., 2000.

[23] D. Yarowsky, "Unsupervised word sense disambiguation rivaling supervised methods," in *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, (Morristown, NJ, USA), pp. 189–196, Association for Computational Linguistics, 1995.

[24] H. Schütze, "Word space," in *Advances in Neural Information Processing Systems 5, [NIPS Conference]*, (San Francisco, CA, USA), pp. 895–902, Morgan Kaufmann Publishers Inc., 1993.

[25] R. Mihalcea, "Using wikipedia for automatic word sense disambiguation," in *HLT-NAACL* (C. L. Sidner, T. Schultz, M. Stone, and C. Zhai, eds.), pp. 196–203, The Association for Computational Linguistics, 2007.

[26] R. Mihalcea and A. Csomai, "Wikify!: linking documents to encyclopedic knowledge," in *CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, (New York, NY, USA), pp. 233–242, ACM, 2007.

[27] R. Navigli, "Word sense disambiguation: A survey," *ACM Comput. Surv.*, vol. 41, no. 2, pp. 1–69, 2009.

[28] H. Schütze, "Automatic word sense discrimination," *Journal of Computational Linguistics*, vol. 24, pp. 97–123, 1998.

[29] D. Lin, "Automatic retrieval and clustering of similar words," in *Proceedings of the 17th international conference on Computational linguistics*, (Morristown, NJ, USA), pp. 768–774, Association for Computational Linguistics, 1998.

[30] J. Cowie, J. Guthrie, and L. Guthrie, "Lexical disambiguation using simulated annealing," in *HLT '91: Proceedings of the workshop on Speech and Natural Language*, (Morristown, NJ, USA), pp. 238–242, Association for Computational Linguistics, 1992.

[31] P. L. F. Vasilescu and G. Lapalme, "Evaluating variants of the lesk approach for disambiguating words," in *Proceedings of the Conference of Language Resources and Evaluations (LREC)*, (Lisbon, Portugal), pp. 633–636, Association for Computational Linguistics, 2004.

[32] S. Banerjee and T. Pedersen, "An adapted lesk algorithm for word sense disambiguation using wordnet," in *CICLing '02: Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing*, (London, UK), pp. 136–145, Springer-Verlag, 2002.

[33] C. Leacock and M. Chodorow, "Combining local context and wordnet similarity for word sense identification," in *WordNet: An electronic lexical database*, pp. 265–283, MIT Press, 1998.

[34] G. Hirst and D. St-Onge, "Lexical chains as representations of context for the detection and correction of malapropisms," in *WordNet: An electronic lexical database*, pp. 305–332, MIT Press, 1998.

[35] P. Resnik, "Using information content to evaluate semantic similarity in a taxonomy," in *IJCAI'95: Proceedings of the 14th international joint conference on Artificial intelligence*, (San Francisco, CA, USA), pp. 448–453, Morgan Kaufmann Publishers Inc., 1995.

[36] J. J. Jiang and D. W. Conrath, "Semantic similarity based on corpus statistics and lexical taxonomy," in *Proceedings of the 10th International Conference on Research in Computational Linguistics*, 1997.

[37] R. Mihalcea and D. I. Moldovan, "A method for word sense disambiguation of unrestricted text," in *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, (Morristown, NJ, USA), pp. 152–158, Association for Computational Linguistics, 1999.

[38] E. Agirre and G. Rigau, "Word sense disambiguation using conceptual density," in *Proceedings of the 16th conference on Computational linguistics*, (Morristown, NJ, USA), pp. 16–22, Association for Computational Linguistics, 1996.

[39] "Priceton university wordnet." `http://wordnet.princeton.edu`.

[40] F. Can, S. Kocberber, E. Balcik, C. Kaynak, H. C. Ocalan, and O. M. Vursavas, "Information retrieval on turkish texts," *J. Am. Soc. Inf. Sci. Technol.*, vol. 59, no. 3, pp. 407–421, 2008.

[41] C. Fox, "A stop list for general text," *SIGIR Forum*, vol. 24, no. 1-2, pp. 19–21, r 90.

[42] "Mit java wordnet interface." http://projects.csail.mit.edu/jwi/, Mar. 2010.

[43] "Zemberek is an open source nlp library for turkic languages." http://code.google.com/p/zemberek/, Mar. 2010.

[44] S. Banerjee, K. Ramanathan, and A. Gupta, "Clustering short texts using wikipedia," in *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, (New York, NY, USA), pp. 787–788, ACM, 2007.

[45] "Full-featured text search engine library." http://lucene.apache.org/java/docs/, Mar. 2010.

[46] "Stemming." http://en.wikipedia.org/wiki/Stemming, Mar. 2010.

[47] "The porter stemming algorithm." http://meta.wikimedia.org, Mar. 2010.

[48] G. Eryiğit and E. Adalı, "An affix stripping morphological analyzer for Turkish," in *Proceedings of the International Conference on Artificial Intelligence and Applications*, (Innsbruck), pp. 299–304, 16-18 February 2004.

[49] "Snowball." http://snowball.tartarus.org/, Mar. 2010.

[50] "Cosine similarity." http://en.wikipedia.org/wiki/Cosine_similarity, Mar. 2010.

[51] "Xpath." http://www.w3.org/TR/xpath/, Mar. 2010.

[52] "Mysql." http://www.mysql.com/, Mar. 2010.

[53] "Trie." http://en.wikipedia.org/wiki/Trie, Mar. 2010.

[54] "Java trie implementation." http://www.koders.com/java/.

[55] "The java wikipedia api." http://code.google.com/p/gwtwiki/, Mar. 2010.

[56] R. A. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1999.

[57] "Google translator toolkit data api." http://code.google.com/intl/tr-TR/apis/gtt/, Mar. 2010.

[58] "Statistical machine translation live." http://googleresearch.blogspot.com/2006/04/statistical-machine-translation-live.html, Mar. 2010.

[59] "Nist 2005 machine translation evaluation official results." http://www.itl.nist.gov/iad/mig//tests/mt/2005/doc/mt05eval_official_results_release_20050801_v3.html, Mar. 2010.

[60] "Flex visualization library." http://code.google.com/intl/tr-TR/apis/gtt/, Mar. 2010.

[61] "Evaluation of methods of word sense disambiguation." http://en.wikipedia.org/wiki/Word_sense_disambiguation#Evaluation_of_methods, Mar. 2010.

Appendix A

WSD EXPERIMENTAL RESULTS

All experimental results are shown for two corpus. Corpus 1 consists of the tagged words with two senses, and corpus 2 consists of the words with three senses. Explanations for abbreviations of WSD results are explained in the Chapter 3.

Table A.1: Precisions of WSD Methods for Different Window Sizes for Corpus 1

Method	Window Sizes								
	2	**4**	**6**	**8**	**10**	**12**	**14**	**16**	**18**
TRG-C	55	55	52	50	50	51	50	50	51
TRG-WO	56	54	52	50	51	52	52	53	52
ENG-C	51	45	45	46	45	47	46	48	48
ENG-WO	46	45	47	50	49	48	48	49	52
HyperG-C	48	51	48	49	48	48	51	48	49
HyperG-WO	50	49	48	49	48	51	50	48	49
HypoG-C	48	52	48	50	48	49	51	48	49
HypoG-WO	52	53	56	55	54	55	57	57	55
HHG-C	51	50	52	50	50	52	51	58	53
HHG-WO	55	51	52	52	50	52	52	50	53
HyperW-C	46	46	46	48	48	48	51	48	50
HyperW-WO	43	45	47	48	49	49	48	49	50
HypoW-C	46	46	48	50	51	48	48	49	51
HypoW-WO	44	47	50	48	46	48	47	48	49
HHW-C	46	45	50	50	50	47	49	49	50
HHW-WO	43	43	48	48	48	47	47	48	49
TRWI-C	53	55	55	53	51	50	50	50	51
TRWI-WO	51	55	57	56	54	52	53	53	52

Table A.2: Precisions of WSD Methods for Different Window Sizes for Corpus 2

Method	Window Sizes								
	2	**4**	**6**	**8**	**10**	**12**	**14**	**16**	**18**
TRG-C	33	36	36	34	33	34	34	35	36
TRG-WO	34	39	38	36	36	35	35	35	37
ENG-C	41	49	44	43	45	48	45	44	44
ENG-WO	38	41	39	37	38	38	38	38	37
HyperG-C	35	37	39	36	35	35	34	37	37
HyperG-WO	38	38	39	38	36	35	36	34	34
HypoG-C	35	37	38	36	35	35	34	37	36
HypoG-WO	36	39	32	29	27	29	30	30	30
HHG-C	40	38	36	31	34	35	34	34	33
HHG-WO	38	36	32	29	28	28	29	28	30
HyperW-C	44	45	45	44	45	46	45	45	45
HyperW-WO	40	43	39	37	39	37	38	37	37
HypoW-C	47	46	41	43	43	43	44	44	44
HypoW-WO	43	41	41	40	38	36	37	37	38
HHW-C	48	46	42	43	44	44	44	44	44
HHW-WO	42	43	40	38	39	36	37	38	36
TRWI-C	38	38	38	38	37	37	35	35	34
TRWI-WO	35	37	38	37	36	37	36	35	35

Appendix B

CLIR EXPERIMENTAL RESULTS

Interpolated average precision values for 11 recall levels are shown in the Table B.1

Table B.1: Precisions for 11 Recall Levels

CLIR Method	Recall Levels										
	0	**10**	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**
Google Trans.	56.2	49	47.7	47.7	35.3	28.8	16.2	10.2	10.2	5.3	3.7
Hyb. Sense-Word	53.3	29.5	29.5	24.9	21.2	9.6	4.5	2.8	2.8	2.8	0
Manual Word Enrich.	56.7	42.2	42.2	42.2	34.7	21.7	7.7	4.9	4.9	2.6	2.6
Manual Dis. Sense	59.3	26	26	22.5	18.7	7.2	1.9	0	0	0	0
Auto. Word Enrich.	53	45.9	40	37.6	33.9	25.3	9.8	5.9	3.4	3.4	3.4
Auto. Dis. Sense	52	30.1	30.1	24.2	20	4.3	2.4	2.4	2.4	0	0