

Vector-Based Semantic Expansion Approach: an Application to Patent Retrieval

Master's Thesis

Golnoosh Farnadi

Vector-Based Semantic Expansion Approach: an Application to Patent Retrieval

THESIS

submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER SCIENCE

by

Golnoosh Farnadi
born in Tehran-Iran



Web information systems (WIS)
Department of Software Technology
Faculty EEMCS, Delft University of Technology
Delft, the Netherlands
<http://wis.ewi.tudelft.nl>

Vector-Based Semantic Expansion Approach: an Application to Patent Retrieval

Author: Golnoosh Farnadi
Student id: 4046412
Email: golnoosh.farnadi@gmail.com

Abstract

Patent collection is increasing incrementally. Most of the new technological information is from patent documents, and retrieving specific patents in such a large pool of documents has become a challenging issue for both patent examiners and normal/inexperienced users. Nowadays, most of the retrieval systems (patent retrieval systems) search for the exact match of the query string inside the collection to return a result. However, based on natural language processing, most words have different meanings and each concept can be expressed by more than one word. Since patent documents include lots of new technical terms, searching among them by exact match needs lots of experiences and it is an exhausting task. In this study we aim to investigate this problem by using expansion techniques. We propose a new semantic expansion approach, namely, vector-based semantic expansion (VSE). To do the expansion, we use external knowledge sources, WordNet and Wikipedia. We also examine the affects of using their combination on the expansion results. We do word (VSWE), query (VSQE), and document (VSDE) expansion in this thesis to demonstrate the performance of our approach. We show that our approach with combination of these two knowledge sources can be effective to find similarity between two units of language which might be word, sentence or text. We apply our technique on Miller and Charles dataset to find word-word similarity. Also, our experiments which are based on clef-ip 2011 shows that our technique increases the recall-rate in query expansion, especially for shorter queries.

Keywords : Semantic expansion, Vector-based similarity measure, Patent retrieval, Document expansion, Query expansion, Word-Word similarity, Semantic similarity measure.

Thesis Committee:

Chair: Prof. dr. ir. Geert jan Houben, Faculty EEMCS, TUDelft
University supervisor: Dr. ir. Laura Hollink, Faculty EEMCS, TUDelft
Committee Member: Prof. dr. ir. Arjen de Vries, Faculty EEMCS, TUDelft

Preface

This is the final work of my master program at TUDelft, where I should say changed my life forever with its opportunities. This document is the result of my work during last year of my study. I am proud of myself for performing this work. I started with an imaginary idea and I have made it real. Although I do not reach my expectations, I have found out the value of scientific works. In this preface I would like to thank all people who helped me to achieve this goal and preparing this thesis.

This project could not have been completed without the guidance, support and feedback of a number of people. First I would like to thank my supervisor, Geert-Jan Houben, for his inspiring and motivating attitude, as well as his critical view on my work. He introduced semantic web to me and helped me to start working scientifically. Second, I would like to thank Laura Hollink who introduced the information retrieval field to me. The field that I find my enthusiasm in it for my future work. Also, I would like to thank Martha Larson for providing valuable feedbacks during my project and let me find my way through patent retrieval. She also introduced me to European Patent Organization (EPO). So I would like to thank Dr. Barrou Diallo, who let me to be more familiar with EPO and patent granting process. Finally, I would like to thank Arjen de Vries, for his support during my thesis and also for being the external committee member for the thesis evaluation.

I would like to express many thanks to my best friend, great supporter and lovely husband, Ashkan, for providing a critical view and constructive feedbacks, support and eternal love during our first year of marriage. Furthermore, I would like to thank all my close friends in Delft for supporting me along this journey.

Last but not least I would like to thank my mother, my father and my sister who are not here with me but without their supports I could not be who I am today.

I hope you enjoy reading my work!

Golnoosh Farnadi
Delft, the Netherlands
November 23, 2011

Contents

Preface	iii
Contents	v
List of Figures	vii
List of Tables	ix
1 Introduction	1
1.1 Context	1
1.2 Research Questions	2
1.3 Approach	3
1.4 Contribution of this thesis	5
1.5 Outline	5
2 Background	7
2.1 Information retrieval (IR) system	7
2.2 IR framework	7
2.3 IR Model	10
2.4 IR Evaluation	13
2.5 Patent retrieval systems	15
3 Vector-based Semantic Expansion (VSE)	19
3.1 Introduction	19
3.2 Expansion Concept	20
3.3 Query Expansion and Document Expansion	20
3.4 Vector-based Semantic Expansion Approach (VSE)	23
3.5 VSWE-WordNet	24
3.6 VSWE- Wikipedia	28
3.7 VSWE-WordNet and Wikipedia	31
3.8 VSDE and VSQE	31
4 Empirical evaluation of VSWE	35
4.1 Introduction	35

4.2	Related works	36
4.3	Word-Word similarity	37
4.4	Choosing values for the VSWE approach	41
5	Retrieval Framework	45
5.1	Introduction	45
5.2	System Model	46
5.3	Pre-Processing	47
5.4	Processing	51
5.5	System Tools	52
5.6	Word Sense Disambiguation (WSD)	55
6	Empirical evaluation of VSQE and VSDE	59
6.1	Introduction	59
6.2	Test Collections	59
6.3	Evaluation approach	60
6.4	Results of VSQE	61
6.5	Results of VSDE	63
6.6	Comparison between results of VSQE and VSDE	65
6.7	Potential of Expansion	66
6.8	Discussion	67
7	Conclusions and Future Work	71
7.1	Revision of Research Questions	71
7.2	Future Work	73
	Bibliography	75
A	Effects of Expansion on similarity value	83
A.1	Same Similarity score after expansion	83
A.2	Increase similarity score after expansion	83
A.3	Decrease similarity score after expansion	84
A.4	Similarity factors	85
A.5	Retrieval Model	86
B	System Features	87
B.1	WordNet	87
B.2	Wikipedia	88
B.3	Dataset	88
B.4	indexing	89
C	EPO Stop Words	91

List of Figures

1.1	Goal Diagram	4
2.1	Document and query set schema	12
2.2	Process of PAC	16
3.1	Semantic expansion schema	20
3.2	Fragment of hierarchy relation of teacher in WordNet	26
4.1	Effects of WSD on results	38
4.2	Effects of different similarity measures on results	40
4.3	Effects of WordNet values on results	42
4.4	Effects of Wikipedia values on results	44
5.1	UseCase diagram	46
5.2	Pre-Processing phase	48
5.3	Indexing GUI	49
5.4	Processing phase	51
5.5	Patent document format	52
5.6	Our Wikipedia Database	55
6.1	Comparison between different data collections based on query/document length, knowledge source and patent parts	62
6.2	Comparison between different data collections based on query length, knowl- edge source and patent parts	63
6.3	Comparison between VSDE and VSQE	65
B.1	WordNet setting	87
B.2	Wikipedia setting	88
B.3	Dataset setting	89
B.4	Indexing setting	89

List of Tables

2.1	Relation between documents and their tokens	8
2.2	The inverted index	8
2.3	DCG example	15
3.1	WordNet statistics	25
3.2	Strength and weakness of using Wikipedia	29
4.1	Word-word similarity results	37
4.2	Effect of WSD on VSWE-Wordnet	39
4.3	Effect of statistic-based measure on VSWE	40
4.4	Correlation results of Word-Word similarity based on different methods	41
5.1	Clef-ip 2011 training set format	53
5.2	Frequency of tokens in each patent parts	54
5.3	Average length of each patent part	54
5.4	Number of entities in our Wikipedia Database	54
6.1	VSQE results based on Query length	61
6.2	VSQE results based on document length	62
6.3	VSDE results based on Query length	64
A.1	Effects of expansion on Cosine similarity measure	85
C.1	English Stop Words	91
C.2	Patent Stop Words	92
C.3	List of all Patent Stop Words	93

Chapter 1

Introduction

1.1 Context

Business models have been changed due to the increase in global competitions; several private companies have adopted the strategy of acquiring, utilizing, and managing intellectual property rights. Therefore, nowadays, patents play a key role in businesses. To retrieve a specific patent or judge the validity of a patent, there is a need for a reliable patent searching application [1]. Patent searching has changed over time, from paper based access systems such as catalogues to on line access systems [2] [3]. Recently, several people within the Information Retrieval (IR) community¹ have been working to address patent retrieval problems [3]. Traditional search methods which are based on patent text and metadata are highly depended on the skill of the searcher and the quality of the database. Therefore, there is a strong need for new methods and methodologies that help inexperienced users to find their results in a short time [3]. One of the reoccurring tasks in patent retrieval is to find similar patents. This is a difficult task for both patent organizations to register a new patent or companies to find patents that are related to their businesses. The Prior Art Candidate (PAC) search is to find patent documents in the target collection that may invalidate a given patent application. PAC is necessary for many strategic issues such as patent granting, freedom to operate, and opposition.

The majority of existing patent retrieval systems use a bag of word (BOW) approach for PAC. In BOW approaches, documents are represented by words inside the document without ordering. We believe that BOW is extremely limited, since it can only retrieve the kind of information that are exactly mentioned in the document with the same vocabulary. Semantic solutions are good candidates for solving this kind of problems and improving the recall while keeping the precision constant or even increasing it [3] [4]. The main purpose of this study is to investigate how and whether the patent retrieval applications can be improved by semantic expansion methods. We make the hypothesis that by using semantic expansion, we can provide high quality results and reduce the workload on experts, in addition to provide a system that can be used by normal users. For this purpose, our main goal is to improve the process of finding

¹New initiatives on patent retrieval include: CLEF-IP, TREC-CHEM, MAREC, PaIR, Apsire, etc, see the IRF website <http://www.ir-facility.org/> for more details, while past initiatives on patent retrieval have been performed by NTCIR, see <http://research.nii.ac.jp/ntcir/> and TREC, see <http://trec.nist.gov>.

similar patents (i.e., PAC) and enrich the retrieval system to provide higher recall-rate. Recall-rate is important because patent retrieval systems are recall-oriented in contrast with most of the retrieval system which are precision-oriented.

To improve the PAC, our aim is to retrieve more similar documents. The process of finding similar patents is time and cost consuming. By extracting more similar patent documents to invalidate the given patent, we can reduce the scope of finding similar documents for patent examiners and reduce the workload on them. Patent experts can study these retrieved documents in order to find the possible conflicts.

By using expansion, we can expand the number of words which are used in query string or collection, and therefore improve the quality of retrieval comparing to BOW approaches. It helps users to retrieve similar documents without using the same vocabulary. By using semantic expansion methods, the goal of expansion is fulfilled by using external knowledge sources. In this thesis we named our semantic expansion approach, which enriches the system by using external knowledge sources, as Vector-based Semantic Expansion (VSE). In the scope of this thesis and for our case study, i.e., patent retrieval system, we enrich our system by using WordNet and Wikipedia as external knowledge sources.

1.2 Research Questions

Searching for similar patents for PAC is an exhausting task and the quality of the results is very expert dependent and non-expert users cannot retrieve useful information from the system. Our goal in this thesis is to address this kind of problems by using expansion techniques. Our general research question that we aim to answer through this study is as follows:

How can we use expansion to improve patent retrieval systems?

We refine this research question itself to five sub-questions. First, we study how useful would be expansion in improving recall-rate.

1. *How can expansion increase the recall-rate?*

We can show in theory that expansion can be useful to retrieve more similar documents and improve the quality of retrieval systems and help expert and non-expert users to retrieve useful information from the system. Therefore, before explaining our novel expansion method, we prove the advantages of using expansion techniques.

2. *Which expansion methods can be used for retrieval purposes?*

We differentiate between query expansion and document expansion as two common expansion techniques for retrieval purposes. We investigate their differences, overview of related work and their usefulness.

3. *Which expansion method can be used for patent retrieval?*

To find the best expansion techniques and methods we study the related work and address this question with two sub questions.

a) *How can patent retrieval systems use expansion techniques?*

We explain how document and query expansion is used in retrieval system. Then, we show how our framework can be used for query/document expansion for PAC.

b) *What kind of knowledge sources can be used for patent retrieval?*

Since our approach for expansion is semantic expansion, we need to use external knowledge sources. By answering this question, we aim to choose suitable knowledge sources for retrieval purposes and specifically for PAC.

After answering the above questions, we answer the following question to show how collection and its characteristics can influence on results.

4. *What is the relation between expansion and data-set?*

Characteristics of retrieval system have a big influence on results. In this study we use patent documents, therefore, by study their characteristics, we can determine the effects of expansion on them. We answer this question with two sub-questions.

a) *What is the effect of query length on expansion?*

We use different parts of patent document for retrieving similar patents. Therefore, we measure the effects of query length on expansion. We extract different query length based on different part of a patent document, therefore, by doing this experiment we can also measure the effects of each part on finding similar patents.

b) *What is the effects of document length inside the data-set on expansion?*

We make six different data collection based on different parts of patent documents. Therefore, we measure the effects of documents length inside the collection on document/query expansion.

5. *What is the effects of expansion on similarity measures?*

The main task in PAC is finding similar patents; therefore it is very important to see how expansion can influence the results. To find similar documents, information retrieval systems use several different similarity measures. To answer this question, first we review related work of similarity measures and then we compare the results of similarity measure based on our expansion approach with other similarity measures on Word-Word similarity dataset. Moreover, to show the effects of expansion on similarity measure, we discuss about the different situations that expansion produces to find similarity.

1.3 Approach

In this thesis, we want to address the problems that we discuss in Section 1.1, by using expansion methods. There are lots of expansions techniques exist; however, to improve the recall-rate and retrieve more similar documents, we choose semantic-based expansion methods. These methods use external knowledge sources to enrich the system and performance. Since we do not have an access to experts to get feedback from

them and do manual expansion, we choose automatic way of expansion. Figure 1.1, shows our approach in this study to solve our research questions in a goal hierarchy diagram. Our goal hierarchy diagram is inspired by goal hierarchy diagram of [5], however, our goal tree is not exactly like the original goal tree.

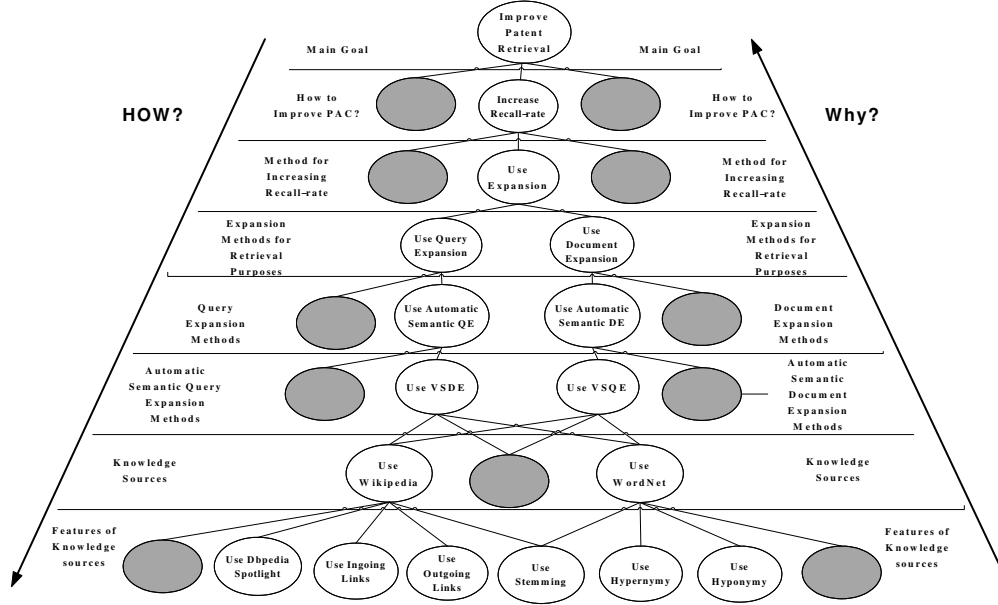


Figure 1.1: Goal Diagram

The diagram shows how we reach our goal. From top to bottom we answer the *How* question and from bottom to top we answer the *Why* questions. Grey ellipses are added to diagram to show although there are other approaches that can answer the *How* questions, we choose one of these solutions in this study.

According to Figure 1.1, our main goal is improving patent retrieval systems by increasing the recall-rate. We do it by using expansion (Word, Query and Document). We use automatic way of semantic-expansion and study the effects of using our approach with Wikipedia and WordNet.

To address the research questions that we mentioned in previous Section 1.2, we study expansion in word, query, and document. Since finding similar patent documents is our goal in this thesis, we compare the differences of semantic similarity measures with our expansion method to show the effects of it on finding similarity. We address the first question, which is about the advantages of using expansion in PAC by evaluating the results on word-word similarity in Chapter 4, we also discuss about increasing the recall-rate by using expansion in Chapter 3 to answer this question. To understand which expansion techniques are useful for PAC and how query/document expansion can improve the PAC, we introduce these two techniques namely VSQE and VSDE, discuss about their advantages and their differences in Chapter 3. We also explain how these two techniques are used in retrieval systems in Chapter 5. To choose the suitable knowledge sources for expansion in PAC, we study the current way of doing PAC in Chapter 3. Finally, to address the effects of data-set and determine the usefulness of expansion for PAC, we evaluate our method by demonstrating the experiments and

discussing their results in Chapter 6.

1.4 Contribution of this thesis

Our contributions in this thesis are followings:

1. We propose a novel kind of semantic expansion method, in which we represent each unit of language (word, sentence, text, etc) by a vector. This way of representation helps us to combine expanded vectors of different sources (e.g. Wordnet and Wikipedia). Since each knowledge source has its own features and characteristics, we believe that by combining these knowledge sources, we can use the advantages of each source to have a more complete source of knowledge for our expansion.
2. We implement our semantic expansion method using WordNet. In our method we use the IS-A relations. We represent each unit of language by its words and expanded words from WordNet. For measuring the effects of un-uniform shape of WordNet, we add some weights to the links that can be changed based on the distances from the original word in the graph.
3. We implement our semantic expansion method using Wikipedia. In our method we use the hyperlink structure of Wikipedia. To find the best links for expansion we use the number of ingoing links for each page. To expand each word we use the Wikipedia page title and page titles of ingoing and outgoing links. We also add weights to the link in the same way that we use for our WordNet approach. Similar to the WordNet method we represent each unit of language by a vector to have a good performance. We also use Dbpedia spotlight for expanding texts to find a correct page in Wikipedia.
4. We implement our semantic expansion method using WordNet and Wikipedia. Our approach let us combine two knowledge sources with each other, therefore we combine our Wikipedia approach with WordNet approach to experiment the effects of combining two different knowledge sources on similarity and retrieval performance.
5. We propose a framework for document/query expansion. We use this framework for PAC. In our framework we implement the two main phases of retrieval systems, processing and pre-processing. We explain all the steps which are needed to construct such a system, especially the indexing system.

1.5 Outline

The rest of this paper is organized as follows: in **Chapter 2**, the background information about information retrieval systems, common techniques for retrieval is presented. At the end of this chapter, we also present brief introduction to patent retrieval systems and prior art candidate search. Expansion and our novel approach based on semantic expansion via WordNet and Wikipedia are presented in **Chapter 3**. Moreover, To use our semantic expansion method for retrieval purposes, we use query expansion and

document expansion techniques, therefore these two techniques and our approach for document expansion and query expansion are also presented in Chapter 3. In **Chapter 4**, similarity measures in addition to prior work and related literature in this area are discussed, and we compare our expansion method with other semantic based measures based on word similarity. To implement the whole retrieval system based on our method, we design a system model, in **Chapter 5**. In **Chapter 6**, our experimental setup and results are described in details and at the end of this chapter the weakness and strength of using expansion and our approach for document retrieval purposes are discussed. **Chapter 7** concludes the thesis and shows the direction of our future work.

Chapter 2

Background

In this chapter, first we briefly explain the concept of information retrieval (IR) and its framework followed by presenting IR techniques, IR models, and IR evaluation. Patent retrieval is our specific case to study information retrieval systems, and therefore at the end of this chapter, we describe patent documents, the characteristics of patent retrieval systems and the purpose of prior art candidate search in patent retrieval.

2.1 Information retrieval (IR) system

In general, information retrieval is used to help users find relevant information according to their needs. To extract the required information, a user has to follow an active information seeking behavior by submitting a query to an information retrieval system. A query that provides the information the user is looking for, can be very difficult. Normal users of WWW might try to query several times to find the relevant information to their needs. The results are dependent to the user experience and the domain. For example in patent retrieval systems, normal users cannot get the correct information since it is very expert dependent and it is very hard for them to submit a correct query. Formulating a query for most of the retrieval systems needs a good knowledge about the whole system, and in this way, they are very expert dependent.

Although the main goals of all retrieval systems are the same, these systems are very different in their characteristics; there are lots of different retrieval systems such as image, video, text retrieval, etc. For each of these systems, a specific query is needed. In most of the text and document retrieval systems, some techniques and approaches are the same, however each retrieval system change the techniques based on its own requirement. In the following sections we explain some general techniques in the text retrieval systems.

2.2 IR framework

For retrieval purposes, not all the words that are used in a document are needed. Some words are more meaningful and more important. To recognize documents to be retrieved based on a user query, documents should be stored in an easy recognizable way. The terms that we extract from the document for further retrieval purposes are named `index terms`, and the approach of splitting the text into tokens is called `tokenizing`,

then by lexical analysis extracted terms from the text become an index terms. For constructing an index system we have to extract the meaningful terms from a document. There are several techniques available to extract important terms and build an indexing system. Here, we explain the indexing systems and some of the most common techniques for extracting index terms such as lexical analysis, deleting stop words and stemming.

2.2.1 Indexing system

The main purpose of using an index is to speed up the retrieval process. Indexes are used in the classical retrieval systems such as databases up to search engines of the WWW. Without using an index, retrieval systems must scan all documents to find the relevant ones. There are several data structure for building an index system. One of the well-known data structure is *inverted index* [6]. In this structure, after finding the important terms inside a document, instead of storing the list of tokens for each document, a list of documents for each token is stored. For example, assume that we have three documents. Table 2.1 shows the relation between each document and their tokens:

Doc name	tokens
<i>doc₁</i>	it, is, a, book
<i>doc₂</i>	this, book, is, mine
<i>doc₃</i>	my, book, is, here

Table 2.1: Relation between documents and their tokens

Table 2.2 presents an inverted index table for our example:

token	Doc names
it	<i>doc₁</i>
is	<i>doc₁, doc₂, doc₃</i>
a	<i>doc₁</i>
book	<i>doc₁, doc₂, doc₃</i>
this	<i>doc₂</i>
mine	<i>doc₂</i>
my	<i>doc₃</i>
here	<i>doc₃</i>

Table 2.2: The inverted index

Using inverted index, speed up the process of finding which words are used in which documents. The typical data structure for this model is hash table and Binary tree [6].

Lexical analysis:

First step for indexing documents is lexical analysis. Lexical analysis is to transform a stream of text to stream of index terms. In retrieval systems, the first step for splitting the text in to tekens is called *tokenizing*. In this step, the system gets the whole

text and splits it into terms which will be used after processing by lexical analysis as index terms in the system. To process tokens and identify index terms, retrieval systems use lexer. Tokens are defined by regular expressions; therefore lexers can read and understand them. Although lexers are different from one language to another one, there are common rules in all languages. For example, one of the common word separator between all languages is space. In lexical analysis there are four cases which are very important for word separation. These four cases, can show how much the retrieval systems are domain dependent. These cases are enumerated as follows:

1. **Digits:** In most of the retrieval systems, identifying digits as an index and add them to index system is not very useful. A digit has a meaning inside the text, however without the text around it, it does not have a special meaning such as H₂O. Therefore, it is wise for most of the retrieval systems to disregard numbers as index terms.
2. **Hyphens:** The other case that is important in lexical analysis is to ignore the hyphens. In most of the cases, the hyphens do not affect the results of retrieval system such as twentieth-century and twentieth century, however, there are some cases like B-342 which loss their meaning without hyphens, or hyphens at the end of sentences for connection between two parts of one word.
3. **Punctuation marks:** Like hyphens, punctuation marks are ignored in retrieval systems, however, they also have a special cases and they are language dependent. For example, if there are a explanation of code inside the text like *x.id* it is wise to not ignoring them.
4. **Lower and upper case:** There is no difference between the different appearances of one word in most (upper case or lower case) of the retrieval systems. Most of the retrieval systems transform all the words to lower case at first, however in some documents like documents about Linux operating systems, the case of the letters have different meaning and ignoring the different appearances of a word might be harmful [7].

Deleting stop words:

There are lists of words in each language and also in each domain which is used generally in most of the documents. Based on processing of natural languages, we can eliminate them without losing any meanings. Generally, words that are used in more than 80% of documents are not useful for retrieval purposes (indexing system). Deleting these stop words before starting the indexing system leads to decreasing the size of indexing system and processing time of retrieval.

Stemming:

The other techniques that information retrieval systems use to retrieve more relevant results is stemming [8]. Stemming is the procedure of transforming a word to its root or stem. There are several algorithms for stemming. For example by stemming, *fished*, *fishes*, *fishing* or *fish* transform to their root which is *fish*. So in such a system,

if users search for *fish*, they can get documents contain *fishing*, *fish*, *fishes*, and etc. in its content.

2.3 IR Model

To represent documents in information retrieval in an efficient way, several different models are available. Various models are categorized based on their mathematical basis or based on the relations between words. In this section we present the main three classic models of information retrieval and then briefly explain their characteristics. These three models are basic models which differ based on their mathematical basis.

2.3.1 Boolean Model

The boolean model [9] is based on linear algebra and set theory. In this model documents are represented as sets of index terms. An index term in the set can have a value of zero if it is available in the document and a value of one if it is not available in the document. In this model the queries are specified as boolean expressions. A query in a form of boolean expression is a combination of index terms and three connectors: *And*, *Or* and *Not*. Most of the users of this model, find it too difficult to express their needs with boolean expression. The main advantage of using the boolean model is its simplicity. However, this model has drawbacks. By using boolean model, only the exact match can be retrieved from the system. Documents are either relevant or non-relevant, which can lead to retrieve too few or too many results. In this model, the system cannot retrieve documents with partial relevancy to a query. Furthermore, in the boolean model, index terms do not have any weights inside a set. Weights can show the importance or non-importance of the terms inside the documents and can influence their relevancy to the query.

2.3.2 Vector Space Model

Another model which is used for information retrieval purposes is the Vector Space Model (VSM) [10]. In this model documents are represented as m-dimensional vectors in which index terms are weighted. One of the main advantages of using VSM instead of boolean model is the ability of retrieving partial-match. Since in this model the similarity between the query and documents is scalar, retrieval system shows the ranked-result to the user. Therefore the result of VSM is more adopted to user needs than the results of boolean model. There are many ways to determine the weights of index terms inside a document vector. Salton et al.[11] discusses different techniques of term-weighting. One of the well-known techniques is *TF/IDF*, which means *Term Frequency* inside a document by *Inverse Document Frequency*, which is the frequency of the term inside the collection. Frequency of terms inside the document is not enough to measure the importance of the terms inside the document, most of the frequent words are used in most of the documents and therefore we have to distinguish in weighting procedure. The main idea of using *IDF* is to distinguish them between relevant documents from non-relevant documents based on terms which appear frequently in the documents. Equation 2.1 shows the term frequency which represents

the importance of the term t_i within a particular document d_j , where $n_{i,j}$ is the number of t_i occurrences.

$$TF(t, d) = \sum \text{tokens } t \text{ in the document } d \quad (2.1)$$

IDF is expressed by Equation 2.2. It measures the importance of term t_i in corpus D .

$$IDF(t) = \log \frac{|D|}{|d : t \in D|} \quad (2.2)$$

Where, $|D|$ shows number of documents in the collection and

$$|d : t \in D|$$

shows number of documents in the collection which have term t . Therefore, TF-IDF is calculated as follows:

$$Tf - IDF = tf(t, d) \times idf(t) \quad (2.3)$$

The similarity between a document and a query in this model is calculated by using the cosine similarity which is explained later in next part: Vector-based similarity measures. Nowadays, VSM is very popular in retrieval systems, because i) this model is fast and simple, and by using weights for terms inside a document, improves the performance, ii) retrieving documents with partial match to the query helps users to find relevant documents, iii) the result is sorted based on the similarity of documents to the query, which is very useful for users to find relevant documents based on their needs easily. This model is very effective and most of the retrieval systems utilize it, however, improving the ranked results in this model is not very easy without using expansion or relevance feedback [7].

Vector-based similarity measures

Here, we introduce three most common similarity measures based on vectors. Most of the IR systems use these measures for calculating the similarity between query and documents. At the end we discuss their differences.

Cosine similarity: One of the most common measures for finding similarity between two vector is cosine similarity. In this method documents are represents as a multi dimensional vector and each token in the vector is represents one dimension [12, 13].

$$\text{Cosine Similarity } (v_1, v_2) = \frac{\sum_{r=1}^N v_{1r} \times v_{2r}}{\sqrt{\sum_{r=1}^N v_{1r}^2} \times \sqrt{\sum_{r=1}^N v_{2r}^2}} \quad (2.4)$$

Extended Jaccard: Another common similarity measure is Extended Jaccard similarity. Since the idea of this measure is from Jaccard similarity, the name of this measure is extended jaccard. The differences between jaccard and extended jaccard is that Jaccard can only used for binary vectors and the results is zero and one [14].

$$\text{Extended Jaccard Similarity } (v_1, v_2) = \frac{v_1 \times v_2}{v_1^2 + v_2^2 - v_1 \times v_2} \quad (2.5)$$

Dice Coefietient: Dice method can calculate vectors of weighted association by using Jaccard measurement as the denominator normalization factor of the total weighted value of non-zero entries in two vectors [15].

$$\text{Dice Similarity } (v_1, v_2) = \frac{2 \times \sum_{r=1}^N \min(v_1, v_2)}{\sum_{r=1}^N v_{1_r}^2 + v_{2_r}^2} \quad (2.6)$$

The main differences between these three measures for information retrieval is shown in Figure 2.1. To make it simpler, we consider document as a set of terms as D and query as Q:

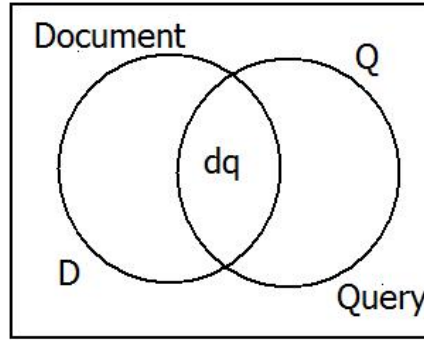


Figure 2.1: Document and query set schema

Therefore these three different measure is defined as follows:

$$\text{Cosine Similarity } (D, Q) = \frac{2 \times |dq|}{\sqrt{|D||Q|}} \quad (2.7)$$

$$\text{Extended Jaccard Similarity } (D, Q) = \frac{|dq|}{|D| + |Q| - |dq|} \quad (2.8)$$

$$\text{Dice Coefietient}(D, Q) = \frac{2 \times |dq|}{|D| + |Q|} \quad (2.9)$$

2.3.3 Probabilistic Model

Another type of retrieval models are probabilistic models which are based on probability of relevancy between documents and a query. The mathematical basis of this retrieval models are probabilistic theorems such as the Bayes' theorem. One of the most well known probabilistic models is the one that first Robertson et al. introduced in [16] which known as Binary Independence Retrieval (BIR) model. In this model, the relevancy of a document to a query is shown as a ratio of probability of relevancy to probability of non-relevancy. By assuming that index terms are independent in the documents, this ratio is the probability of relevancy of each term. The result can be improved in an iterative process over the time. At first, the probability of relevancy or non-relevancy of documents and the query is calculated based on an assumption, for example 50% of relevancy and 50% for non-relevancy. However, after a while by

repeating the procedure and changing the threshold of the model, the result can be improved without human assistant. Changing the threshold in each round is based on the probability of words occurrence in first r results of a previous round [7]. Probabilistic models are not simple, however, models in this category can recursively improve the results.

2.4 IR Evaluation

In retrieval systems the result are not exact answers to the user query, therefore there is a need for evaluation to show how good the results are. There are lots of methods which are used to measure the retrieval system's performance. These methods are based on test collections with some examples of user requests (queries) and sets of relevant documents provided by specialists for these queries. Recall and precision are important measurements that are used in information retrieval to estimate the accuracy of a retrieval algorithm. In this section, these two well-known techniques are explained and then four alternative measures, average precision, MAP, DCG and NDCG are presented.

2.4.1 Recall

Recall captures for relevancy of the retrieved documents. Recall is used for completeness; it shows the relevancy of retrieved documents by the actual relevant subset within data-set. Recall is calculated by the following equation:

$$Recall = \frac{|\{Relevant\ documents\} \cap \{Retrieved\ documents\}|}{(|\{Relevant\ documents\}|)} \quad (2.10)$$

Relevant documents are those documents which are relevant to the user's query.

Retrieved documents are those documents which are extracted by the retrieval system for the user's query.

2.4.2 Precision

Precision is used for exactness of the retrieved documents; it shows how much the retrieved documents are relevant. The precision is calculated by the following equation:

$$Precision = \frac{|\{Relevant\ documents\} \cap \{Retrieved\ documents\}|}{(|\{Retrieved\ documents\}|)} \quad (2.11)$$

2.4.3 Average Precision

Recall and precision are single value metrics. For retrieval systems which return ranked list results, it is better to take this ranking into account to measure the performance of a system. Recall and precision do not measure it. In average precision the relevant documents with better rank gain more weights and it is calculated as follows:

$$Average\ Precision = \frac{\sum_{r=1}^N (P(r) \times Rel(r))}{(|\{Relevant\ documents\}|)} \quad (2.12)$$

$$P(r) = \frac{|\{\text{Relevant documents of rank } r \text{ or less}\} \cap \{\text{Retrieved documents of rank } r \text{ or less}\}|}{r} \quad (2.13)$$

$$Rel(r) = \begin{cases} 0 & \text{If } r \text{ is not relevant} \\ 1 & \text{If } r \text{ is relevant} \end{cases} \quad (2.14)$$

2.4.4 Mean Average Precision (MAP)

In retrieval systems, the quality of results are measured based on a set of queries. Therefore, MAP is a metric for calculating the mean of average precision per query, and it is calculated as follows:

$$MAP = \frac{\sum_{q=1}^N AveP(q)}{(|\{Q\}|)} \quad (2.15)$$

2.4.5 Discounted cumulative gain (DCG)

Another measure for calculating the effectiveness of retrieval is Discounted cumulative gain (DCG). DCG is also used for retrieval systems that produce ranked list results, however, its main advantage is that DCG can handle graded relevance judgment. In other measures based on ranked list results, only the position of relevant documents are important, but DCG consider the relevancy degree of relevant document in to account. Therefore, the main difference of DCG is calculation of gain or usefulness of each relevant document based on relevancy degree. DCG is calculated as follows:

$$DCG_p = \sum_{i=1}^p \frac{2_i^{rel} - 1}{\log_2^{i+1}} \quad (2.16)$$

2.4.6 Normalized DCG

When the length of result list for each query is different from other queries, DCG is not enough for measuring the performance of a retrieval system and there is a need for a normalized measure. NDCG is a normalized DCG. NDCG measure the DCG based on the position of results in result list. NCDG is calculated as follows:

$$nDCG_p = \frac{DCG_p}{IDCG_p} \quad (2.17)$$

IDCG means the Ideal DCG, which means DCG of the ideal ranking of the result list is calculated. For example if we get the result as it shown it Table 2.3:

According to Table 2.3, DCG of this result list is:

$$DCG_5 = 7 + 1.88 + 3.5 + 0.43 + 1.17 = 13.98$$

In the ideal order the relevancy degree must be: 3,3,2,2,1

$$IDCG_5 = 7 + 4.40 + 1.5 + 1.29 + 0.39 = 14.58$$

$$NDCG_5 = \frac{13.98}{14.58} = 0.96$$

Doc_i	relevancy degree	$2_i^{rel} - 1$	$\log_2^{i+1}$
Doc_1	3	7	1
Doc_2	2	3	1.59
Doc_3	3	7	2
Doc_4	1	1	2.32
Doc_5	2	3	2.56

Table 2.3: DCG example

2.5 Patent retrieval systems

2.5.1 Introduction

In the previous section, we have explained the general information retrieval systems. Here, we explain patent retrieval systems as our case study in this thesis. A patent retrieval system is different from other retrieval systems. To show the differences, in this section we introduce a patent, and then explain the specific characteristics of patent retrieval systems. At the end of this section, patent prior art candidate search is defined. The main purpose of this study is to improve recall of patent prior art candidate search.

2.5.2 Patent

"A patent is an exclusive right granted for an invention, which is a product or a process that provides a new way of doing something, or offers a new technical solution to a problem. A patent provides protection for the invention to the owner of the patent for a limited period, generally 20 years." [17]

This sentence explains what a patent document is. Nowadays, patents are the main part of industries innovative products. 85-90% of a new technology cannot be found anywhere else [18]. Industries have to search for patents in their field to start a new product or to avoid duplication with current patents. Patent have lots of usage such as: 1) We can understand how to use something; patent document explains the way of using patents. Moreover, we have private patents and firm patents. By finding firm patents, 2) We can find information about the activities of the firm. 3) For researchers and start-up companies finding patent can be helpful in order to identify a new research front and opportunities. 4) A patent is a good way for protecting an idea by the rights that it can give to its owner [18].

Generally, patents are categorized to three types [18]: first type is "Utility patents" which are 90.5% of all registered patents which are expalining new technology; second type which is 9% of patents are "Design patents" that focus on functional design to improve the usability, such as coca cola bottle. Last category is "Plant patents" which are 0.5% of all patents, which are about a new idea for particular things or fields, such as coca cola formula. Not every document can be a patent; the following criteria are basic conditions for a patent to be registered [18]:

1. **Novel (Unique):** The patent must be unique and new and never be published before the date of the filled application
2. **Non-obvious:** Patent must be not obvious to the skilled of the filed

3. Useful

4. **Full disclosure:** Patent must provide the full disclosure of the technology

Patent documents have their own structure. Each patent contains the following parts:

1. A first page comprising basic "bibliographic" information, such as the title of the invention and the name of the inventor, as well as the application number and date, etc.
2. The bibliographic information also contains a category number which defines the domain of the patent. The most well known patent classification is International Patent Classification (IPC).
3. An Abstract which is a brief description of the invention.
4. Detailed description of the invention, indicating how it is constructed, how it is used and what benefits it brings compared with what already exists.
5. One or more claims, defining the scope of protection sought.
6. One or more drawings for explaining the invention.
7. Sometime patent documents also contain a report which list the documents found by a patent office during the granting procedure.

2.5.3 Prior art candidate search (PAC)



Figure 2.2: Process of PAC

Patent organizations or patent offices are the places around the world that can examine the applications, register and grant a patent. There is a list of all the patent offices around the world on the Internet in [19]. Each of these offices around the world has their own approaches for a patent registration. There is no worldwide patent, and patents must be registered independently in the countries which the owner chooses to protect his/her idea. The rule and regulations of patent registration can also be different from one country to another. Patent prior art candidate search (PAC) is patent examiners job in patent offices which is done for registering a new patent. Figure 2.2, shows the simplified version of granting a patent by patent examiners. PAC finds past patents for avoiding conflict between claims, and it is critical for granting a new patent[20].

In the other words, PAC is a way to retrieve published or registered patents, which contain conflicting part for invalidating the patent. In this study, the goal of our system is to improve PAC and reduce the number of exhausting search that patent examiners do.

2.5.4 Characteristic of patent retrieval systems

To analyze a system, we need to understand its characteristics. In this part, we discuss the main characteristics of a patent retrieval system (PRS). These characteristics make patent retrieval systems different from other information retrieval systems.

1. **Recall oriented:**

Patent retrieval systems are recall-oriented in contrast with most of the retrieval system which are precision-oriented [21]. To do PAC, there is a need to retrieve all relevant patent documents. Therefore, in this system, recall is more important than precision.

2. **Structured based documents:**

Patent documents have their own structure; they have a title, an abstract, a claim, a detailed description as well as bibliographic information [22] [23]. This structure helps owners to protect their rights. Since claims define the scope of the patent and used to protect the rights, this structure is also help examiners to grant a new patent. Therefore, patent retrieval systems have to consider them for retrieving documents. Moreover, patent documents have classification, which shows the domain of the patent document. Although, finding the right cluster for patent (classifying patents) is an issue, retrieval system can use this classification code in their systems.

3. **Lengthy documents:**

Patents are long structural documents. In average patents are 24 times longer than NEWS and 20 times longer than articles[24].

4. **Full of technological information:**

Today, 90% of technology is from patents. Patents have lots of technical words and are domain- specific[25]. Patents have lots of knowledge about design technologies and methods. It is reported that by effectively using patents 60% of R&D time and 40% of R&D funding is saved[26].

In this thesis, we use these characteristics in our system. Therefore, our aim in this study is improving the recall-rate. We also, consider different parts of patent documents for retrieval. We use different parts to show the relation of the length of query/document on retrieval in addition to study each part as a query for PAC. Furthermore, we use knowledge sources to deal with technological information of patent documents and help inexperienced users to retrieve patent documents which they don't know their vocabulary to use in their query string.

Chapter 3

Vector-based Semantic Expansion (VSE)

There are many existing techniques for retrieving documents. One of these techniques which helps to improve the recall is *expansion*. In this chapter, we first introduce the expansion concept and techniques. Then, we present how it can improve a retrieval system. We also review the related work. Then we propose our novel expansion approach, which we call it Vector-based Semantic Expansion (VSE), and we describe how VSE utilizes WordNet and Wikipedia. Finally, we present our expansion approach for Word (VSWE), Query (VSQE) and Document (VSDE).

3.1 Introduction

Expansion literally means an increase in extent, size, volume, scope, etc. In computer science, mostly refers to expand one word to several words, which can be based on different relations between words such as synonymy, IS-A, Part-Of, etc. Expansion is used for different purposes such as terminology expansion, personalized recommender systems and information retrieval. In this study we use it in information retrieval.

One of the problems in information retrieval is word-mismatch. In these kind of systems, users only get results which have the exact terms that they choose for formulating the query string. Based on the research in [27], 25% of results which are relevant to the query do not have the exact words which are come in the query string. Mostly, one concept can be described with more than one word (synonymy), however, users use one of these words to submit a query string [27]. In this way, expansion can be helpful for non-experienced users to retrieve more relevant documents based on their needs. Moreover, in retrieval systems such as patent retrieval which are very expert dependent, expansion could let normal users extract useful information from the system without having prior knowledge about the field.

By expansion, system can help users to retrieve results faster and easier. In information retrieval, expansion is used for queries which is called Query Expansion (QE) to solve the word-mismatch problem, or it is used for documents which is called Document Expansion (DE) to solve the errors and noises inside the documents and improve the quality of collection, mostly for speech recognition [28]. Both of these techniques used to improve the recall and help users to find more relevant documents. We discuss

DE and QE and their differences in more details later in Section 3.3.

3.2 Expansion Concept

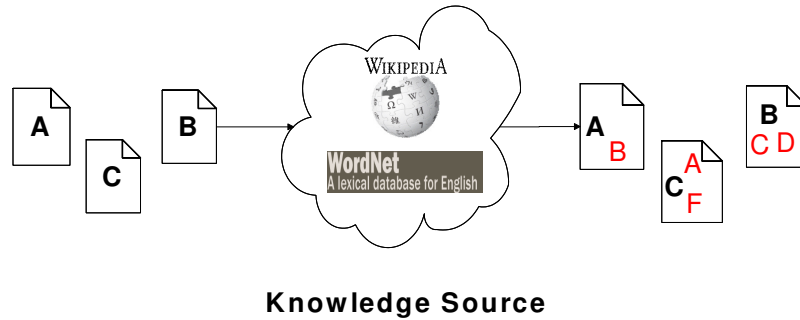


Figure 3.1: Semantic expansion schema

Expansion can be done in two different ways. First, expansion can be done by using the collection and its statistical information, **statistical expansion**. For example, by analyzing the collection we find that in more than 80% of documents, the word `tree` is used with the word `leaf`. Therefore we can expand the word `tree` by using this statistical information to `tree` and `leaf`. One of the advantages of using statistical information is the performance, however, it is domain dependent and we cannot use the information of one collection in other collections. For example, if our collection is about the algorithms, word `tree` can be expanded to `binary` and `tree`, or, `node` and `tree`. We explain this category in more detail in Section 3.3. Second way of expanding is based on the meaning of the word, **semantic expansion**. Semantic Expansion methods aim to reach the goals of expansion by using external knowledge sources. The important advantage of using external sources is an ability to use different relations. Words have different relations with each other and using these relations can improve the quality of expansion. For example by using dictionary we can find the synonymy relations. Moreover, by using semantic expansion methods the results of expansion can be used for other collections, and increase the size of collection do not influence the expansion, therefore, expansion for each word is not dependent to the characteristics of a collection. For instance, by using dictionary and synonymy relations, we can expand the word, `sick` to `ill`. We explain these two groups in more details in Section 3.3.

Furthermore, using external knowledge sources such as Wikipedia can help users to find new technical documents based on their needs, although users do not have any prior knowledge about them.

3.3 Query Expansion and Document Expansion

For retrieval purposes there are two main techniques which use expansion: Document Expansion (DE) and Query Expansion (QE). In this chapter, these two techniques are introduced, their differences are presented and the related work is reviewed.

3.3.1 Document Expansion vs Query Expansion

In information retrieval, most of the time, expansion refer to query expansion. Query expansion is a process of expanding the input of system to find more relevant documents. The main goal of query expansion is finding additional documents by adding more words to the query string. Another retrieval expansion technique is Document expansion, which is expanding all documents inside the collection. The main goal of query expansion and document expansion is improving recall, however these two have differences.

1. **Length:** In query expansion, expansion is done for one query string. Length of the query string most of the time is less than the length of one document. However, in document expansion we have to expand all documents inside the collection.
2. **Phase:** Query expansion is done at processing time, however, document expansion is done at pre-processing time.
3. **Index:** Query expansion do not need a new indexing system, however, document expansion need a new indexing system to store the expanded tokens (documents).
4. **Time:** Since the length of document in document expansion increase, the time of processing document in query expansion is less than document expansion.
5. **Base:** In query expansion, increasing the words in the query is for finding new documents, however, in document expansion, increasing the words in document is for finding new relation between documents or improving the quality of collection.

3.3.2 Advantages of using DE and QE

As an example, assume that we have three documents. As you can see in Figure 3.1, in each document we have one word: A, B and C. By expanding the documents via external knowledge sources, assume A expand to (A, B) , C to (C, A , F) and B to (B, C, D). Therefore in such a system, if the user submit a query string B, by using normal retrieval, s/he can only retrieve document B, while by using de with qe, the user receives all three documents, and by using only qe, document B and C comes in the result list. Most of the time, in normal retrieval systems, it is impossible to be sure that we get all the information from the system.

Relations between words are important factors in retrieving documents. Most of the time when we look for the word *Sick*, we are pleased to get documents about *ill* and *diseased*, etc. in a rank listed result.

In patent retrievals, inventors use different meanings of one word to escape from normal retrieving procedure. For example they may use *screen* to describe *monitor*. Therefore, expansion can be a very good solution for patent retrieval systems which are recall-oriented as we mentioned in Chapter 2 and they are very sensitive to quality of retrieval. In Patent retrieval systems the quality depends on the experience of the user, therefore it is necessary to have reliable system which is not qualified based on the user experience.

3.3.3 Related Work

As we discussed in Section 3.2, expansion is done in two ways, statistic-based and semantic-based. This category is valid for query expansion, however, query expansion can be done also manually or automatically. Automatic query expansion has a significant advantage over manual query expansion: it save the effort of user to improve the system. Our system is also automatic query expansion. Moreover, since we do not have an access to patent examiners or experts to get feedback, our system works automatically

Semi-automated or manual expansion category of techniques which are based on the user feedbacks, is *relevance feedback*. In this category, number of relevance terms are extracted from the document collection and returned to user as an expansion suggestion then based on the users feedback the expansion can be improved.

Document expansion mostly used for speech recognition [28]. The main purpose of using document expansion for speech recognition is avoiding vocabulary mismatch and errors, and in text-based image retrieval to expand the short annotated image documents and find more relevance images [29]. It is also can be used for text retrieval, (Sing-hal and Pereira, 1999) [28] in the context of the vector space retrieval model. For document expansion, there are also lots of works based on clustering techniques, such as (Kurland and Lee, 2004; Liu and Croft, 2004) [30] [31]. In particular, the cluster-based language model proposed in (Liu and Croft, 2004) [31] uses clustering information for document expansion. There are also several techniques for DE which use pseudo-relevance feedback (PRF), or as it known in IR nearest neighbor.

Based on the literatures, statistic category is mostly called **probabilistic query expansion**. Probabilistic query expansion is also categorized to local and global query expansion. Global techniques extract expanded term from the whole collection, however, in local techniques only the top n documents which are extracted based on the original query is used as a collection for extracting terms. The important fact about the local technique is that all the calculations for these techniques have to be done on-line, which produce delay to response to user. However, global techniques can calculate expansion of terms off line without knowing about the query. One of the important global techniques is clustering techniques, such as term clustering (Jones, 1971) [32]. In this technique, expansion is based on the co-occurrence of terms in the document doclection. The other well-known global technique is Latent Semantic Indexing (Deerwester et al., 1990) [33] and (Jing and Croft, 1994) [34] which is based on the relation of term and documents in the matrix.

For local techniques,(Attar and Fraenkel, 1977) [35] as a first local analyzer, use term clustering but for the first top n documents in the result list. The local techniques hypothesis is that top n documents are more relevant to the user query. Probabilistic techniques are popular because they are effective and all the relations are generated from the collection. However, Semantic based expansions, which use the external sources and ontologies for expansion, use relations between concepts which are edited manually and stored in an ontology.

In principle and as we show in the example in Figure 3.1, by expanding three document A, B and C, document expansion can retrieve more results than query expansion, however, document expansion is more time and cost consuming. Generally, the quality

of semantic document/query expansion is related to the knowledge sources. Here there is a list of factors which influence on quality of semantic expansions:

- **Domain:** Semantic expansion is domain-dependent. The result of expansion can varied from one source to the other; therefore, choosing the suitable source is very important. For example choosing Mesh over Wikipedia for medical domain.
- **Noise:** The quality of source influence the quality of expansion. If expansion leads to decreasing the semantics, expansion produces noise. For example, there are lots of discussions of the trustworthy of Wikipedia pages as one of the largest knowledge sources. We discussed it in Chapter [ch:similarity], expansion via Wikipedia can produce noise in expansion. Moreover, even the sources like WordNet which are constructed by professionals can produce noise in expansion. If we use Hypernymy links in WordNet, beside the un-uniform weights of the links, by expanding all tokens to the root of the WordNet graph, we generalize the meaning of the words inside the document and decrease the semantics.
- **Uniform expansion:** number of expanded tokens has a big influence on measurement. For example, if by expanding two words, one of them expanded to 3 more tokens and the other one expanded to 30 more tokens, the weight of their vectors become so different. Therefore, it is very important to expand documents uniformly. By limiting the length of expansion, we have to choose between the expanded tokens. For example, for Wikipedia, we use links which have more ingoing links; therefore we choose number of links that have more ingoing links. In this way, we can also reduce the noise, however, by choosing the most important pages in Wikipedia, we lose more semantics and in some how we generalize our documents.
- **Dimension of document or query:** Number of tokens inside one document is showing the dimension of expansion. Increase the length of the documents can influence the quality of expansion, we analyzed it for document/query expansion in Chapter 6.

3.4 Vector-based Semantic Expansion Approach (VSE)

In this section, first we describe the main idea of Vector-based Semantic Expansion (VSE), how it works and which knowledge source is suitable to extract more relevant information to increase the recall in PAC. In next sections, by introducing WordNet and Wikipedia, we describe the algorithms of Vector-based semantic Word expansion (VSWE) based on WordNet, Wikipedia and their combinations. In next section, we describe VSE for query and document expansion namely VSQE and VSDE.

Our approach for expansion is semantic expansion. For expanding with external sources, there are lots of options. Depending on the domain of the collection, we can use different sources, for example, for medical domain we can use Mesh [36] or UMLS [37]. To improve the way that patent examiners do their job, first we have to take a look at their own process of doing the job. In Figure of PAC process in Chapter 2, we presented a simplified version of process of granting a given patent. Patent examiners

search for important keywords from the given patent and based on the concept of a given patent they also search for the synonyms. After several try, they extract similar documents which have conflicts with the given patent and then they write a report about all the conflicting parts in order to grant the patent. Patent examiners have three main abilities which let them be an expert for doing this job:

1. Language understanding
2. Knowledge of the field
3. Experience

If we assume that by providing these abilities and enrich the retrieval system, computer can do the job, then we have to find a suitable knowledge sources that provide language understanding and technological knowledge. We provide WordNet [38] as an English dictionary for ability of understanding of language and Wikipedia [39] as a source of knowledge for understanding the field. Since patent documents have lots of new knowledge and technological terms, Wikipedia is the best option. Wikipedia by far is the largest encyclopedia in existence with over 1 million articles contributed by hundreds of thousands of volunteers with remarkable quality [40] [41]. Therefore for expansion, since our case study is patent retrieval systems, we choose Wordnet and Wikipedia. Moreover, based on the broad coverage of these two knowledge sources, they also can be used in other domains. In our approach we represent each token, sentence or text by a vector like vector space model (VSM) which we explained in Chapter 2. In this way we can use the advantages of using vectors in our model, such as performance. Moreover, by vector representation, we can easily combine expanded vectors. At the end of this chapter, we show how the expanded Wordnet and Wikipedia vectors are combined.

3.5 VSWE-WordNet

In this section, we introduce WordNet as one of the largest available knowledge sources on the web for free as one of our knowledge source. And then we explain how we use it for our method.

3.5.1 WordNet

WordNet [38] is a semantic taxonomy for English language which is developed at Princeton University. English nouns, verbs, adjectives and adverbs are organized into synonym sets, each representing one underlying lexical concept. The IS-A relations form IS-A taxonomies with nouns and verb synsets, and they are major relations in WordNet. The relations in taxonomies of nouns are hypernym/hyponym and holonym/meronym relations while in verb taxonomies are hypernym/troponym.

Number of nouns and verbs in Wordnet is listed in the following table:

3.5.2 VSWE-WordNet

To expand each word we have to choose words that share the same characteristic, best option in WordNet is IS-A relation. Therefore we use hypernymy and hyponymy re-

POS	Unique Strings	Synsets	Total Word-Sense Pairs
Noun	117798	82115	146312
Verb	11529	13767	25047
Adjective	21479	18156	30002
Adverb	4481	3621	5580
Totals	155287	117659	206941

Table 3.1: WordNet statistics

lations and all the noun word forms of one concept(synset) for an expansion. Since nouns are the most meaningful part of each sentence, We choose nouns over verbs, adverbs, ect. Figure 3.2 shows the fragment of hierarchy relations of the word *teacher* in WordNet.

For expansion, we can use hypernymy links, hyponymy links or both of them. In Chapter 4, we explain the effects of choosing them on expansion to see which one is more suitable for expanding one word. For example for the word *teacher* a **Hypernymy** vector is as follows:

[abstraction, adult, causal agent, concept, educator, entity, idea, person, physical entity, professional, teacher]

and **Hyponymy** vector :

[Bahai, Cyril, English teacher, French teacher, art teacher, baseball coach, basketball coach, batting coach, catechist, coach, conditioner, dancing-master, demonstrator, docent, football coach, games-master, geometry teacher, governess, hockey coach, instructress, line coach, math teacher, missionary, music teacher, piano teacher, pitching coach, preceptor, reading teacher, riding master, schoolmarm, schoolteacher, science teacher, section man, teacher, teaching fellow, tennis coach]

To expand each word in a vector, we give different weights to expanded tokens. We assume that expanded tokens do not have the exact meaning that the original word has, therefore we add the given word in the vector with a weight equals to 1, and then all the other expanded tokens have a weight in $[0, 1]$. In Wordnet, since we use hypernymy and hyponymy, in addition to the weight of the links we have to consider the depth of the words from the given word. For example for *teacher*, the words *educator* and *coach* are in depth 1 and words *professional* and *basketball coach* are in depth 2. We assign a higher weight to the tokens which are closer to the given word in the graph and lower weight to the further ones. We discuss the effects of them in next Chapter, Chapter 4.

For this purpose, the weight is calculated by given link's weight to the power of the token's depth (Distance from original token). Equation 3.1 shows how we calculate weights of the expanded tokens from the link weights.

$$Weight_{Token} = Weight_{Link}^{Depth} \quad (3.1)$$

For example if we assume that the hypernyms weigh is 0.5, then:

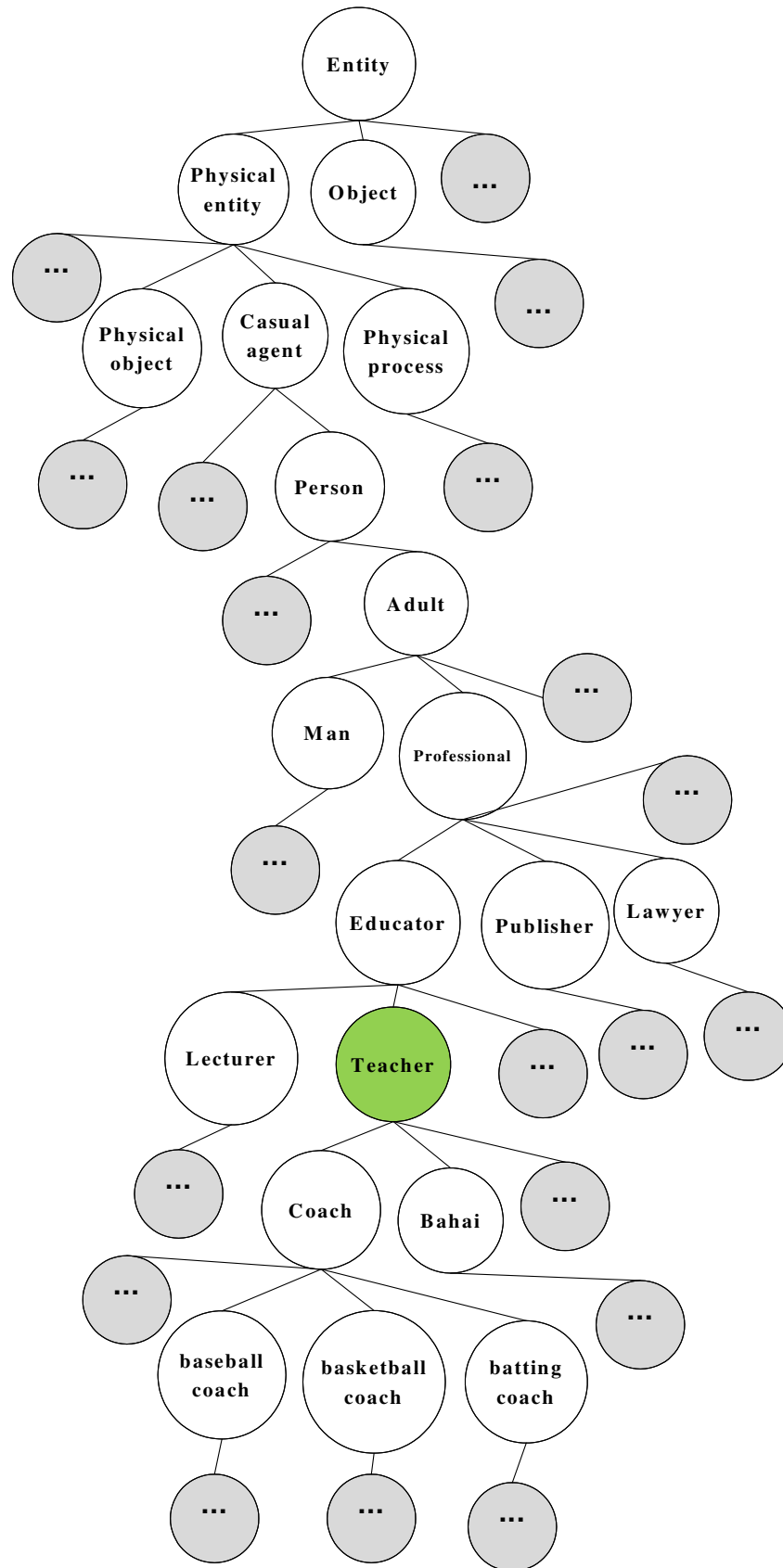


Figure 3.2: Fragment of hierarchy relation of teacher in WordNet

[(teacher, 1), (educator, 0.5), (professional, 0.25), (adult, 0.0625), (person, 0.0078125), (causal agent, 4.8828125E-4), (physical entity, 1.52587890625E-5), (entity, 2.384185791015625E-7), (abstraction, 1.862645149230957E-9), (concept, 7.275957614183426E-12), (idea, 1.4210854715202004E-14)]

Algorithm 1 explains how we construct a expanded vector for each token in WordNet.

Algorithm 1 Token WordNet Expansion Vector

```

TokenVector  $\leftarrow$  null
HypernymVector  $\leftarrow$  null
HyponymVector  $\leftarrow$  null
if hypernyms  $\equiv$  true then
  {If user chooses only hypernymy relations for expansion}
  HypernymVector  $\leftarrow$  Hypernym(Token, HypernymDepth, HypernymWeight)
end if
if hyponyms  $\equiv$  true then
  {If user chooses only hyponymy relations for expansion}
  HyponymVector  $\leftarrow$  Hyponym(Token, HyponymDepth, HyponymWeight)
end if
  {If user chooses both hypernymy and hyponymy relations for expansion}
TokenVector  $\leftarrow$  HyponymVector + HypernymVector

```

Algorithm 1 shows how we build a hypernym vector for each token. To construct a Hypernym vector for each token we need Hypernymy Depth and Hypernymy Weight. Hyponym vector is constructed in the same way as hypernymy, however, instead of hypernymy we add the hyponymy links. This means that we go down to the leaf nodes in the graph instead of going up to the root node. To have a better performance and find more words in WordNet graph, we use `stemming`.

When we want to calculate the Hypernymy or Hyponymy, if we don't find the word in WordNet graph, we check the stem of the word. The main reason that we don't use stemming before searching the wordNet graph is the differences between meanings of the words that have the same root. For example, words fish and fishing which have the same stem as fish, have different branches in WordNet. Therefore, we choose to use stemming only if we do not find the original word in WordNet graph to increase the number of hits to find words in WordNet graph. An other reason that we don't use stem of the words for searching Wordnet is based on the suggestion of Karen Spärck Jones for classifying term for term frequency in [42]. She suggested to use all terms for matching and not classifying them to increase the matching potential. As far as using stemming without searching the original word in WordNet is kind of classifying words to their stem, we use original words and then if we don't find them the original word in WordNet graph, we use their stem. In this way if users want to expand fish and fishing, they can get different expansion vectors.

Algorithm 2 Token WordNet Hypernyms Vector

```

index = 1
TokenVector ← null
depth ← 1
TokenVector.token[0] ← (word)
TokenVector.weight[0] ← 1
Stem ← Stemming(Word)
if Word ∉ WordNet then
    Word ← Stem
end if
startDepth = index
endDepth = 0
if Word ∈ WordNet then
    hypernyms ← WordNet.Hypernymy(word)
    while (hypernyms.length ≠ 0) and (depth ≤ HypernymDepth) do
        i = 0
        while i ≤ hypernyms.length do
            if endDepth ≡ 0 then
                endDepth = hypernyms.length
            end if
            TokenVector.token[index] ← (hypernyms[i])
            TokenVector.weight[index] ← (HypernymWeightdepth)
            index ← index + 1
            i ← i + 1
        end while
        if startDepth ≡ endDepth then
            depth ← depth + 1
            endDepth ← index - 1
        end if
        startDepth ← startDepth + 1
        hypernyms ← WordNet.Hypernymy(TokenVector.token[startDepth])
    end while
end if

```

3.6 VSWE- Wikipedia

In this section, first we introduce Wikipedia and its weakness and strength points, then we explain how we use it for our method.

3.6.1 Wikipedia

Wikipedia is a free web-based encyclopedia which is collaboratively written by volunteer around the world. Since we use wikipedia as a knowledge source for patent retrieval systems, in the Table 3.2, we summarized the strength and weakness of this approach. The nature of Wikipedia is very interesting. There are lots of discussions of the quality of measures and methods which are based on wikipedia [41]. For ex-

ample, according to literature, methods based on Wikipedia can be improved during the time by growing Wikipedia. As it is shown in Table 3.2, Wikipedia is written in multiple

Strength	Weakness
Broad coverage	Edited by non-professional people (volunteers around the world)
Free	Not always correct
High rate of updating	Links do not have the same meaning
Multi lingual	Links do not have the same weight (distributed un-uniformly)

Table 3.2: Strength and weakness of using Wikipedia

languages and since patent can be granted in more than one country, multilingual characteristic of Wikipedia pages can be very helpful. Even the weakness points that we summarized in Table 3.2 have some advantages, for example, we mentioned the meaning of the links. Besides the weights of the links and problem based on un-uniform shape of the Wikipedia network which is the same as WordNet and we explain it later in this section, Wikipedia links do not share the defined meanings. Wikipedia Links can have the IS-A or Part-Of relations in addition to relations which do not have the name. All the relations in WordNet are based on the lexical meanings however it is not true in Wikipedia. It can be a problem to categorize the links, however, these links show the relations between words which cannot be found in any other knowledge sources.

By un-uniform shape we mean the un-uniform distribution of nodes in hierarchy relations. For example, in WordNet graph which have the meaningful relations between nodes, like IS-A relation or Part-Of relations, links do not have the same meaning values. Especially, in specific fields, relations in WordNet as a dictionary are very general.

Furthermore, Wikipedia is edited by thousands of non-professional volunteers around the World which improve the quality of relations, and according to increasing the consumption of Wikipedia on Web, the quality of Wikipedia pages are reasonably high [41].

3.6.2 VSWE- Wikipedia

To use VSE with Wikipedia, we choose Link structure. We don't use the category here because as we discuss in previous section, our aim is to use the link structure to use all the relations that one word could have with other words. Moreover, Since in WordNet we used the Hyponymy and Hypernymy, and we want to combine Wikipedia and WordNet, therefore we avoid to use the same kind of information from Wikipedia. By using hyperlinks, for each word we find the page and then we expand it by page title of ingoing links and outgoing links. To choose more reliable page titles for expansion, we choose links which are more important than the others. We choose the number of ingoing links as an importance degree, because based on PageRank [43] algorithms, ingoing links on web are more important than outgoing links. To construct VSWE, we can choose to use ingoing links or outgoing links, or both of them. The effects of using them is discussed in Chapter 4. For example, for word *teacher*, we choose the importance degree = 1000 in going links and we just expand it by using 10 extra links, therefore

By using **Ingoing links** we expand it to:

[Bob Feller, Foster care, Framingham Massachusetts, Geraldine Ferraro, House arrest, Laura Bush, Leslie Ward, Linus Pauling, List of United States Senators from Tennessee, Wes Unseld, teacher]

And also by using **Outgoing link** we expand it to:

[Certified Public Accountant, Curriculum, Education in England, Education in Scotland, Education in the United States, Guru, Knowledge, Life insurance, Pedagogy, Samarkand, teacher]

Algorithm 3 Token Wikipedia Expansion Vector

```

TokenVector  $\leftarrow$  null
IngoingVector  $\leftarrow$  null
OutgoingVector  $\leftarrow$  null
if Ingoing  $\equiv$  true then
    {If user chooses only ingoing links for expansion}
    IngoingVector  $\leftarrow$  Ingoing(Token, IngoingLinkDepth,
        IngoingLinkWeight, Threshold, ExpansionSize)
end if
if Outgoing  $\equiv$  true then
    {If user chooses only outgoing links for expansion}
    OutgoingVector  $\leftarrow$  Outgoing(Token, OutgoingLinkDepth,
        OutgoingLinkWeight, Threshold, ExpansionSize)
end if
    {If user chooses both ingoing and outgoing links for expansion}
    TokenVector  $\leftarrow$  IngoingVector + OutgoingVector

```

Algorithm 3 shows how we build an ingoing vector for each token. To construct an ingoing vector for each token we need IngoingLinksDepth, IngoingLinksWeight, Threshold and ExpansionSize (number of links). Weights and depths are same as WordNet. Link weights are calculated based on their distance from the original page, like how we calculate for WordNet weights in Section 3.5.2 and also depth shows the distance. Outgoing vector is constructed in the same way as an ingoing vector; however instead of ingoingLinksDepth and IngoingLinksWeights, we use OutgoingLinksWeights and OutgoingLinksDepth. We also use stemming in the same way that we use stemming for WordNet. Threshold and expansion size are added as additional factors because most of the pages in Wikipedia have lots of links and it is impossible to use all these links for expanding one token. Therefore, by using these two factors, we control the size of expansion. Using all the links can produce lots of noises, too. We discuss it in Chapter 6 Moreover, in expansion, it is very important to expand uniformly. Uniform expansion means to expand equally and control the size of the vectors. In addition to threshold which means the minimum number of ingoing links to each page (Importance degree), we have an expansion size, which shows the number of expanded tokens per each token.

3.7 VSWE-WordNet and Wikipedia

One of the advantages of using VSE-vector is the ability of combining vectors from different knowledge sources. Combining of VSE-vectors is very easy. For example to combine the WordNet vector and Wikipedia Vector, we just do the expansion for the word by using both knowledge sources and then extend the vector by simply add two vectors together. By using more than one knowledge source, the extension can become more realistic. By extending the word in WordNet we find the meaning of the word, and in Wikipedia we find the relations between the given word and other words which do not have the same meaning or category with the original word. Expanding with Wikipedia leads to find the relations which cannot be found from the dictionary. For example, the relation between stars and the flag of USA. We show the results of combining two vectors in Chapter 4 and in Chapter 6.

3.8 VSDE and VSQE

For both query expansion and document expansion, we need to expand the whole query or whole document. The length of them varied, from one word to more than 3000 words. To expand the query or documents, first we need to do the expansion for all the tokens. In Chapter 2 we explain how retrieval systems use tokenizing and in Chapter 5, we explain how we tokenized documents for our system. After tokenizing we do the expansion for each token, as we explain in here, we can expand each token by using VSWE, therefore for each token we have one token vector. Following Algorithm 6 shows how we construct a document/query vector from its token vectors.

Algorithm 4 Token Wikipedia IngoingLinks Vector

```

TokenVector  $\leftarrow$  null
depth  $\leftarrow$  1
index  $\leftarrow$  1
TokenVector.token[0]  $\leftarrow$  (word)
TokenVector.weight[0]  $\leftarrow$  1
Stem  $\leftarrow$  Stemming(Word)
if Word  $\notin$  Wikipedia then
    Word  $\leftarrow$  Stem
end if
startDepth = index
endDepth = 0
if Word  $\in$  Wikipedia then
    ingoings  $\leftarrow$  Wikipedia.IngoingLinks(word)
    ingoings  $\leftarrow$  SORT(ingoings)
    {Sort is based on number of ingoing links}
    while (ingoings.length  $\neq$  0) and (depth  $\leq$  IngoingLinksDepth) do
        i = 0
        while i  $\leq$  ingoings.length do
            if endDepth  $\equiv$  0 then
                endDepth = hypernoms.length
            end if
            if (ingoings.NumberOfIngoigLinks  $\leq$  Threshold) and (index  $\leq$  ExpansionSize) then
                TokenVector.token[index]  $\leftarrow$  (ingoings[i])
                TokenVector.weight[index]  $\leftarrow$  (IngoingLinksWeightdepth)
                index  $\leftarrow$  index + 1
            end if
            i  $\leftarrow$  i + 1
        end while
        if startDepth  $\equiv$  endDepth then
            depth  $\leftarrow$  depth + 1
            endDepth  $\leftarrow$  index - 1
        end if
        startDepth  $\leftarrow$  startDepth + 1
        ingoings  $\leftarrow$  Wikipedia.IngoingLinks(TokenVector.token[startDepth])
        ingoings  $\leftarrow$  SORT(ingolings)
        {Sort is based on number of ingoing links}
    end while
end if

```

Algorithm 5 Token WordNet and Wikipedia Expansion Vector

```

TokenVector  $\leftarrow$  null
WordNetVector  $\leftarrow$  null
WikipediaVector  $\leftarrow$  null
if WordNet  $\equiv$  true then
    { If user chooses only WordNet for expansion }
    WordNetVector  $\leftarrow$  WordNet(Token, HypernymDepth,
        HypernymWeight, HyponymDepth, HyponymWeight)
end if
if Wikipedia  $\equiv$  true then
    { If user chooses only Wikipedia for expansion }
    WikipediaVector  $\leftarrow$  Wikipedia(Token, IngoingLinkDepth,
        IngoingLinkWeight, OutgoingLinkDepth, OutgoingLinkWeight,
        Threshold, ExpansionSize)
end if
    { If user chooses both WordNet and Wikipedia for expansion }
TokenVector  $\leftarrow$  WordNetVector + WikipediaVector

```

Algorithm 6 vector-based Query/Document Expansion

```

DocumentVector  $\leftarrow$  null
Index  $\leftarrow$  0
for  $i = 0 \rightarrow$  Tokens.length do
    for  $j = 0 \rightarrow$  TokenVector.length do
        DocumentVector.token[index]  $\leftarrow$  TokenVector.token[j]
        DocumentVector.weight[index]  $\leftarrow$  TF - IDF(Tokens[i])  $\times$ 
            TokenVector.weight[j]
        Index  $\leftarrow$  Index + 1
    end for
end for

```

Chapter 4

Empirical evaluation of VSWE

To retrieve relevant documents, information retrieval systems use similarity measures. There are several techniques and approaches to find similarity between two entities which can be terms, words, or texts. In this chapter, we first introduce the meaning of similarity and briefly discuss existing similarity measures and their categories for retrieval purposes. Then, we propose a similarity measure with VSWE to find similarity between two words. We compare VSWE similarity approach with other similarity measures based on word-word similarity data-set.

4.1 Introduction

To explain the similarity measure, first we have to understand the meaning of similarity. Similarity is a point, feature, or detail in which two items are alike, based on dictionary. Closeness is a generic notion that can be expressed in terms of proximity, similarity or distance. However, distance is a metric and similarity is a measure. Distance can be converted to similarity but not vice versa (because of triangle inequality). In IR, textual semantic distance shows how close two units of language are [44]. The unit of language can be words, terms, concepts, phrases, sentences, paragraphs, or documents.

In general, humans consider two units to be similar to each other if they share similar lexical meanings. However, human judge is not always constant and two people can agree or disagree on one distance estimate. Similarities are subjective and generating results, that satisfy humans, is a difficult task [45]. In order to evaluate the similarity measure, it has to have following properties [46] which known as similarity problems:

1. **Isomorphism:** (i) a is similar to a, (ii) if a is similar to b then b is similar to a, (iii) if a is similar to b and b is similar to c then a is similar to c.
2. **Insertion:** The similarity between a and b does not change by new insertions. This means the similarity value between two units must not influenced by adding c to the collection.
3. **Relative similarity:** Similarity must focus on one specific domain similarity such as corpus based similarity [47] or domain-specific [48].

These properties can be changed based on two factors [49]:

1. **Context effects:** similarity between two units cannot be measured without considering third unit as a context. a and b are similar based on context c, and if the context c is changed, the similarity of a and b may be also changed.
2. **Perceiver effects:** experience and vocabulary of perceivers can change their similarity estimations.

Finding similarity is a time and cost consuming [50] task. It is context dependent and dynamic [51], therefore it is not an easy task. Many researchers have proposed different methods to find the relatedness between two units. They use different sources to reach the human estimates, however, for similarity measures human estimates is also not constant. In the next section, Section 4.2 we discuss these similarity measures.

4.2 Related works

Based on the available similarity measure techniques that we have found in the literature, we classify them as follows:

1. **Statistic-based approach:** measure similarity between two units based on mathematical approach [52]. Most of the time units (especially texts) are represented as bags of words (words without order) and they are compared with each other based on associated frequencies of occurrence.
2. **Semantic-based approach:** measuring similarity between two units based on using external knowledge sources, such as using the graph of WordNet or Wikipedia.

4.2.1 Statistic-based approach

In this category of measurements, we explain the methods which use the statistical information of the corpus, without using any other resources. Most of the similarity measures in this category use vector to represent each document. The most important advantages of the measures in this category is their performance. Most of the measures in this category can retrieve documents very fast. Moreover, the quality of results in this category is reasonably high; therefore nowadays most of the retrieval systems use these measures for measuring the similarity between collections and query. In Chapter 2 we briefly introduced three most common measures of this category namely Cosine similarity, Dice Coefficient, extended Jaccard. All of them are vector-based similarity measures. There are also methods in this category which are not based on vectors. For example, LSA or LSI [53], is a measurement based on matrix. In this measure, the similarity between terms and documents are calculated based on global and local weightings [33].

4.2.2 Semantic-based approach

Finding semantic similarity by using ontologies is becoming more popular. Ontologies help to find inherent relations between entities and find similar documents based on their contents and not query them based on bag of words (BOW) approaches [54]. Measuring similarity based on ontology is not just about hierarchical relations, but

also the level of concepts in a hierarchy is used for finding similarity distances [55]. Lots of work [56] has been done for measuring similarities by using ontologies such as WordNet and Mesh, or even use some part of one ontology and construct a new ontology for measurement [57]. Most of the measures in this category use taxonomy for finding the relatedness between two language units [58]. To understand the meaning of the words or their semantics, use of external sources is unavoidable. One of the main problems of using these measures based on ontology is lack of attention to un-uniform shape of their graphs as we explained in Chapter 3. In most of the measures in this category, similarity is measured based on the position of nodes in the graph, however, the links are not distributed uniformly, we explain it in Chapter 3. For example, edge-based measures, like Rada (1989) [59] measure is based on the shortest path (length of path between two nodes) or Wu (1994) [60] and Li (2003) [61] measures are based on depths in addition to length of paths between two nodes. There are also several measures which add additional information to the measure to reach better results, information-content measures such as Resnik (1999) [62] introduce the idea of using information content for measuring the similarity. He suggests using the amount of information that two nodes share which each other as an important factor for measurement. Lin (1998) [63] and Jiang (1997) [64] are also use the concept of information content in their measurement. They add the information content of nodes in addition to the share information content that Resnik introduced.

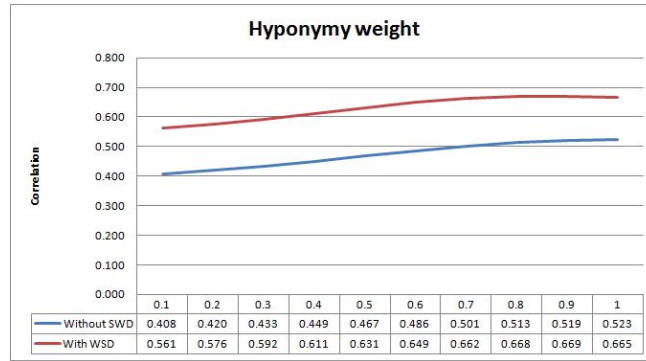
4.3 Word-Word similarity

Word1-Word2	Human Judgements	Rada	Leacock-Chodorow	Jiang-Conrath	Wu-Palmer	Resnik	VSWE-WordNet+Lesk	VSWE-Wikipedia	VSWE-WordNet+Lesk+Wikipedia
car-automobile	3.92	1	3.6889	12876699.5	1	7.0036	0.981584992	0	0.745791427
gem-jewel	3.84	1	3.6889	12876699.5	1	9.8198	0.509785925	0.589536693	0.457282938
journey-voyage	3.84	0.5	2.9957	0.3534	0.9565	7.4641	0.411746237	0.381754002	0.355351222
boy-lad	3.76	0.5	2.9957	0.2929	0.9333	6.7419	0.079403463	0.262259262	0.164997643
coast-shore	3.7	0.5	2.9957	1.6154	0.9231	8.1022	0.508511883	0.571560647	0.501517302
asylum-madhouse	3.61	0.5	2.9957	2.4663	0.9565	10.6671	0.473973174	0.636801929	0.458216258
magician-wizard	3.5	1	3.6889	12876699.5	1	11.0726	0.660338413	0.25092399	0.351507366
midday-noon	3.42	1	3.6889	12876699.5	1	9.5685	0.791368255	0	0.280410955
furnace-stove	3.11	0.1	1.3863	0.0597	0.5714	2.4934	0	0.360909055	0.209464239
food-fruit	3.08	0.1	1.3863	0.0861	0.4706	1.7798	0	0.783256437	0.185970442
bird-cock	3.05	0.5	2.9957	0.2681	0.9565	6.9374	0.112267573	0	0.083922968
bird-crane	2.97	0.25	2.3026	0	0.88	6.9374	0.234737443	0	0.150335361
tool-implement	2.95	0.5	2.9957	0.8484	0.9412	6.3104	0.447931497	0	0.372298469
brother-monk	2.82	0.5	2.9957	0.0689	0.9565	10.1563	0.806202401	0	0.51564585
crane-implement	1.68	0.2	2.0794	0.0784	0.7778	3.4451	0.071931914	0	0.050205085
lad-brother	1.66	0.2	2.0794	0.083	0.7143	1.9033	0	0	0.046594179
journey-car	1.16	0.0556	0.7985	0.0707	0.1905	0	0	0	0
monk-oracle	1.1	0.125	1.6094	0.0588	0.5882	1.9033	0	0	0
cemetery-woodland	0.95	0.1111	1.4917	0.0539	0.5	1.1692	0	0	0
food-rooster	0.89	0.0625	0.9163	0.0671	0.2857	0.6144	0	0	0
coast-hill	0.87	0.2	2.0794	0.2187	0.7143	6.1381	0	0	0.026338329
forest-graveyard	0.84	0.1111	1.4917	0.0587	0.5	1.1692	0	0	0
shore-woodland	0.63	0.2	2.0794	0.0594	0.6667	1.1692	0	0	0.03079461
monk-slave	0.55	0.2	2.0794	0.0661	0.7143	1.9033	0.089435643	0	0.069845466
coast-forest	0.42	0.1667	1.8971	0.0628	0.6154	1.1692	0	0	0
lad-wizard	0.42	0.2	2.0794	0.0759	0.7143	1.9033	0.291810751	0	0.226702942
chord-smile	0.13	0.0909	1.291	0.0597	0.375	0.7794	0	0	0
glass-magician	0.11	0.125	1.6094	0.0604	0.5333	1.8747	0	0	0
noon-string	0.08	0.0833	1.204	0.0653	0.3529	0.7794	0	0	0
rooster-voyage	0.08	0.0417	0.5108	0.0501	0.1481	0	0	0	0

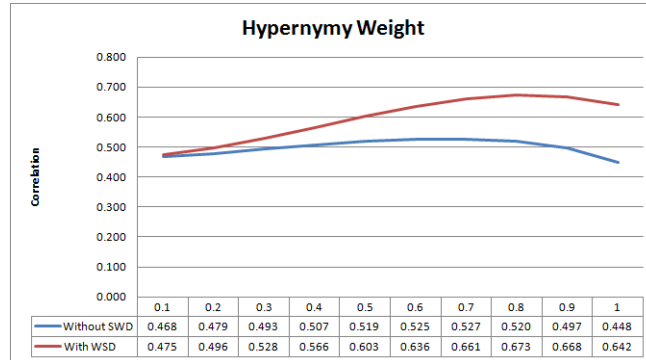
Table 4.1: Word-word similarity results

The main purpose of semantic-based similarity measures is using knowledge sources to measure similarity based on the content and not the frequency of the words. Our proposed method for expansion have the same characteristic, in addition to using statistical information, therefore we can use our expanded vectors with statistic-based measures to measure similarity between two unit of language. Most of the similarity measures in semantic-based approach use WordNet for measuring the similarity between two units of language. In this section, we compare our expansion approach for

measuring similarity with some of these semantic-based similarity approaches to measure the similarity between words. Since our VSWE approach construct a vector, for measuring the similarity between two vectors, we use statistic-based measures cosine similarity, dice coefficient and extended Jaccard. To have a fair comparison between our WordNet-based approach with other WordNet-based approaches we use WordNet 3.0 [65]. Our evaluation is based on Miller and Charles dataset [66] which is ground truth data for measuring the similarity between words. Miller and Charles asked an assessor to rate 30 pairs of words between 0 and 4 and their dataset is commonly used for research purposes.



(a) WordNet Link Weights



(b) WordNet Link Depths

Figure 4.1: Effects of WSD on results

First we evaluate our VSE method without using sense disambiguation. Word-Sense disambiguation (WSD) [67] is one of the language processing problem to identify the correct sense(meaning) of a word in the sentence. Each English word can have several meanings, and by using WSD the correct meaning is distinguished from the sentence. In the second round, we add sense disambiguation by using Lesk's algorithm [68]. Algorithm 7 explains how we use Lesk in our system. In our Lesk algorithm, we compare different definitions of two words to find the definitions that have more words in common.

As we explained in Chapter 3, to make VSWD-WordNet we have to calculate weight and depth for each expanded node, therefore our values for WordNet experiment in first and second round are: Hyponyms Depth = 3, Hypernyms Depth=4, Hyponyms

Weight=0.8, and Hypernyms Weight=0.9. We explain these values in Section 4.4

WSD-Lesk	VSWE-WordNet-Cosine correlation
Yes	0.686
No	0.512

Table 4.2: Effect of WSD on VSWE-Wordnet

According to Table 4.2, results shows that WSD have a big influence on results. We perform number of runs based on different values of weight and depth and even kind of links (Hyponymy or Hypernymy). (We explain about the values in Section 4.4). Here in Figure 4.3, we present our runs based on Hyponymy and Hyponymy links. We change the weights and from the figure, always the correlation with WSD is better than correlation without using WSD.

Algorithm 7 Lesk Algorithm for SE-WordNet

```

Synset[] synsets1 ← getSynsets(word1, SynsetType.NOUN);
Synset[] synsets2 ← getSynsets(word2, SynsetType.NOUN);
SynsetDefinitionWord1 ← synsets1.getdefinition(word1)
SynsetDefinitionWord2 ← synsets2.getdefinition(word2)
DefinitionIndex1 ← 0;
DefinitionIndex2 ← 0;
Subscription ← null;
MaxSubscriptionSize ← 0;
for i = 1 → SynsetDefinitionWord1.length do
  for j = 1 → SynsetDefinitionWord2.length do
    Token1 ← Tokenized(SynsetDefinitionWord1i);
    Token2 ← Tokenized(SynsetDefinitionWord2j);
    Subscription ← Token1 ∩ Token2;
    if Subscription.size() ≥ MaxSubscriptionSize then
      MaxSubscriptionSize ← Subscription.size();
      DefinitionIndex1 ← i;
      DefinitionIndex2 ← j;
    end if
  end for
end for
return DefinitionIndex1, DefinitionIndex2;

```

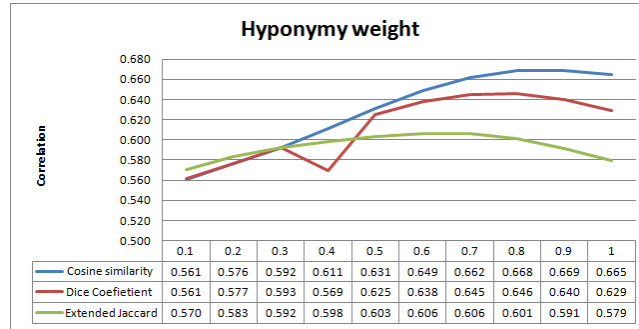
We also compare different similarity measure. In Table 4.3, the result of our experiment for best run is presented. Based on number of runs by changing the values of weight and depth and even kind of links (Hyponymy or Hypernymy), we find out that most of the time the cosine similarity is outperforms the dice coefficient and extended Jaccard measures. Here in Figure 4.3, we present our runs for on hyponymy and hyponymy links based on different weights to show to differences between these three measures. Therefore, we choose cosine similarity for measuring the similarity between VSE- vectors.

Then we evaluate word-word similarity with Wikipedia. Based on Chapter 3, we

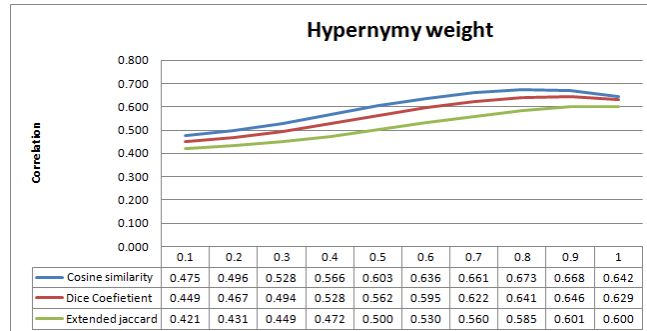
Method	Cosine	Extended Jaccard	Dice Coefficient
VSWE-WordNet correlation	0.686	0.608	0.654

Table 4.3: Effect of statistic-based measure on VSWE

have to calculate some values for VSWE based on Wikipedia, therefore our values for Wikipedia expansion in this experiment are: ingoing links Depth =1, ingoing links Weight = 0.6 outgoing links Weight = 0.9 Links =10, and Threshold = 500. We discuss these values in Section 4.4.



(a) WordNet Link Weights



(b) WordNet Link Depths

Figure 4.2: Effects of different similarity measures on results

The result of VSWE-Wikipedia is not very high according to table 4.3. Several reasons can lead to these results: i) By searching for a word in page titles, we only extract the exact match, however, there are several words which do not have any page in Wikipedia. For example by searching *car* in Wikipedia, we do not find any results, however when we search through web for car, automatically Wikipedia, link to automobile. In our approach, by searching the Wikipedia database we do not find any results and then we do not get any expansion, therefore the similarity between car and automobile is 0 according to Table 4.3. ii) Wikipedia can produce lots of noise in the vector which lead to less similarity. For example, by expanding the word *rooster*, in vector we have *France national rugby union team*. Although expanding to the words like *France national rugby union team* produce noise in finding the similarity with food, it also can show some relations which can be useful to find the similarity be-

tween rugby and rooster.

To get a better results, we construct a VSWE- vector based on both Wikipedia and WordNet by using Lesk. Table 4.3 shows the Pearson's correlation coefficient of different methods [69]. However, our VSE-approaches have less correlation than Resnik to find the word-word similarity, it have its own advantages. As we explained in previous chapter, Chapter 3, Our VSWE approach can use any kinds of external knowledge sources for expansion and it is not limited to WordNet, moreover, we can easily combined these knowledge sources to get a better results. As you can see in the Table 4.3, we use Wordnet with the result of 0.686, then we use Wikipedia with the result of 0.611, then the combination of these two knowledge sources have a better results which is 0.792.

Similarity measure	Correlation
Web Dice	0.267
Lesk	0.346
Web overlap	0.382
Sahami	0.579
PMI	0.534
Vector-pair	0.605
VSWE-Wikipedia	0.611
Jiang and Conrath	0.635
Hirst St. Omze	0.666
VSWE-WordNet+Lesk	0.686
Rada	0.755
Lin	0.769
Wu and Palmer	0.768
Leacock and Chodorow	0.779
VSWE-Wikipedia+WordNet+Lesk	0.792
Resnik	0.823

Table 4.4: Correlation results of Word-Word similarity based on different methods

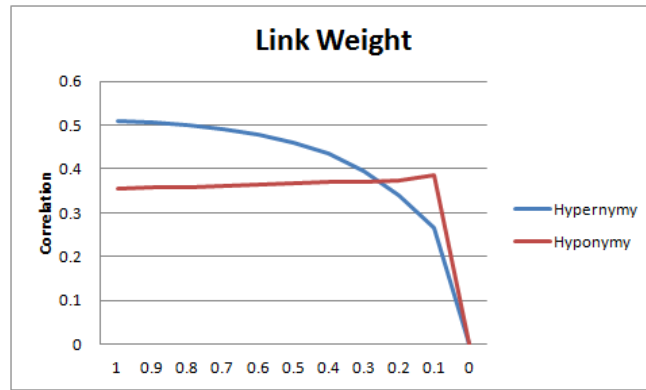
4.4 Choosing values for the VSWE approach

Different combination of values leads to different results. Therefore in this section we discuss the effects of different values on word-word similarity results. To get the results for each value and to measure the effects of it on similarity, we make other values constant and just change one value.

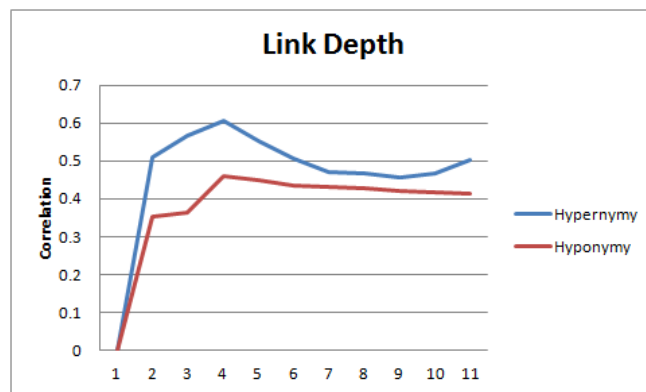
First we discuss the WordNet values and then Wikipedia Values. Finally, we discuss the combinations of these values.

4.4.1 WordNet

To construct VWSE via WordNet, we have to find the effects of two values, Weights and Depths. Figure 4.4.1 represents the effects of these two values.



(a) WordNet Link Weights



(b) WordNet Link Depths

Figure 4.3: Effects of WordNet values on results

According to Figure 4.4.1, consider both hyponymy links and hypernymy links for expansion have a positive influence on correlation, however, effects of hypernymy links are more than hyponymy links. Is-A relation in WordNet is like inheritance relation. Therefore Hypernymy links are parents and Hyponymy links are children. Based on inheritance, children have all the features of their parents. Therefore, since hyponymy relations are like children relations in inheritance, their effects are less than Hypernymy. Children in inheritance relation have more features than their parent and these extra features make them different from each other and also from their parents, and this different feature influence on similarity.

Depth diagram in Figure 4.4.1, shows the effects of distance in the graph. We examine hypernymy depth till depth equal to 45. The results shows that the best value for this collection is 4, more depth do not change the correlation too much and more depth in hypernymy links generalize the meaning of the words. We also examine the depth for hyponymy, and the results show that the best value is for depth equal to 3. More depth produces links which have lots of extra features compare to the original word. Therefore, more depth in hyponymy relation produces noise in our expansion.

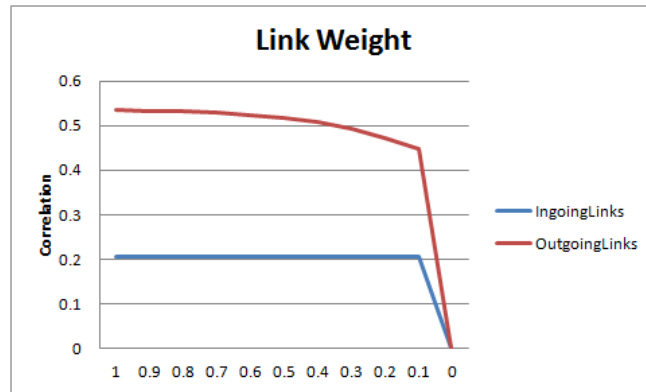
4.4.2 Wikipedia

For Wikipedia values, we examine the effects of four different values, weight and depth of ingoing and outgoing links, threshold or importance degree (number of ingoing links) and also size of expansion for each word.

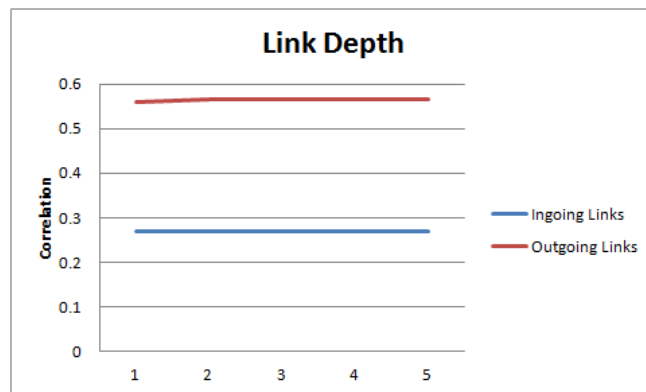
According to Figure 4.4.2, both outgoing and ingoing links influence on expansion and using both could links improve the correlation. All results in Figure 4.4.2 shows that expansion with outgoing links outperform the results of ingoing links. Outgoing links are mostly from the description of the wikipedia page, most of the times, these links are much related to the meaning of the page title than the links which are refer to it. Therefore, by increasing the weights of outgoing links and decrease the weights of ingoing links the result can improve. Moreover, increasing depth (distance from original word in the wikipedia graph) improves the results, however it is very time consuming and the improvement is not very significant. Threshold, which shows the importance of links (importance degree) have a big influence on results. For example by changing the threshold of ingoing links vector to 100, the result become negative, therefore choosing the best threshold is very important. The result of both ingoing and outgoing links shows that the best option for importance degree is 500. Threshold more than 500 is the pages with the general concepts and threshold less than 500 are not trustworthy enough for expansion and choosing them can increase the noise. For number of links or expansion size, it is also very important to choose the value. However, by choosing the threshold we control the number of links that we can add for expanding one word, by using the number of links or expansion size we limit the size of the vector. In average Wikipedia pages have more than 100 links and consider all the links for expansion is impossible. Therefore by choosing the expansion size we can control the size of vectors. Results also show the importance of choosing this value, for example for expansion via outgoing links, the best value is 10 extra links.

4.4.3 Combinations of values

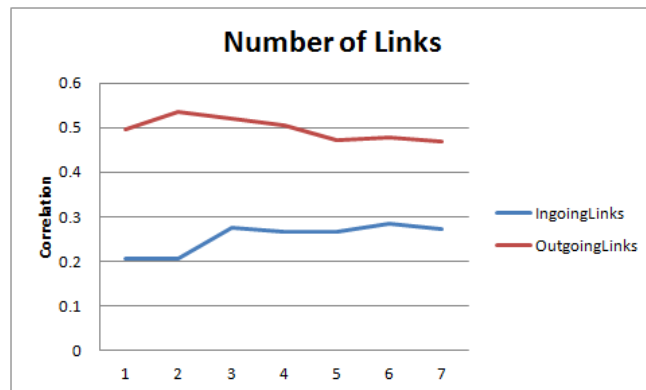
Choosing the best values for experiment is very difficult. It is impossible to use all the combinations of the values, therefore we use greedy [70] way to choose the best option. Therefore, based on the results that we discuss in previous section, we choose the best values for combining the values in the vectors.



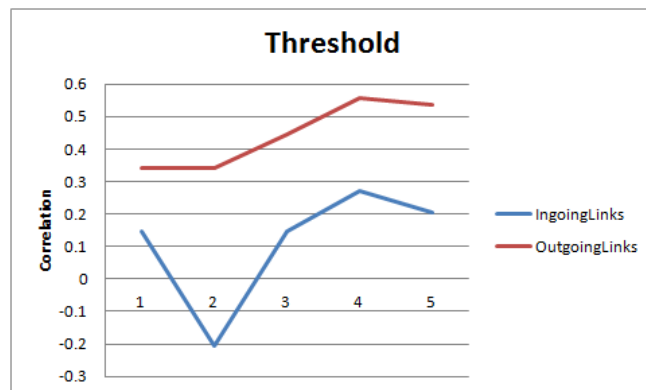
(a) Wikipedia Link Weights



(b) Wikipedia Link Depths



(c) Wikipedia Expansion Size



(d) Wikipedia Threshold

Figure 4.4: Effects of Wikipedia values on results

Chapter 5

Retrieval Framework

In Chapter 2, we presented the information retrieval framework, retrieval techniques and retrieval models. In this chapter we described our system model based on those techniques, and using VSE approach for retrieving patent documents. Similar to most retrieval systems, our model has two phases, pre-processing phase and processing phase. We describe these two phases in more details in addition to their technical designs in this chapter. At the end of this chapter, we briefly introduce datasets, test collections, libraries and tools that we use for this study. Finally at the end of this Chapter, we briefly introduce the data-sets, tools and libraries that we use in our system.

5.1 Introduction

The proposed VSE method, allows integration of one or more knowledge sources in the system. In previous chapters, we explained how VSE can work with two of these knowledge sources which are available on the web, VSE-WordNet and VSE-Wikipedia. In this section, the system model of our proposed method is presented. This system model is not the only model that can work with VSE. We choose this model to present our work and implement the whole system. In Chapter 2, we explained how information retrieval systems works and different techniques that are used for indexing are presented. Here we explain how we use these different techniques in our system.

Figure 5.1, shows the use-case diagram of our system model. Any retrieval system has two main phases, pre-processing and processing. In pre-processing step of our system, user can do normal or semantic indexing. Normal indexing system is used for normal search; and, query expansion and semantic indexing system is used for document expansion search. Therefore, in processing phase, user can submit query and perform three different kinds of search (Normal, Query expansion-based and Document expansion-based), and as a result, system returns a ranked list results with measures that helps user to compare these results with each other. The rest of this chapter discusses these two phases in more details.

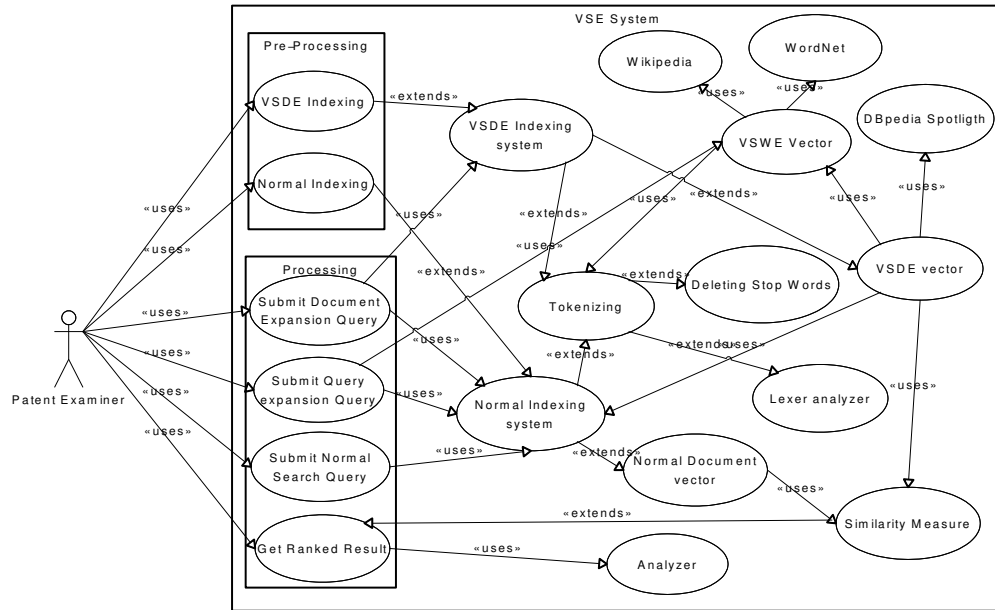


Figure 5.1: UseCase diagram

5.2 System Model

The first step in most retrieval systems is Pre-processing phase, which is performed once for all future retrieval tasks. In pre-processing phase of our system, we have to perform nine steps as follows:

1. Tokenizing
2. Deleting Stop words
3. Token-Document indexing system (Normal Indexing System)
4. VSE-Token vector
5. Token-Token vector indexing system
6. Token Dictionary Builder
7. Reference vector
8. VSE-Document vector
9. *Optional: Document-Document vector indexing system

First three steps are common steps for the normal indexing system. The normal system is a vector space model (VSM) that we explained in Chapter 2.

The second phase of retrieval is at the processing time which is Processing phase. In our system model, there is no limitation for the length of query string, and the system can handle the whole document as a query. Since our system is a patent retrieval

system, the length of the query string may be very long. The main purpose of the system is prior art candidate (PAC) search. Therefore the system is useful for patent examiners who do this task. For this task, as we explained in Chapter 2, patent examiners submit several queries to the system to retrieve similar patents. Since we are not expert in this field and we do not have access to examiners, we use different parts of a given patent as a query, for extracting similar documents in both normal search and VSE-based searches.

For Processing phase we have to perform the following five steps as follows:

1. Tokenizing
2. Deleting stop words
3. Construct query string vector
4. Comparison between query string vector with data-set document vectors based on similarity measure
5. Construct ranked result with Analyzer

Step three can be changed according to the user choice. In this step, if the user chooses to use semantic search, query string vector is constructed based on:

1. VSWE-Token vector
2. Loading reference vector
3. VSE-Document vector (VSDE)

Furthermore, step four may also be changed according to user choice. If the user chooses document expansion search, the comparison between the query string vectors is done with expanded document vectors inside the collection. In other words, if the user chooses query expansion or normal search, the system uses the normal indexing system and if the user chooses document expansion, the system uses semantic indexing system.

In the next two sections of this chapter, we explain these two phases in detail.

5.3 Pre-Processing

In this section, we explain pre-processing phase. Figure 5.2 shows different parts of the pre-processing phase. In this phase, based on the user choice, the system makes the indexing system. Indexing is a very important step in information retrieval systems to increase the performance of the systems. The user has two choices for building the index system, normal indexing system which is used for normal search and query expansion search; and semantic indexing system which is used for document expansion. In this section, our main focus is to explain the semantic indexing system; we explained the normal indexing system in Chapter 2.

According to the characteristics of our system, the first step of pre-processing phase for our system is building the system property file, which is not common in most of

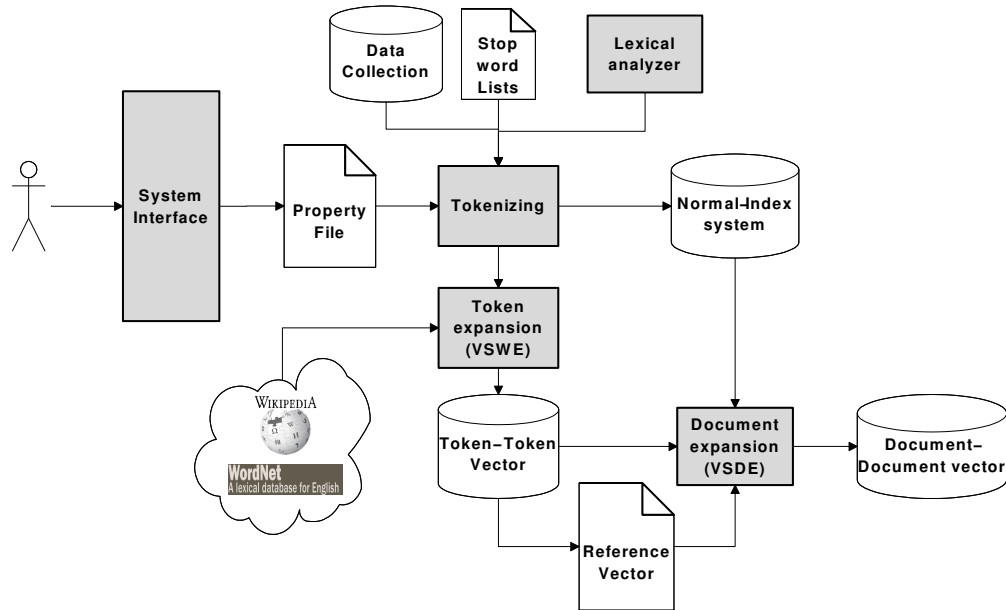


Figure 5.2: Pre-Processing phase

the retrieval systems. However, we also do this step for our normal indexing system because we have different data collections and according to the user choice, the system makes the indexing system for that collection.

5.3.1 Building System Property File

In this step, we save all the characteristics of the system. For example, if the user chooses the semantic indexing, we save all the chosen features. These features are very different from one knowledge source to the other knowledge source. For example, for building the VSE-system based on Wikipedia we have to save features such as link types (Ingoing or Outgoing or both), weights of links, depth, DBpedia spotlight, etc. Property file also have features about the retrieval system, for example, for the patent retrieval system, we have four different data-set that user can choose from, these data collections are based on different parts of patent document, such as title, abstract, claim and all (combination of title, abstract, description and claim part of patent documents). Figure 5.2 shows the GUI of our system and different features that user can select for indexing. Explanation of all the features that we could save in our system for the normal and the semantic indexing system in in Appendix B.

5.3.2 Tokenizing

Like the normal information retrieval system, we have to extract the tokens from the document. The following code is used in our system to tokenize all documents. We ignore spaces, punctuations and digits.

```
(pPunct|pSpace|(pAlpha*pDigit+pAlpha*))+
```

This step is used in both normal and semantic indexing system.

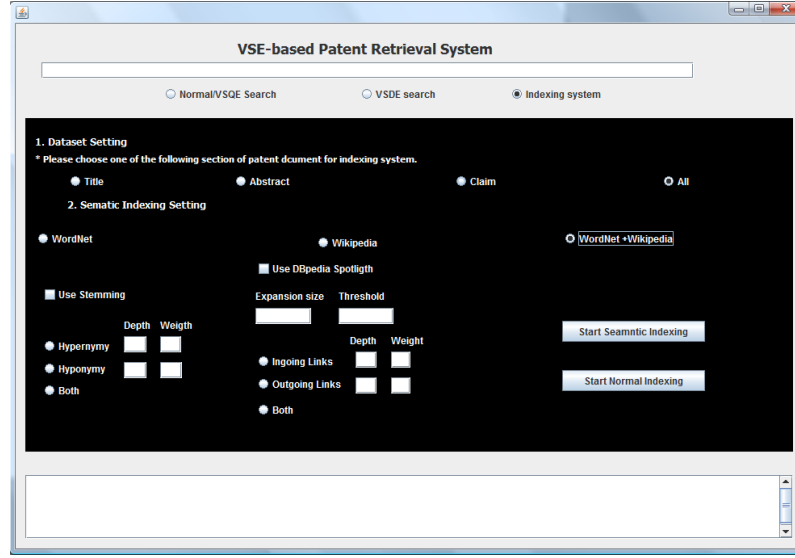


Figure 5.3: Indexing GUI

5.3.3 Deleting Stop words

We explained about the usefulness of removing the stop words in Chapter 2. In our system we remove the stop tokens from the document. We find the list of patent documents stop tokens from EPO. The whole list is available in Appendix C. This list is general English stop words like a, an and the in addition to the patent stop words like claim.

This step is also used in both normal and semantic indexing system.

5.3.4 Token - document indexing system

As we discussed indexing systems in Chapter 2, there are several indexing architecture. In this work, we use inverted index. Also we choose Hash-table as our main indexing data structure. By using hash-table we can easily calculate TF-IDF of each token for further analysis.

For semantic indexing system, if user chooses Dbpedia spotlight to annotate the tokens inside the document with Wikipedia pages, in this step the tokens with their Wikipedia page title for each document is also stored in a hash-table. Therefore, for each document we have all the tokens in addition to Wikipedia page titles.

5.3.5 VSWE- token vector

This step is only used for semantic indexing. The main differences between our system with normal system is how we use knowledge sources to expand the document. For this purpose, for semantic indexing system, first we expand each token by using knowledge sources. The *VSWE- token vector* is constructed for each token based on knowledge sources as we explained in Chapter 3.

In this step, each expanded token inside the token vector has an weight based on user's choice. The weight is extracted from property file that is stored in the first step.

5.3.6 Token- Token vector indexing system

This step is only used for semantic indexing. In this step, we construct a hash-table to have an access to semantic token vector of each token. In processing phase, when the user submits the query string, system first look in to this hash table to retrieve its semantic vector. VSWE for query tokens are only constructed when system do not find the token inside this hash-table. Therefore, by using this hash-table, we can easily reduce the processing time. The probability of accessing the token vector in processing time has a direct relation with size of the hash-table. However, the size of this hash-table increases the pre-processing time. The size of hash-table can be varied based on the size of collection.

5.3.7 Build Token dictionary and Reference vector

This step is only used for semantic indexing. In previous step we expand each token to number of tokens which are extracted from knowledge sources. Then we construct a vector out of them for each token. In this step, we build a dictionary out of these token vectors. By using this dictionary we can analyze how many tokens we use for our data-set. For example in our system we use [0-25] tokens to expand each token. The data-structure of this dictionary is also a hash-table and each token is assigned to one number as their ID. Based on this dictionary we can build a reference vector for the documents inside the data-set. we use token IDs to build a vector named *reference vector* and then for each document we compare their vector with this reference vector to have an uniform document vector.

5.3.8 VSDE- document vector

This step is only used for semantic indexing. In this step, we construct a *SDE- document vector (VSDE)*. We use the hash-table of step in Section 5.3.4 to retrieve all the tokens in one document. Then by using the hash-table of VSWE of step in Section 5.3.5 we retrieve the token vectors of these tokens and then merge them to construct a document vector as we explained in Chapter 3.

5.3.9 Document- document vector indexing system

This step is only used for semantic indexing. This step in pre-processing phase is optional. To have a better performance and reduce the processing time we do this phase in pre-processing phase, however, there is a tradeoff between processing time and size of indexing system. If we construct uniform document vectors in pre-processing time, we need a large memory to save their vectors. These vectors are sparse vectors and most of the vectors have lots of zero values.

After constructing a document vector for each document, we save them in a hash-table. At processing time, it is very easy to compare these uniform document vectors with each other.

The other way of reducing the processing time without using lots of memory is using inverted index. After constructing the VSDE vectors, we can make an inverted index system based on expanded documents. An important difference with normal inverted index system is the weights of the expanded tokens.

5.4 Processing

At processing phase, user can choose to submit a query based on normal search, query expansion search or Document expansion search. By choosing any of these systems, the results are different from each other. Figure 5.4 shows the processing phase. In normal and query expansion search, system use the normal indexing system and for document expansion search system use semantic indexing system. For both document and query expansion search the query string is expanded semantically based on user choice, however, for document expansion this expanded query string is compared with expanded documents inside the collection and for query expansion this expanded query string is compared with original documents inside the collection.

As it shown in Figure 5.4, in this phase and for all three kind of searches, the first four steps in pre-processing phase which we explain in Section 5.3 is repeated. Therefore, we do not explain these four steps in this section. All the characteristics of system are also loaded from property file which is stored in the property file.

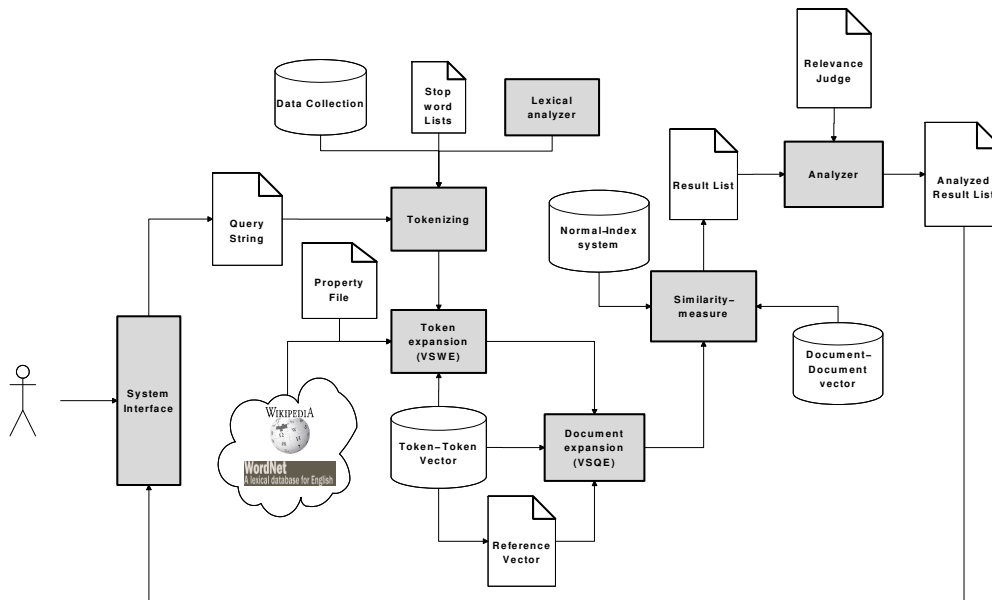


Figure 5.4: Processing phase

5.4.1 Normal Search

Normal search is based on VSM that we explained in Chapter 2, therefore similarity of query string vector with documents inside the collection is calculated based on cosine similarity (Chapter 4). In Chapter 4 we proved the efficiency of using cosine similarity compare to other vector-based measures.

5.4.2 VSQE Search

After tokenizing, the semantic vector of query string is constructed based on the property file as we explain in Chapate 3. Then the similarity between this vector and other

documents which are extracted from normal indexing system is calculated based on cosine similarity.

5.4.3 VSDE Search

Query string for document expansion is constructed like query string vector of query expansion. Then the similarity between this vector and other expanded vectors, extracted from semantic indexing system, is calculated based on cosine similarity.

5.4.4 Analyzer

Ranked results are present to user in decreasing order. To show the performance of our system to user, we have two strategies:

1. First, we analyze the result based on different evaluation measures, which we explained in Chapter 2 and Chapter 6. For each query the evaluation measures like recall shown to user with rank result list.
2. Second, user can do semantic and normal indexing in pre-processing. Therefore in processing time, there is a possibility to compare the result of one query based on three different search options.

By using these two strategy, user can get the validation of results based on evaluation measures of each query in addition to comparison of semantic based results (query and document expansion) with normal searching system. We explain more about analyzer and evaluation measures in next Chapter 6

5.5 System Tools

In this study, we use different resources. Each of these resources has their own characteristics. In this section, we discuss the data collection that we use clef-ip 2011 and then we explain how we adopt WordNet and Wikipedia as our external knowledge sources in the system. We also explain about other tools that we use in our system such as DBpedia spotlight

5.5.1 Patent Data-Set

There are not several data-set and training set for patent retrieval systems especially for prior art candidate search. We use clef-ip 2011 [71]: prior art search track. The whole clef-ip 2011 dataset include 2.6 million patent documents. Patent documents in this collection have specific format as it shown in Figure 5.5.

EP-0000001-A1		
Country Code	Doc- Number	Kind Code

Figure 5.5: Patent document format

These topics are patent documents in A1 or A2 format.

1. **Country code:** patent documents inside clef-ip 2011 data-set is gathered from 2 different sources: European patent organization (EPO) [72] and World Intellectual patent Organization (WIPO) [73]. Therefore the country code of patent document inside collection is EP or WO.
2. **Doc-Number:** Each patent document inside the collection have a 7 digits code as a document identifier.
3. **Kind code:** Patents have two different category :A and B. Patent documents in category A are not granted and need to be reviewed by patent examiners for invalidity. The patent documents in category B are granted in patent office. In kind- code, in addition to patent category, there is a number. If the patent belongs to category A, the number shows the number of times that patent document is reviewed by patent examiners. If the patent belongs to category B, the number indicates the number of times that patent is granted. Each granted patent documents (patent documents in B) can have several numbers based on different versions of patent based on additional features and claims.

Training set

For testing purposes, in this data-set there are 300 topics. Each topic is a patent document without citation information. The training set of clef-ip 2011 is formatted as follows: The first column is patent topics and the second column is a relevant doc-

Patent Topic	Relevant patent	Relevance degree
EP-0000001-A1	EP-7654321	1
EP-0000001-A1	EP-7654322	2
EP-0000001-A1	EP-7654323	1

Table 5.1: Clef-ip 2011 training set format

ument to that topic. Third column show the relevancy degree.

The topics in clef-ip 2011 data-set are distributed uniformly between three languages (French, German, and English). We choose to work on English documents. Therefore we have 100 English topics out of 300 topics. Based on these 100 English topics we have 662 relevant patent documents. These patent documents do not have any categories. Therefore, by extracting all the categories and version of these relevant patent documents we have 1177 relevant patent documents. These relevant patent documents are also distributed in three languages (English, French and German). Therefore, we extract the English documents which are 906 patent documents. To analyze the result of our proposed method on different part of patent document we extract the title, claim, abstract and description of all of these patent documents. based on our analysis, most of the patent documents do not have all parts of patent document. The following table shows the number of patent documents that have each part:

Another analysis that we do for clef-ip 2011 dataset is based on the length of each of these parts:

According to the limitations of the system that we implement our system on it, the complete clef-ip data-set analysis and use is impossible. The complete clef-ip

Topic	Abstract	Description	Claim
906	423	562	561
100%	46%	62%	62%

Table 5.2: Frequency of tokens in each patent parts

Tokens	Topic	Abstract	Description	Claim	All
Average length	15	150	7000	2150	9315
Totally	14434	74002	4619001	1417361	6124798

Table 5.3: Average length of each patent part

2011 data collection is from MAREC data collection and consist of 2.6 million patent documents, 1.3 million patent documents from EPO and 1.3 million patent documents from WIPO. In Chapter 6 we explain how we make our test collections out of these parts.

5.5.2 Wikipedia

Wikipedia data-set is available for public in Wikipedia website [39]. There are several dump files from Wikipedia pages based on different date. English Wikipedia dump files are available in XML and SQL format. For this study we use dump file of March of 2011. The files that we used are :enwiki-20110317-pages-articles.xml.bz2, enwiki-20110317-pagelinks.sql.gz. To convert Wikipedia dump files to Mysql database, we use JWPL java library [74]. After constructing Wikipedia database, to improve the performance of system, we construct a simplified version of the whole Wikipedia database. Our simplified Wikipedia database has three tables:

1. Page : which contain the page id and the page title and the number of ingoing links to that page
2. Page_Inlinks: this contains the page id and the inlink page id
3. Page_Outlinks: which contains the page id and the outlink page id

In the following figure, the relations between these three tables are presented.

In the simplified version of database that we construct from Wikipedia, the number of links and pages are listed in the following table:

Name of the table	Page title	Pagelinks
Number of rows	3623907	403760626

Table 5.4: Number of entities in our Wikipedia Database

According to Table 5.4, in average for each page in Wikipedia there are least 111 links (ingoing or outgoing links). Therefore, without limiting the number of links that we want to use in our expansion, the length of expanded vector become so long. Increase the size of the expanded vectors lead to increase the size of index system.

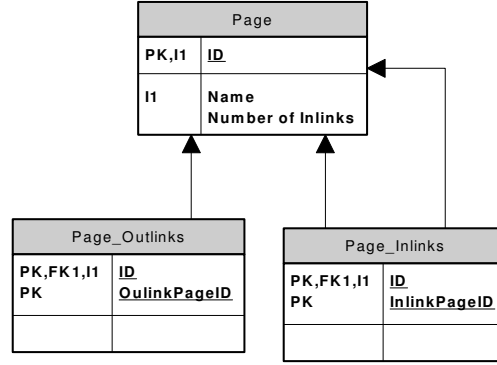


Figure 5.6: Our Wikipedia Database

Size of index system is also influence the processing time. Therefore, use all the links for expansion, if is not impossible, is time and cost consuming.

5.5.3 WordNet

Wordnet dataset is available via wordnet website [38]. According to the specific structure of WordNet and its relations, we use JAWS library [75]. A JAW is a java library that implements the access of wordnet relations such as Hyponymy and Hypernymy. In our system we use the WordNet 3.0.

5.6 Word Sense Disambiguation (WSD)

One of the important factors in VSWE which we show by evaluation in Chapter 4, that improves the results in WordNet is using sense disambiguation algorithm called Lesk [68]. For document expansion, using Lesk as we use for word expansion is very time consuming, therefore, we don't use sense disambiguation in document expansion via WordNet. However, we use sense disambiguation for Wikipedia expansion. To annotate the tokens inside the document to correct Wikipedia page (correct sense), we use an external tool: Dbpedia spotlight. Here, first we describe DBpedia spotlight and then explain how we use it in our system.

5.6.1 DBpediaSpotlight in VSDE/VSQE- Wikipedia

Dbpedia spotlight is a tool for linking unstructured information to the Linked Open Data cloud through DBpedia [76]. Therefore, the main purpose of Dbpedia spotlight is linking to correct DBpedia pages (Wikipedia pages). Since Dbpedia spotlight have sense disambiguation, the annotated Wikipedia pages are mostly annotated to the correct sense of the word. Therefore to construct the document vector from Wikipedia, in addition to the expansion method that we explained in Chapter 3 to construct VSWE-Wikipedia, we use Dbpedia spotlight to find correct Wikipedia pages. One of the problems that VSWE-Wikipedia has for query/document expansion is multi parts word. In Wikipedia, most of the page title is not just a one word, therefore when we are looking for one word in Wikipedia we cannot get these pages. Moreover, if we have the

exact page title in our text, by tokenizing we separate these words from each other, for example United States become united and states after tokenization. Using dbpedia spotlight help us to solve these two problems that VSWE-Wikipedia cannot solve (WSD and multi parts word). After using Dbpedia Spotlight, the final document vector is combination of all VSWE-Wikipedia of all the tokens inside the document with VSWE-Wikipedia of all Dbpedia spotlights annotated pages. Algorithm 8 shows how we construct query/document vector by using Dbpedia spotlight.

Algorithm 8 vector-based Query/Document Expansion -Wikipedia

```

DocumentVector  $\leftarrow$  null
Index  $\leftarrow$  0
DBpediaPages  $\leftarrow$  DBpediaSpotlight(Document)
{DBpediaPages contains tokens and annotated wikipedia pages}
for  $i = 0 \rightarrow \text{Tokens.length}$  do
  if  $\text{Tokens}[i] \in \text{DBpediaPages.tokens}$  then
    TokenVector  $\leftarrow$  TokenVector(DBpediaPages.annotation)
  end if
  for  $j = 0 \rightarrow \text{TokenVector.length}$  do
    DocumentVector.token[index]  $\leftarrow$  TokenVector.token[j]
    DocumentVector.weight[index]  $\leftarrow$  TF - IDF(Tokens[i])  $\times$ 
      TokenVector.weight[j]
    Index  $\leftarrow$  Index + 1
  end for
end for

```

5.6.2 DBpedia Spotlight

Based on the DbpediaSpotlight site [77], in their core application they use enwiki-20100312-pages-articles.xml version of dump file from Wikipedia, which means the English Wikipedia pages until 12 of march 2010. In our application we use the web service to annotate the patent documents to Wikipedia pages. It is a RESTful/SOAP Web API, which exposes the functionality of annotating. Following code shows an example of using this tool via web service, the results from the web service can be XML, Html, JSON. As an example, here we present the result of Pop star Michael Jackson in XML format.

```

<Annotation text="Pop star Michael Jackson"
confidence="0.3" support="30"
types="Person,Place,...">
<Resources>
<Resource URI="dbpedia:Michael_Jackson"
support="5761"
types="MusicalArtist,Artist,Person"
surfaceForm="Michael Jackson" offset="9"
similarityScore="0.31504717469215393"
/>
</Resources>

```

</Annotation>

This webservice use two parameters for annotating:

1. **Disambiguation Confidence:** this parameter presents the topical pertinence and the contextual ambiguity. Based on their paper [78], the best value for this parameter is 0.7.
2. **Support:** number of links that Wikipedia page must have at least for annotation.

We also set the Disambiguation Confidence to 0.7. To get a reliable page and also have a high recall based on DBpedia spotlight we choose set our support value to 50.

Chapter 6

Empirical evaluation of VSQE and VSDE

6.1 Introduction

To find out the usefulness of our approach (VSDE and VSQE), we have to perform several tests. In this chapter, we examine our approach on patent documents to show how semantic expansion can improve the prior art candidate search. In Chapter 5, we briefly introduced our datasets, the libraries and tools that we use for this study. In this Chapter, we present our results based on the experiments. At the end of this Chapter we discuss the advantages and disadvantages of using semantic expansion for retrieval purposes.

6.2 Test Collections

In Chapter 5, we explained about clef-ip data collection. Here, we introduce our test collections based on clef-ip which are used for this study as test collections:

1. **Title/Title:** This collection consists of all English titles of relevant patent documents. As we explained in Chapter 5, there are 100 English patent topics and 906 English relevant patent documents have title. Therefore number of documents inside this collection is 1006 documents, and our queries are 100 titles.
2. **Abstract/Abstract:** This collection is constructed the same as Title/Title. However, there are 100 English abstracts and 423 English relevant patent documents have abstract. Therefore number of documents inside this collection is 523 documents, and our queries are 100 abstracts.
3. **Claim/Claim:** This collection is constructed the same as Title/Title; However, there are 100 English claims and 561 English relevant patent documents have claim. Therefore number of document inside this collection is 661 documents, and our queries are 100 claims.
4. **Title/All:** This collection includes all English relevant patent documents. In this collection we have 1006 patent documents. As we explained in Chapter 5,

some of the documents inside this collection do not have all the parts. Documents inside this collection are text files which are extracted from XML patent documents and have title, abstract, description and claim. For those documents which do not have one (more) part(s), we just add the other part(s) inside the documents. For this collection we use 100 titles.

5. **Abstract/All:** This collection is the same as Title/All collection, however, in this collection our queries are 100 abstracts.
6. **Claim/All:** This collection is also the same as Title/All collection, however, in this collection our queries are 100 claims.

To demonstrate the effects of query and document length on VSDE and VSQE, our expansion evaluations are based on different parts of patent documents. However, according to the limitation of our system, we do not use claim, description or whole document as a query/document in some of our experiments. Moreover, we ignore description part in both document and query, because based on literature review, description is not a good target for PAC [79], [80], [81], [1].

6.3 Evaluation approach

To measure the effectiveness of our approach we have to use Retrieval Evaluation. In Chapter 2, we introduced four measures. Here we explain how we use these measures in our system.

The given query to the system is the topic document, and by using test collection, we know the relevant and non-relevant patent documents for each topic. For each topic documents we retrieve first 100 results and then we use these results for calculating precision. We calculate the precision by calculating the number of relevance documents (relevant patents to the patent topic from training set). For recall-rate we just ignore the zero values, therefore recall is number of relevant patents to the patent topic by all the relevant patents which are inside the collection for the given query. Since we do not use all the documents and our collections are different from each other, therefore we first calculate number of relevant documents to each patent topics inside the collection and then by using this number as a number of relevant patents, we calculate the recall. For Average precision and NDCG we use their Equation in Chapter 2. We also calculate the MAP, Mean NDCG, mean recall and Mean precision as follows:

$$Mean Recall = \frac{\sum_{q=1}^N Recall(q)}{(|\{Q\}|)} \quad (6.1)$$

$$Mean Precision = \frac{\sum_{q=1}^N precision(q)}{(|\{Q\}|)} \quad (6.2)$$

$$Mean NDCG = \frac{\sum_{q=1}^N NDCG(q)}{(|\{Q\}|)} \quad (6.3)$$

All the comparisons is based on these factors: Average recall, Average precision, MAP and NDCG. We use the relevancy degree of patent documents in calculating

NDCG. However, there is no explanation about these values and their values in Clef-ip test collection are only one and two. Therefore the accuracy of these numbers is not specified.

6.4 Results of VSQE

For query expansion via VSQE, we use all test collections that we introduce in Section 6.2. Moreover, to find which knowledge source suits for expansion, we compare the results of WordNet, Wikipedia and their combinations.

In query expansion the effects of adding weights to expanded tokens is very low. Based on our experiments, changing the weights for both Wikipedia and Wordnet do not influence the results. We believe, since query expansion increase the numbers of tokens in the query string to extract more documents, changing the weights do not affect too much. The values that we choose for VSQE for both WordNet and Wikipedia is based on our experiment for Word-Word similarity in Chapter 4.

6.4.1 Query length

First, we evaluate based on our first three test collections. Our aim of using these collections is to differentiate between query lengths. We compare the results of Title/Title and Abstract/Abstract based on Wordnet, Wikipedia and their combinations. For Claim/Claim we just evaluate based on WordNet and Wikipedia. Based on the limitation of our system, evaluating for Wikipedia takes 1 week, therefore we do not evaluate Claim based on Wikipedia and WordNet. Table 6.4.1, shows the results of our evaluation.

Test Collection	Method	Mean Recall	Mean NDCG	MAP	Mean Precision
Title/Title	Normal	58.03%	42.73%	25.16%	5.07%
	VSQE-WordNet	67.96%	39.49%	19.97%	5.35%
	VSQE-Wikipedia	66.27%	40.45%	18.62%	5.36%
	VSQE-WordNet and Wikipedia	71.52%	39.30%	17.37%	5.13%
Abstract/Abstract	Normal	91.08%	53.25%	38.58%	3.83%
	VSQE-WordNet	92.55%	44.59%	27.78%	3.55%
	VSQE-Wikipedia	93.87%	43.52%	24.46%	3.54%
	VSQE-WordNet and Wikipedia	94.46%	40.77%	22.14%	3.52%
Claim/Claim	Normal	95%	56.20%	40.77%	5.21%
	VSQE-WordNet	95%	48.94%	27.87%	4.41%
	VSQE-Wikipedia	95%	46.37%	24.36%	4.23%

Table 6.1: VSQE results based on Query length

Based on Table 6.4.1, our expansion technique increase the recall. However, the effects of expansion with the shorter queries are more significant. As you can see, for Claim, recall do not change after expansion, however in title, expansion increase the recall more than 20%. There are two main reasons that lead to have a better results for shorter queries, first, expansion aim to increase the number of tokens in the query therefore longer queries do not change a lot with expansion, and second, based on the results of normal system, most of the relevant documents use the same vocabulary, therefore expansion cannot be more effective. Except Recall-rate, the other evaluation

measures are decreasing. We discuss the weakness of our expansion at the end of this Chapter, as a discussion, in Section 6.8.

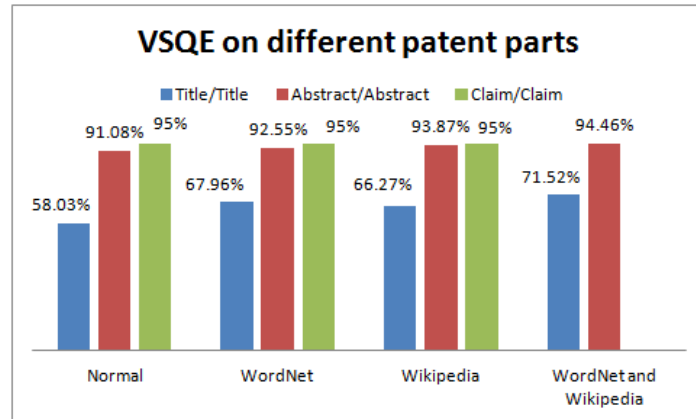


Figure 6.1: Comparison between different data collections based on query/document length, knowledge source and patent parts

6.4.2 Documents length

In previous part, we compare different queries based on their lengths. The main difference between this Section and previous section it that, we compare different queries based on same collection of document. In previous section the length of the documents inside the collections are also different from each other. In this part, we compare the results based on Title/All, Abstract/All and Claim/All. In this part, we compare the result based on query length like previous part and then document length by using previous results. In next part, based on the results of this part and previous part we also compare the effects of different patent parts for retrieving relevant documents.

Test Collection	Method	Mean Recall	Mean NDCG	MAP	Mean Precision
Title/All	Normal	74.46%	50.80%	29.09%	6.05%
	VSQE-WordNet	82.36%	44.17%	21.21%	5.8%
	VSQE-Wikipedia	80.47%	43.57%	18.67%	5.53%
	VSQE-WordNet and Wikipedia	84.68%	42.37%	17.83%	5.42%
Abstract/All	Normal	84.25%	57.06%	38.01%	7.22%
	VSQE-WordNet	90%	48.50%	24.13%	5.96%
	VSQE-Wikipedia	88.94%	46.52%	21.54%	5.64%
	VSQE-WordNet and Wikipedia	91.74%	44.27%	18.18%	5.42%
Claim/All	Normal	88.47%	57.46%	37.20%	7.15%
	VSDE-WordNet	91.14%	47.90%	21.78%	5.6%

Table 6.2: VSQE results based on document length

6.4.3 Patent parts

For retrieving similar patent documents, we try different queries with different length in this study. We use title, abstract and claim. We also retrieve them based on their own parts or the whole document. The result in Table 6.4.2shows that there is a little

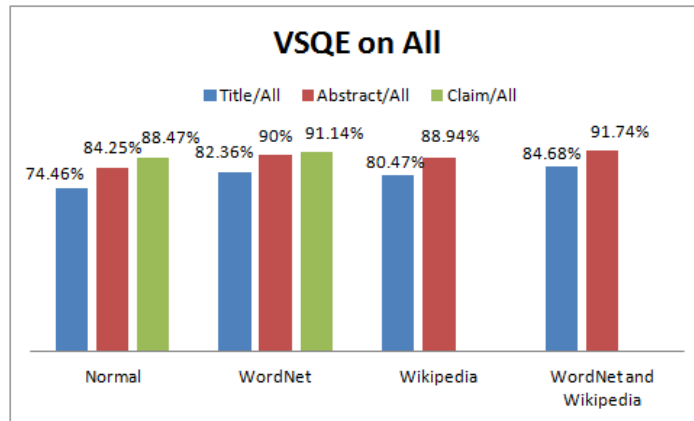


Figure 6.2: Comparison between different data collections based on query length, knowledge source and patent parts

differences between abstract and claim for finding similar patents. Moreover, using different parts of patent document for retrieval can decrease the number of similar documents; however, the precision is higher. Based on the results, except for title, finding relevant documents with query string of claim part or abstract part based on them have better results than finding them based on the whole patent document. The reason that title is performing better on whole document can be the limitation of words inside titles. Therefore, in title, there are not enough information for retrieval and finding expanded titles based on the whole document is more effective than finding them based on titles.

6.4.4 Knowledge source

One of the important factor in semantic expansion as we explained in Chapter 4, is choosing the suitable knowledge source. In this study we use WordNet, Wikipedia and their combinations. For query expansion we examine based on all of them. As it shown in Table 6.4.2 and Table 6.4.1, combining knowledge sources help us to have a better recall-rate. Moreover, there is a little differences between choosing WordNet and Wikipedia as an external knowledge sources for expansion. However, the expanded vectors are completely different. The results for both rounds of experiment on query length in Figure 6.1 or document length in Figure 6.4.2 shows that combining knowledge sources (Wikipedia and WordNet) can be useful and efficient to find similarity between words (based on the results of Chapter 4) and documents.

6.5 Results of VSDE

As we explained in Chapter 5 we evaluate the expansion technique for retrieving patent documents based on Query expansion and Document expansion. In previous Section, we study the results of our approach on query expansion. In this Section, we evaluate the results based on document expansion. For experiment, we use first three data collections. Using the whole document for expansion needs lots of memory and it is

very time consuming which make it impossible for us to examine it on our system. Moreover, based on the results of query expansion, experiment queries (patent parts) based on their own part is more useful to show the effects of expansion, and also performs better.

We also use both knowledge sources for VSDE. We experiment different values with some examples and the results are same as our results in Chapter 4. Therefore, we conclude that for choosing the values it is not lots of differences between the unit of language, word, sentence or text. However, to make our experiment more efficient for both WordNet and Wikipedia, we evaluate on depth equal to one. Since, the graphs of both knowledge sources grow exponentially, for document expansion it is very time consuming. The other values are the same as

Test Collection	Method	Mean Recall	Mean NDCG	MAP	Mean Precision
Title/Title	Normal	58.03%	42.73%	25.16%	5.07%
	VSDE-WordNet	70.11%	39.84%	16.26%	4.49%
	VSDE-Wikipedia	70.11%	43.42%	18.57%	3.88%
	VSDE-WordNet and Wikipedia	70.11%	44.28%	18.95%	4.4%
Abstract/Abstract	Normal	91.09%	53.25%	38.58%	3.83%
	VSDE-WordNet	69.70%	42.85%	22.27%	2.81%
	VSDE-Wikipedia	69.70%	41.82%	20.97%	2.41%
	VSDE-WordNet and Wikipedia	69.70%	43.51%	22.48%	2.63%

Table 6.3: VSDE results based on Query length

6.5.1 Patent parts

Based on the results in Table 6.5, expanding documents or document expansion is not very effective in patent retrieval. Results show that only expanding titles can improve the recall-rate. For Abstract, expansion even decreases the recall-rate. We discuss this in discussion at the end of this Chapter. The main reasons can be (i) noise that expansion produce in collection and (ii) the quality of collection. Based on our literature review DE is effective for word mismatch inside the collection, and since the quality of patent documents in terms of word-mismatch is high and also expansion via our knowledge sources are not perfect, Document expansion produce lots of noise inside the collection which lead to get the low recall-rate compare to normal system. Based on the results, DE can be useful for title (shorter documents) because DE can solve the lack of information in titles.

6.5.2 Query Length

In document expansion, we get the same results as we get from query expansion. Shorter queries are more useful for expansion than longer queries. It is also true for documents inside the collection. Expansion on shorter documents is more effective.

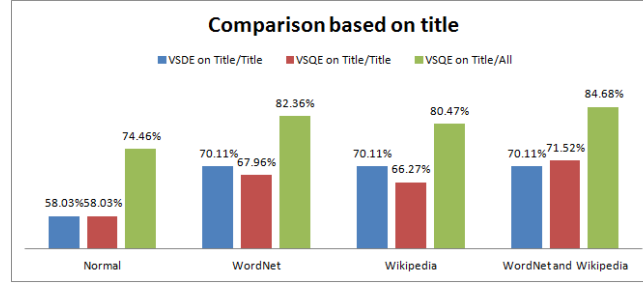
6.5.3 Knowledge source

For document expansion the results of using different knowledge sources on recall is the same, however, same as query expansion the combination of these two sources have

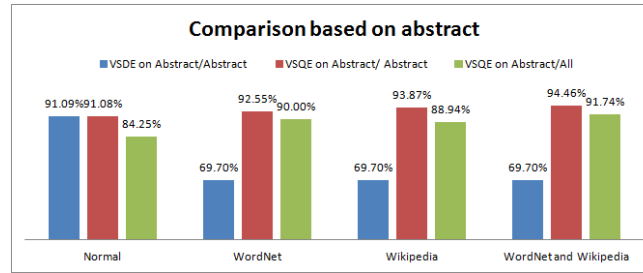
a better results compare to the WordNet and Wikipedia according to other measures (Mean NDCG and MAP).

6.6 Comparison between results of VSQE and VSDE

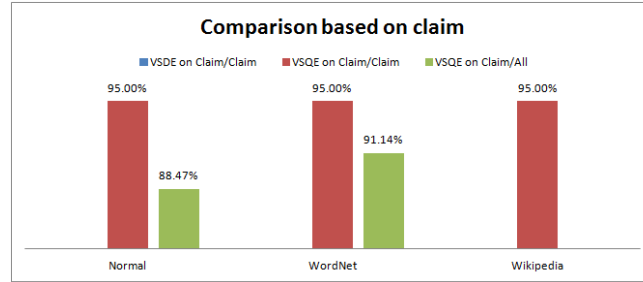
In previous sections we evaluate based on VSQE and VSDE, in this section we compare the results of these two techniques.



(a) Comparison based on Title



(b) Comparison based on Abstract



(c) Comparison based on Claim

Figure 6.3: Comparison between VSDE and VSQE

In Figure 6.3, we compare the results of VSDE and VSQE based on title, abstract, and claim. As we mentioned in previous parts the effects of expansion on shorter documents and queries is more than longer ones. Moreover, combination of knowledge sources in both approaches outperforms the results of using one of the knowledge sources. For both abstract and claim, the results shows that retrieving them based on their corresponding part is more useful for PAC search.

These results show that for PAC, expansion could be a solution for miss-match problem and using query expansion (VSQE) is more effective based on:

1. **Recall-rate:** By using query expansion we can retrieve more relevant documents compare to normal systems and document expansion.
2. **Time:** In pre-processing phase we do not need to calculate the expanded vectors for all documents and therefore in processing time we do not need to retrieve based on large indexing system, however, since query expansion is done at processing time, compare to normal system, query expansion increase the processing time.
3. **cost:** By using the query expansion, we do not need to construct a new indexing system and therefore we can save the memory, money and efforts for making a new one.

Query expansion, by using WordNet and Wikipedia can help both examiners and normal users to find more relevant patent documents based on their needs and could retrieve those kinds on patents which do not have the same vocabulary as the query string.

6.7 Potential of Expansion

In this section, we provide some technical advice for using expansion in IR systems. Here we provide some guides for using expansion in IR systems as results of our experiments.

1. **Quality of data collection:** Query expansion performs better than document expansion when the quality of collection is high in terms of word-mismatch or errors. Mostly, DE is used for speech recognition because the quality of collection is poor and by expansion, they can provide a better data collection for retrieval purposes. Here, we use patent collection. Although the documents inside our collection do not have all the patent parts but each part is complete and have a high quality. Therefore, document expansion for adding semantics to collections or increase the quality of collections is more recommended.
2. **Time:** Expansion is time consuming. There is a tradeoff between the performance and time. Results of this study shows that query expansion can perform better based on time and recall. Therefore, for collections that have a high quality data collection, using query expansion can be more useful and less time consume than DE.
3. **Features:** Using QE can be done with relevance feedback. In this way even the users can help to improve the quality of expansion. And this way can be done with query expansion. Another advantages of using relevance feedback is increasing the adoptability of the system by user. One of the main reasons that automatic query expansion can be harmful is user interaction. Most of the users do not expect to retrieve the documents that they do not search, therefore based on [82], expansion with suggestions are the best approach for using expansion in IR systems. Suggestion help users to formulate their needs better and clarify the ambiguity.

4. **Quality of Knowledge source:** Based on our result, the quality of knowledge source is more harmful for De than QE. Therefore, noise in DE can decrease the quality of retrieval more than noise in QE. In this study we use the same kind of knowledge sources and approaches for De and QE and result shows that with the same amount of noise QE can be more useful than DE.

6.8 Discussion

In this section, we discuss some reasons that lead to get a bad results from query and document expansion via our approach.

6.8.1 Field Of Expansion

We study the impact of our semantic expansion approach on patent retrieval systems, and this area is just a specific case. Therefore, by this study we cannot judge about the usefulness of our approach in other areas. As we did experiment on non-technological terms in Chapter 4, this approach compare to other semantic-based similarity measure performs good. We also do some test experiments to see whether this approach can use for documents of other areas or not. For example, we assume that we have different documents from different areas, the concepts of these documents are different from each other: animals, fruit, furniture, business, etc. Then we try to retrieve documents in that specific field without using the exact word that is mentioned inside the document. For example, for our fruit case, we have a document which talks about apple and then we search for orange. In the result list, normal search do not return anything, however, semantic search (both query expansion and document expansion) returns the document about apple. Therefore, the area of retrieval has a big influence on results. Moreover, since our case study is patent retrieval, in this study we use long queries; however, for most of the retrieval systems, users use very short sentences or words to describe their needs. We believe expansion can be more useful for shorter queries. We also prove this based on our results in Section 6.4.

6.8.2 Noise

One of the main problems in using expansion for retrieval purposes is noise. Especially for semantic expansion, and by using external knowledge sources, there is a high probability of producing noise in the system. Since volume of expansion in document expansion is more than query expansion, distribution of noise in documents is more than query. We categorized the noise in to two categories as follows:

1. **Irrelevant noise:** Expanded words which are completely different from the concept of the original word, for example, if we expand the word `table` with `car`.
2. **Relevant noise:** Expanded words which decrease the semantic, for example, expanding the word to very general or specific words, such as expanding word `table` to `entity`.

There are several reasons which we find that could produce noise.

1. **Knowledge source:** Choosing not suitable knowledge source can produce noise in the system. By using WordNet relation, we have to take care of relevant noises, and by using Wikipedia we have to take care both irrelevant noises and relevant noises. Moreover, by choosing WordNet for very specific case such as medical science, expansion cannot be a added value. In this study, we use WordNet and Wikipedia; however, it would be better to use a specific area with a specific knowledge source to measure the usefulness of expansion, like medial area and UMLS.
2. **WSD (word sense disambiguation):** since in English, words can have several meanings, choosing the correct meaning of the word is very important. Without using WSD, there is a high probability that we produce lots of irrelevant noises.
3. **Structure of Knowledge source:** Even suitable knowledge sources can produce noise if we do not study their structure. As we mentioned in Chapter 3, both Wikipedia and WordNet have a un-uniform shape. Therefore, using their links even for general retrieval systems can produce lots of noise (irrelevant and relevant noises).

One of the solutions which can solve these kinds of problem is choosing weight for the links. In this study we choose same weight for all the links which are going out from one node, however, it can be a better choice to give weights to the links based on their information content (ICI). One of the reasons that Resnik measure perform better in measuring the similarity can be choosing this feature. Based on the time limitation for this study, we leave it as a future work to add weights to the links based on their average information content of the nodes. This can solve un-uniform shape of the knowledge sources. Moreover, in this way we can easily find a correct combination of values, therefore we do not need to use greedy algorithm. Greedy way of choosing value, most of the time makes it impossible to find the optimal value.

In this study we also design a noise detection system which prevents the expansion of noise in vectors. We ignore expended tokens which do not have any common nodes with the other vector. By using this noise detection, we try to prevent expansion which do not have any added value. However, the results of our noise detection system are not very significant, but it can be possible to use noise detections based on the characteristics of each knowledge sources or data collections. We also leave this part as a future work to find a noise detection which suits our system.

Moreover, selecting the suitable relations and nodes in knowledge sources are very important. In this study for Wikipedia we choose link structure. To find the expanded tokens we choose the links with more ingoing links, however, to choose more suitable links we should take the semantics of links into account, for example assume we want to expand the word Radio with Wikipedia, therefore by choosing in link size the expanded vector is:

(Radio,Air band,Reginald Fessenden,Radio direction finder,On off keying,United States Court of Claims,History of radio,Radio navigation,Radio controlled car,Telstar,Table of years in radio)

However, in this expansion, some of the expanded tokens are really irrelevant. To find more related links, now assume we make expanded vector of radio based on the

order of links inside the text (description of Radio in Wikipedia page), therefore the vector is as follows:

(Radio, Transmission, Electromagnetic waves, Frequencies, Visible light, Travels, Electromagnetic fields, Vacuum, Modulating, Amplitude, Frequency)

We leave the evaluation of this for future work. We believe by choosing better nodes we can reduce the amount of noise in the expansion.

6.8.3 Retrieval Model

Based on our experiments on VSQE and VSDE, except recall-rate, the other evaluation measures are decreasing after expansion. In addition to the problem that noise and field of expansion have, and we mentioned them in previous parts, probably our model for expansion is not the suitable model. Expansion introduces new situation for retrieval, which we discuss them in Appendix A, these situations make expansion different from normal retrieval systems. The way that we measure the similarity between two documents after expanding them or the way that we weight documents after expansion can influence on the quality of our expansion. Therefore, suitable framework for query and document expansion can be a good solution to solve the problems that external knowledge sources can produce in system.

As a model we use VSM and Tf-ISD weighting, however, by choosing other weighting mechanism such as BM25 [83], [84], [85], or other model such as probabilistic retrieval framework, the problem that we mentioned can be solved. Since find a suitable framework for expansion is out of the scope of this thesis, we leave this part as our future work. We could use the framework for VSE approaches in both word-word similarity (VSWE) and document and query expansion (VSQE and VSDE).

Chapter 7

Conclusions and Future Work

Patent retrieval poses specific challenges within the field of information retrieval. This is due to both special characteristics of patent documents and importance of retrieving them for both patent examiners and normal users. Granting a new patent is an exhausting task that patent examiners must retrieve similar patents to invalidate the new patent or grant it. The process of retrieving similar patents is cost and time consuming, and the quality of results is expert-dependent. Due to the difficulty of understanding a patent document for non-expert users, patent retrieval is very expert dependent task. Therefore, the main focus of this thesis is to use expansion to improve the procedure of granting a patent and reduce the workload on experts. Moreover, we aim to help non-experienced users to extract useful information from patent retrieval systems by using expansion.

We implement our semantic expansion approach namely Vector-based Semantic Expansion (VSE) for word (namely Vector-based Semantic Word Expansion (VSWE)), for query (namely Vector-based Semantic Query Expansion (VSQE)), and for document (namely Vector-based Semantic Document Expansion (VSDE)). We use Wordnet and Wikipedia as our external knowledge sources for expansion.

Our VSE-WordNet uses IS-A relations in Wordnet graph, we find the word in the graph and use its hyponymy and hypernymy links. We weights the nodes based on their distance from the original word. Our VSE-Wikipedia uses hyperlink structure of Wikipedia. We find the wikipedia page based on the words and then we use ingoing and outgoing links of that page. To find a correct Wikipedia page we add DBpedia Spotlight in our system. We also use the same weighting system that we use for our Wordnet approach. Our expansion is based on vector space model, therefore we use vectors. Using vector enables us to combine these two knowledge sources easily; therefore we also have VSE based on WordNet and Wikipedia.

7.1 Revision of Research Questions

Expansion for improving information retrieval systems is used for different systems. Based on our literature review and the experiments, expansion cannot be used for any kind of system. In this study we aim to investigate the applicability of semantic expansion on patent retrieval systems based on prior art candidate search on *clef-ip 2011*. Therefore, most of our choices, for the method and the knowledge sources are

based on our case study. However, this method could be adapted to the requirement of any other retrieval systems.

In this thesis, we aim to answer the following main research question: ***How can we use expansion to improve patent retrieval systems?***

To help both patent examiners and inexperienced users, we choose expansion technique. Since patent retrieval is recall-oriented, our objective is to increase the recall-rate. For this purpose, Chapter 3 answers this question: ***How can expansion increase the recall-rate?***. We theoretically proved that by using expansion, recall-rate increases. Then by reviewing related work, we proposed our approach for automatic semantic expansion which answers the question of ***Which expansion methods can be used for retrieval purposes?***.

The other research question, ***What kind of knowledge sources can be used for patent retrieval?***, is tackled in Chapter 3 where we discuss the process of granting a patent. We have found out the main reasons that examiners can do the PAC beside their experiences, is their knowledge of language and field. Therefore we choose WordNet and Wikipedia as our external knowledge sources.

We also study the similarity measures and review related work. Then by using our expansion technique with statistic-based similarity measures (Cosine Similarity, Dice coefficient, Extended jaccard), we answer the following research question ***What is the effects of expansion on similarity measures?***.

We compare our approach with other semantic similarity measures in Chapter 4. By evaluating VSWE on Miller and Charles dataset, we show that our approach can return acceptable results. By combining different knowledge sources, our approach can provide higher quality results. The WordNet-based (VSWE-WordNet) approach has a correlation of 0.68 and the Wikipedia-based (VSWE-Wikipedia) approach has a correlation of 0.61. Finally, the combination of these two approaches has a correlation of 0.79. We prove that using Cosine similarity outperforms the Dice and Extended Jaccard. Moreover, we study the effects of word sense disambiguation (WSD) by using Lesk's algorithm for our approach on WordNet. The result shows that choosing the correct sense of the words improves the results, and the evaluation improvement is more than 30%.

By proving the usability of our expansion technique, we try to use it for retrieval purposes and we answer the following question: ***Which expansion method can be used for patent retrieval?*** We explain query expansion and document expansion techniques in Chapter 3. We review related work in Chapter 3 and then by evaluation of VSQE and VSDE on different data collections, we examine our approach based on different query and document length, and also different parts of patents for retrieving similar documents. The framework of these techniques is also introduced in Chapter 5 which answer this question: ***How can patent retrieval systems use expansion techniques?***. The evaluation results answer the following question in Chapter 6. ***What is the relation between expansion and data-set?***

We conclude that expansion has better results for shorter queries in VSQE approach, and also shorter documents get better results in both VSDE and VSQE approaches. Expansion is time and cost consuming, therefore it is very important to choose the best approach which improves the system. In our case study, document expansion (VSDE) do not have a better results compare to query expansion (VSQE). Since VSDE takes more time and cost than VSQE, and it needs lots of memory for

saving new indexing system, we believe that for high quality data sets query expansion can be more useful than document expansion. Experimental results show that query expansion on shorter queries which are expanded with combination of knowledge sources have a better performance and on their corresponding part. However, document expansion can be more useful for shorter documents, and for documents with errors and word-mismatches.

7.2 Future Work

According to the discussion in Chapter 6, this work has lots of possibilities for further improvements and more in depth research work. To improve the usability of the system and provide high quality results in term of recall and precision, we need to solve the following problems. As we mentioned in Chapter 6, there are three main problems. The first one is **noise**. Our basic proposal for this problem is to choose a suitable knowledge source with WSD and to implement a noise detection mechanism that prevents noise distribution in the system. One of the main problems of the both knowledge sources that we used is un-uniform shape of them. We propose to change the way that we expand from them, i.e., although in this work we add weights to the links, this way of adding weight needs lots of experiments and it is impossible to find the optimal value. Therefore we propose to use the information content of the nodes in the graph to assign weights to the links.

Second problem is **field of expansion**. In this study we examine our approach based on patent retrieval systems. As a future work we can study the effects of this approach for different collections in different concepts.

The last problem that we discussed in Chapter 6, is the **retrieval model**. In this study we use vector space model with TF-IDF and vector-based similarity measures. However, according to Appendix A, expansion needs its own framework, therefore, we leave it as our future work to find a suitable retrieval model.

Finally, in this study we do not have access to real users to try relevance feedback expansion. In most of the retrieval systems, using relevance expansion can be more useful than automatic query expansion, and also by studying the users' behavior, the system can improve itself without involving any other external systems.

Bibliography

- [1] H. Mase, T. Matsubayashi, Y. Ogawa, M. Iwayama, and T. Oshio, "Proposal of two-stage patent retrieval method considering the claim structure," vol. 4, pp. 190–206, June 2005. [Online]. Available: <http://doi.acm.org/10.1145/1105696.1105702>
- [2] H. Joho, L. A. Azzopardi, and W. Vanderbauwhede, "A survey of patent users: an analysis of tasks, behavior, search functionality and system requirements," in *Proceeding of the third symposium on Information interaction in context*, ser. IiiX '10. New York, NY, USA: ACM, 2010, pp. 13–24. [Online]. Available: <http://doi.acm.org/10.1145/1840784.1840789>
- [3] D. Bonino, A. Ciaramella, and F. Corno, "Review of the state-of-the-art in patent information and forthcoming evolutions in intelligent patent informatics," *World Patent Information*, vol. 32, no. 1, pp. 30 – 38, 2010. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0172219009000465>
- [4] W. L., "Patexpert annual report," 2007.
- [5] I. J. Jureta and S. Faulkner, "Clarifying goal models," in *Tutorials, posters, panels and industrial contributions at the 26th international conference on Conceptual modeling - Volume 83*, ser. ER '07. Darlinghurst, Australia, Australia: Australian Computer Society, Inc., 2007, pp. 139–144. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1386957.1386979>
- [6] D. Cutting and J. Pedersen, "Optimization for dynamic inverted index maintenance," in *Proceedings of the 13th annual international ACM SIGIR conference on Research and development in information retrieval*, ser. SIGIR '90. New York, NY, USA: ACM, 1990, pp. 405–411. [Online]. Available: <http://doi.acm.org/10.1145/96749.98245>
- [7] S. K., "Studies on improving retrieval accuracy in web information retrieval," *Tokyo: Nara Institute of Science and Technology*, 2004.
- [8] J. B. Lovins, "Development of a stemming algorithm," *Mechanical Translation and Computational Linguistics*, vol. 11, pp. 22–31, 1968.

- [9] A. H. Lashkari, F. Mahdavi, and V. Ghomi, "A boolean model in information retrieval for search engines," *Information Management and Engineering, International Conference on*, vol. 0, pp. 385–389, 2009.
- [10] G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," *Commun. ACM*, vol. 18, pp. 613–620, November 1975. [Online]. Available: <http://doi.acm.org/10.1145/361219.361220>
- [11] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*. New York, NY, USA: McGraw-Hill, Inc., 1986.
- [12] J. Pan, C. pang Jack Cheng, G. T. Lau, and K. H. Law, "Utilizing statistical semantic similarity techniques for ontology mapping with applications to AEC standard models."
- [13] G. Salton, *Automatic text processing : the transformation, analysis, and retrieval of information by computer / Gerard Salton*. Addison-Wesley, Reading, Mass. :, 1989.
- [14] D. Roussinov and J. L. Zhao, "Automatic discovery of similarity relationships through web mining," *Decis. Support Syst.*, vol. 35, pp. 149–166, April 2003. [Online]. Available: [http://dx.doi.org/10.1016/S0167-9236\(02\)00102-1](http://dx.doi.org/10.1016/S0167-9236(02)00102-1)
- [15] L. R. Dice, "Measures of the amount of ecologic association between species," *Ecology*, vol. 26, no. 3, pp. 297–302, July 1945. [Online]. Available: <http://www.jstor.org/pss/1932409>
- [16] S. E. Robertson and K. Sparck Jones, *Relevance weighting of search terms*. London, UK, UK: Taylor Graham Publishing, 1988, pp. 143–160. [Online]. Available: <http://dl.acm.org/citation.cfm?id=106765.106783>
- [17] "World intellectual property organization (wipo)." [Online]. Available: http://www.wipo.int/aboutip/en/studies/publications/ip_definitions.htm
- [18] *MIT libraries*. [Online]. Available: <http://libguides.mit.edu/patents>
- [19] *Patent offices on the internet*. [Online]. Available: <http://members.pcug.org.au/~arhen/>
- [20] Y. Kim, J. Ryu, and S.-H. Myaeng, "A patent retrieval method using semantic annotations." in *KDIR'09*, 2009, pp. 211–218.
- [21] S. Bashir and A. Rauber, "Analyzing document retrievability in patent retrieval settings," in *Proceedings of the 20th International Conference on Database and Expert Systems Applications*, ser. DEXA '09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 753–760. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-03573-9_63
- [22] J.-H. Kim and K.-S. Choi, "Patent document categorization based on semantic structural information," *Inf. Process. Manage.*, vol. 43, pp. 1200–1215, September 2007. [Online]. Available: <http://dx.doi.org/10.1016/j.ipm.2007.02.002>

- [23] I.-S. Kang, S.-H. Na, J. Kim, and J.-H. Lee, "Cluster-based patent retrieval," *Inf. Process. Manage.*, vol. 43, pp. 1173–1182, September 2007. [Online]. Available: <http://dx.doi.org/10.1016/j.ipm.2006.11.006>
- [24] Y. Kim, J. Ryu, and S.-H. Myaeng, "A patent retrieval method using semantic annotations," in *KDIR*, A. L. N. Fred, Ed. INSTICC Press, 2009, pp. 211–218. [Online]. Available: <http://dblp.uni-trier.de/db/conf/ic3k/kdir2009.html#KimRM09>
- [25] W. Yi, Y. Sun, S. Zhang, Y. Wu, and Z. Chu, "The application and research of ontology construction technology," in *Knowledge Discovery and Data Mining, 2008. WKDD 2008. First International Workshop on*, jan. 2008, pp. 618–623.
- [26] Y. Xu, "Apply text mining in analysis of patent document," in *Computer-Aided Industrial Design Conceptual Design, 2009. CAID CD 2009. IEEE 10th International Conference on*, nov. 2009, pp. 2350–2352.
- [27] B. Billerbeck, "Efficient query expansion," RMIT University, Melbourne, Victoria, Australia, September, 2005.
- [28] A. S. Fernando and F. Pereira, "Document expansion for speech retrieval," 1999.
- [29] J. Min, J. Leveling, D. Zhou, and G. J. F. Jones, "Document expansion for image retrieval," in *Adaptivity, Personalization and Fusion of Heterogeneous Information*, ser. RIAO '10. Paris, France, France: LE CENTRE DE HAUTES ETUDES INTERNATIONALES D'INFORMATIQUE DOCUMENTAIRE, 2010, pp. 65–71. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1937055.1937070>
- [30] O. Kurland and L. Lee, "Corpus structure, language models, and ad hoc information retrieval," in *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, ser. SIGIR '04. New York, NY, USA: ACM, 2004, pp. 194–201. [Online]. Available: <http://doi.acm.org/10.1145/1008992.1009027>
- [31] X. Liu, "Cluster-based retrieval using language models," in *In Proceedings of SIGIR*. ACM Press, 2004, pp. 186–193.
- [32] K. S. Jones, "Automatic keyword classification for information retrieval," 1971.
- [33] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE*, vol. 41, no. 6, pp. 391–407, 1990.
- [34] Y. Jing and W. B. Croft, "An association thesaurus for information retrieval," in *In RIAO 94 Conference Proceedings*, 1994, pp. 146–160.
- [35] R. Attar and A. S. Fraenkel, "Local feedback in full-text retrieval systems," *J. ACM*, vol. 24, pp. 397–417, July 1977. [Online]. Available: <http://doi.acm.org/10.1145/322017.322021>

- [36] *MESH Ontologies*. [Online]. Available: <http://mklab.itl.gr/content/mesh-ontologies>
- [37] *Unified Medical Language System*. [Online]. Available: <http://www.nlm.nih.gov/research/umls/>
- [38] *WordNet*. [Online]. Available: <http://wordnet.princeton.edu/>
- [39] *Wikipedia*. [Online]. Available: <http://www.wikipedia.org/>
- [40] J. Kamps and M. Koolen, "Is wikipedia link structure different?" in *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, ser. WSDM '09. New York, NY, USA: ACM, 2009, pp. 232–241. [Online]. Available: <http://doi.acm.org/10.1145/1498759.1498831>
- [41] T. Zesch and I. Gurevych, "The more the better? assessing the influence of Wikipedia's growth on semantic relatedness measures," in *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, N. C. C. Chair, K. Choukri, B. Maegaard, J. Mariani, J. Odijk, S. Piperidis, M. Rosner, and D. Tapias, Eds. Valletta, Malta: European Language Resources Association (ELRA), may 2010.
- [42] K. S. Jones, "Idf term weighting and ir research lessons," *Journal of Documentation*, vol. 60, pp. 521–523, 2004.
- [43] A. Arasu, J. Novak, J. Tomlin, and J. Tomlin, "Pagerank computation and the structure of the web: Experiments and algorithms," 2002.
- [44] S. Mohammad, "Measuring semantic distance using distributional profiles of concepts," Ph.D. dissertation, 2008.
- [45] M. Alvarez and S. Lim, "A graph modeling of semantic similarity between words," in *Semantic Computing, 2007. ICSC 2007. International Conference on*, sept. 2007, pp. 355–362.
- [46] H. S. Delugach, "An exploration into semantic distance," in *Proceedings of the 7th Annual Workshop on Conceptual Structures: Theory and Implementation*. London, UK: Springer-Verlag, 1993, pp. 119–124. [Online]. Available: <http://dl.acm.org/citation.cfm?id=645485.656697>
- [47] A. Islam and D. Inkpen, "Semantic text similarity using corpus-based word similarity and string similarity," *ACM Trans. Knowl. Discov. Data*, vol. 2, pp. 10:1–10:25, July 2008. [Online]. Available: <http://doi.acm.org/10.1145/1376815.1376819>
- [48] A. Theobald, "An ontology for domain-oriented semantic similarity search on xml data," in *IN BTW 2003*, 2003, pp. 217–226.
- [49] A. Schwering, "Approaches to semantic similarity measurement for geo-spatial data: A survey," *Transactions in GIS*, vol. 12, no. 1, pp. 5–29, 2008. [Online]. Available: <http://doi.wiley.com/10.1111/j.1467-9671.2008.01084.x>

- [50] J. F. Roddick, K. Hornsby, and D. de Vries, "A unifying semantic distance model for determining the similarity of attribute values," in *Proceedings of the 26th Australasian computer science conference - Volume 16*, ser. ACSC '03. Darlinghurst, Australia, Australia: Australian Computer Society, Inc., 2003, pp. 111–118. [Online]. Available: <http://dl.acm.org/citation.cfm?id=783106.783120>
- [51] D. Bollegala, Y. Matsuo, and M. Ishizuka, "A relational model of semantic similarity between words using automatically extracted lexical pattern clusters from the web."
- [52] V. Tsang and S. Stevenson, "A graph-theoretic framework for semantic distance."
- [53] P. Foltz, *Behavior Research Methods Instruments and Computers*, pp. 197–202.
- [54] R. Thiagarajan, G. Manjunath, M. Stumptner, R. Thiagarajan, G. Manjunath, and M. Stumptner, "M.: Computing semantic similarity using ontologies," Tech. Rep.
- [55] J. Ge and Y. Qiu, "Concept similarity matching based on semantic distance," in *Semantics, Knowledge and Grid, 2008. SKG '08. Fourth International Conference on*, dec. 2008, pp. 380–383.
- [56] G. Varelakis, E. Voutsakis, E. G. M. Petrakis, E. E. Milios, and P. Raftopoulou, "Semantic similarity methods in wordnet and their application to information retrieval on the web," in *In: 7 th ACM Intern. Workshop on Web Information and Data Management (WIDM 2005)*. ACM Press, 2005, pp. 10–16.
- [57] C. Zhang, Y.-J. Wang, B. Cui, and G. Cong, "Semantic similarity based on compact concept ontology," in *Proceeding of the 17th international conference on World Wide Web*, ser. WWW '08. New York, NY, USA: ACM, 2008, pp. 1125–1126. [Online]. Available: <http://doi.acm.org/10.1145/1367497.1367688>
- [58] D. Chen, Y. Jianzhuo, F. Liying, and S. Bin, "Measure semantic distance in wordnet based on directed graph search," in *E-Learning, E-Business, Enterprise Information Systems, and E-Government, 2009. IEEE '09. International Conference on*, dec. 2009, pp. 57–60.
- [59] R. Rada, H. Mili, E. Bicknell, and M. Blettner, "Development and application of a metric on semantic nets," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 19, no. 1, pp. 17–30, jan/feb 1989.
- [60] Z. Wu and M. Palmer, "Verbs semantics and lexical selection," in *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, ser. ACL '94. Stroudsburg, PA, USA: Association for Computational Linguistics, 1994, pp. 133–138. [Online]. Available: <http://dx.doi.org/10.3115/981732.981751>
- [61] Y. Li, Z. Bandar, and D. Mclean, "An approach for measuring semantic similarity between words using multiple information sources," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 15, no. 4, pp. 871–882, july-aug. 2003.
- [62] P. Resnik, "Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language," *Journal of Artificial Intelligence Research*, vol. 11, pp. 95–130, 1999.

- [63] D. Lin, “Principle-based parsing without overgeneration,” 1993.
- [64] J. Jiang and D. Conrath, “Semantic similarity based on corpus statistics and lexical taxonomy,” in *Proc. of the Int’l. Conf. on Research in Computational Linguistics*, 1997, pp. 19–33. [Online]. Available: <http://www.cse.iitb.ac.in/~cs626-449/Papers/WordSimilarity/4.pdf>
- [65] H. Liu, J. Wang, and D. Xu, “A simple and effective concept vector for word-net semantic measurement,” in *Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on*, vol. 2, aug. 2010, pp. V2–342–V2–345.
- [66] G. A. Miller and W. G. Charles, “Contextual correlates of semantic similarity,” *Language & Cognitive Processes*, vol. 6, no. 1, pp. 1–28, 1991. [Online]. Available: <http://eric.ed.gov/ERICWebPortal/recordDetail?accno=EJ431389>
- [67] N. Ide and J. Véronis, “Word sense disambiguation: The state of the art,” *Computational Linguistics*, vol. 24, pp. 1–40, 1998.
- [68] M. Lesk, “Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone,” in *Proceedings of the 5th annual international conference on Systems documentation*, ser. SIGDOC ’86. New York, NY, USA: ACM, 1986, pp. 24–26. [Online]. Available: <http://doi.acm.org/10.1145/318723.318728>
- [69] S. Kulkarni and D. Caragea, “Computation of the semantic relatedness between words using concept clouds.”
- [70] L. Cormen and Rivest, “Greedy algorithms,” in *Introduction to Algorithms*, 1990, pp. 329, Chapter 16.
- [71] *Clef-Ip 2011*. [Online]. Available: <http://www.ir-facility.org/clef-ip>
- [72] *European patent organization (EPO)*. [Online]. Available: <http://www.epo.org/>
- [73] *World Intellectual Patent Organization (WIPO)*. [Online]. Available: <http://www.wipo.int/portal/index.html.en>
- [74] T. Zesch, C. Müller, and I. Gurevych, “Extracting Lexical Semantic Knowledge from Wikipedia and Wiktionary,” in *Proceedings of the Conference on Language Resources and Evaluation (LREC)*, 2008.
- [75] *Java API for WordNet Searching*. [Online]. Available: <http://lyle.smu.edu/~tspell/>
- [76] *Dbpedia*. [Online]. Available: <http://dbpedia.org/>
- [77] *DbpediaSpotlight*. [Online]. Available: <http://dbpedia.org/spotlight>
- [78] P. Mendes, M. Jakob, and A. García-Silva, *DBpedia Spotlight: Shedding Light on the Web of Documents*, 2011. [Online]. Available: <http://www.wiwiss.fu-berlin.de/en/institute/pwo/bizer/research/publications/Mendes-Jakob-GarciaSilva-Bizer-DBpediaSpotlight-ISEM2011.pdf>

- [79] L. Chen, N. Tokuda, and H. Adachi, "A patent document retrieval system addressing both semantic and syntactic properties," in *Proceedings of the ACL-2003 workshop on Patent corpus processing - Volume 20*, ser. PATENT '03. Stroudsburg, PA, USA: Association for Computational Linguistics, 2003, pp. 1–6. [Online]. Available: <http://dx.doi.org/10.3115/1119303.1119304>
- [80] X. Xue and W. B. Croft, "Automatic query generation for patent search," in *Proceeding of the 18th ACM conference on Information and knowledge management*, ser. CIKM '09. New York, NY, USA: ACM, 2009, pp. 2037–2040. [Online]. Available: <http://doi.acm.org/10.1145/1645953.1646295>
- [81] K. Indukuri, A. Ambekar, and A. Sureka, "Similarity analysis of patent claims using natural language processing techniques," in *Conference on Computational Intelligence and Multimedia Applications, 2007. International Conference on*, vol. 4, dec. 2007, pp. 169–175.
- [82] C.-H. Chang and C.-C. Hsu, "Integrating query expansion and conceptual relevance feedback for personalized web information retrieval," *Comput. Netw. ISDN Syst.*, vol. 30, pp. 621–623, April 1998. [Online]. Available: [http://dx.doi.org/10.1016/S0169-7552\(98\)00076-2](http://dx.doi.org/10.1016/S0169-7552(98)00076-2)
- [83] S. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, and M. Gatford, "Okapi at trec-3," 1996, pp. 109–126.
- [84] K. S. Jones, S. Walker, and S. Robertson, "A probabilistic model of information retrieval: Development and status," 1998.
- [85] S. Robertson, S. Walker, M. Beaulieu, and P. Willett, "Okapi at trec-7: Automatic ad hoc, filtering, vlc and interactive track," *In*, vol. 21, pp. 253–264, 1999.

Appendix A

Effects of Expansion on similarity value

In this part, we analyze the effect of expansion on similarity measure. If we have two documents $d1$ and $d2$ with similarity score of s , by using semantic expansion three different situation can happened to s . let's assume after semantic expansion $d1$ transform to $de1$ (in both Document expansion and Query expansion) and $d2$ transform to $de2$ (only in document expansion because in query expansion $de = d$) and s to se .

1. **Same similarity:** Therefore $se = s$
2. **Increase similarity:** Therefore $se \geq s$
3. **Decrease similarity:** Therefore $se \leq s$

To explain different situations, in this section we choose the following documents as an example:

Doc_1 : A, B, C

Doc_2 : A, D, E

A, B, C, D, E, F are representing the words (tokens) inside the documents.

A.1 Same Similarity score after expansion

This situation can happen in two situations. First, if we do not find any expansion for both documents $d1$ and $d2$. And second, if the weights inside document vectors increase uniformly with the same rate (like copy and paste the content of one document inside it). Therefore, in both situations, the content of documents and the relations are not changed after expansion, so the similarity is the same before and after expansion.

A.2 Increase similarity score after expansion

There are 2 different scenarios which lead to increase similarity between two documents after expansion:

1. **Find new similarity in expanded documents:** This means that the content of these two documents have similarity, however, the words are different from each other and after expansion we find a new common word(s), therefore the similarity value must increase. This situation is only valid for document expansion and not query expansion, because in query expansion we just expand the Doc 1, therefore it is not possible to find new similar token in both documents.

Doc₁ : A, B, C, **F**

Doc₂: A, D, E, **F**

2. **Find new similarity between two documents:** Like the previous situation, if the content of two documents are similar without any common words and after expansion we find a common word(s), between expanded tokens of one document and non expanded tokens of the other document therefore the similarity must increase. This situation can happen in both: document expansion and query expansion

Doc₁ : A, B, C, **D**

Doc₂: A, D, E

A.3 Decrease similarity score after expansion

There are 3 different scenarios which lead to decrease similarity between two documents after expansion:

1. **Find new dissimilarity:** This situation can happen in both document expansion and query expansion. This situation is very common in expansion, and most of the noises are produced in this kind of situations. In this situation, tokens are expanded to tokens which are not changing the similarity between two documents.

Doc₁ : A, B, C, **F**

Doc₂: A, D, E

2. **Increase the weight of dissimilar value in one document:** This situation can happen in both document and query expansion

Doc₁ : A, B, C, **B**

Doc₂: A, D, E

3. **Increase the weight of similar value in one document:** This situation can happen in both document expansion and query expansion.

Doc₁ : A, B, C, **A**

Doc₂: A, D, E

In the last situation, by choosing cosine similarity as a similarity measure, it returns a wrong answer. For example assume that in first document three words are used: (Color, Painting and Drawing) and in second document words (Color, Car and Polish)

are used. From the meaning of the words, the main focus of first document is color, however for the second document it is car. Assume that we expand the word, paint and get (paint and color) as an expansion result. This means that the main focus of the first document is about the color, however it is not true for the second one. In this situation similarity between two documents must decrease, however cosine similarity shows that the similarity is increasing. In next Section, we explain the reason. In Table A.3, we show all the situations and the similarity values that cosine similarity calculate with expanded weight $x=0.5$.

Effect	Situation	Vector ₁	Vector ₂	similarity measure	Result for $x=0.5$	Correctness
	Before Expansion	[1,1,1,0,0]	[1,0,0,1,1]	$\frac{1}{3}$	0.33	
Same	Same similarity	[1,1,1,0,0]	[1,0,0,1,1]	$\frac{1}{3}$	0.33	Correct
Increase	New similarity in both documents	[1,1,1,0,0,x]	[1,0,0,1,1,x]	$\frac{1+x^2}{\sqrt{3+x^2}\sqrt{3+x^2}}$	0.38	Correct
	New Similarity between documents	[1,1,1,x,0]	[1,0,0,1,1]	$\frac{1+x}{\sqrt{3+x^2}\sqrt{3}}$	0.48	Correct
	New Dissimilarity	[1,1,1,0,0,x]	[1,0,0,1,1,0]	$\frac{1}{\sqrt{3+x^2}\sqrt{3}}$	0.32	Correct
Decrease	Increase Dissimilar weight	[1,1+x,1,0,0]	[1,0,0,1,1]	$\frac{1}{\sqrt{2+(1+x)^2}\sqrt{3}}$	0.28	Correct
	Increase similar weight of one document	[1+x,1,1,0,0]	[1,0,0,1,1]	$\frac{1+x}{\sqrt{2+(1+x)^2}\sqrt{3}}$	0.42	Not Correct

Table A.1: Effects of expansion on Cosine similarity measure

We answer this problem by two directions, first similarity measure (Cosine similarity) and Model (Weighting problem). In next two sections we discuss it in more details.

A.4 Similarity factors

In previous section, the problem that cosine similarity have to measure the similarity after expansion is explained. To find out the reasons, in this section we discuss different factors that influence the similarity.

1. **Similar features:** If we compare the vectors of two documents with each other we can understand that the similar feature is an important factor. For example, the document with word apple have more similarity to document that have word apple to document that don't have apple. Therefore number of similar features between two documents increases the similarity of two documents. Therefore the number/ weights of similar features can increase or decrease the similarity score.
2. **Dis-similar features:** Like the first factor, dissimilar features in two documents have an influence on the similarity between two documents. For example, the document with words apple and orange has more similarity to the document with words (apple and banana) than document with words (apple and watermelon and banana) Therefore number/weights of dissimilar features between two documents decrease the similarity of two documents
3. **Ratio of Similar features :** However the similar features influence similarity scores, but not only their weights or numbers but also their ratio of appearance

inside the document. For example document with words apple, orange, apple have similarity with document (apple, apple, banana, banana) but it is less than the similarity with document with (apple, banana, and apple). However in both documents the weights and numbers of similar feature (apple) is the same, but ratio of apple in first document is $2/3$, in the second document is $1/2$ and in the last document is $2/3$. Since the ratio of similar features shows the ration of dissimilar features, therefore there is no need to consider the ratio of dissimilar values.

There are more factors that can influence on similarity which can be very different from the above list. The above list is just as example to show the problem of cosine similarity. In cosine similarity, weigh of similar features in both vectors are the main factor which increase the similarity score. Moreover, the size of both vectors in denominator decrease the similarity. Therefore by increasing the size of the vectors similarity score decreasing. However, size of the vectors in denominator is used for normalizing the results. The problem that we discuss in previous section shows that by increasing the weight of similar feature in one vector the similarity increasing, however, since increasing the similar feature in one document, changing the ratio of similarity. Therefore it must be other factors which influence on similarity. Such as using distance between similar ratios as a factor. Finding a new similarity measure for expanding document is out of the scope of this study, therefore we just mention it for future work.

A.5 Retrieval Model

Another reason, that increases the weight(s) of similar value(s) in one document lead to increase the similarity value in cosine similarity after expansion, can be the model. The model that we choose in this study is vector space model and we choose TF-IDF for weighting the words. As we discussed in previous section, in cosine similarity increase the value of similar features leads to increase the similarity, therefore, if we choose another way of weighting the words inside document and include the ration of similar features inside the model, the problem can be solved. Therefore, we leave this part as our future work to find a suitable model for document and query expansion via vector space model and cosine similarity as a similarity measure.

Appendix B

System Features

We use two knowledge sources for expansion in our system, Wikipedia and WordNet. Each of them have their own features and user can choose one of them or both of them for expansion. Moreover, we use 4 different data collections in our system, therefore user have to choose between them at pre-processing phase for constructing an index system, either normal or semantic. In this part, we describe these features in more detail.

B.1 WordNet

1. Dataset Setting
* Please choose one of the following section of patent document for indexing system.

☐ Title ☐ Abstract ☐ Claim ☐ All

2. Sematic Indexing Setting

☒ WordNet ☐ Wikipedia ☐ WordNet + Wikipedia

☐ Use Stemming

	Depth	Weight
☐ Hypernymy		
☐ Hyponymy		
☐ Both		

☐ Use DBpedia Spotlight

Expansion size Threshold

☐ Ingoing Links ☐ Outgoing Links ☐ Both

Depth Weight

Figure B.1: WordNet setting

To expand via Wordnet, user can choose the following options:

1. **kind of the link:** which can be only hyponymy links or only hyponymy links or both of them
2. **Depth:** which shows the distance from the original node in the graph
3. **Weight:** which shows the weight that user give to expanded tokens. The realtion between weight and depth is explained in Chapter 3

4. **Stemming**: user also can choose to use stemming or not.

B.2 Wikipedia

Figure B.2: Wikipedia setting

To expand via Wikipedia, user can choose the following options:

1. **kind of the link**: which can be only ingoing links or only outgoing links or both of them
2. **Depth**: which shows the distance from the original node in the graph
3. **Weight**: which shows the weight that user give to expanded tokens. The relation between weight and depth is explained in Chapter 3
4. **Stemming**: user also can choose to use stemming or not.
5. **expansion size**: user can choose the size of expansion by choosing the number for expansion size.
6. **threshold**: Threshold means the number of ingoing links. By using this option user can choose to use general pages with more links or specific and new pages with less ingoing links, however, this option is also influence on size of expansion. Therefore, more ingoing links means less expansion.
7. **DBpedia spotlight**: since we use this tool for WSD, user can choose to not use this tool.

B.3 Dataset

The user can choose between different data collections. We add this option to let us query based on different parts of one patent document. User can choose between four options: (1) Title, (2) Abstract, (3) Claim and (4) All, which is the combination of Title, abstract, claim and description.

1. Dataset Setting
 * Please choose one of the following section of patent document for indexing system.

☐ Title ☐ Abstract ☐ Claim ☒ All

2. Semantic Indexing Setting

☐ WordNet ☐ Wikipedia ☒ WordNet + Wikipedia

☐ Use Stemming ☐ Use DBpedia Spotlight

Expansion size Threshold

Depth Weight

☐ Hypernymy ☐ Ingoing Links ☐ Both

☐ Hyponymy ☐ Outgoing Links ☐ Both

☐ Both

Figure B.3: Dataset setting

B.4 indexing

1. Dataset Setting
 * Please choose one of the following section of patent document for indexing system.

☐ Title ☐ Abstract ☐ Claim ☒ All

2. Semantic Indexing Setting

☐ WordNet ☐ Wikipedia ☒ WordNet + Wikipedia

☐ Use Stemming ☐ Use DBpedia Spotlight

Expansion size Threshold

Depth Weight

☐ Hypernymy ☐ Ingoing Links ☐ Both

☐ Hyponymy ☐ Outgoing Links ☐ Both

☐ Both

Figure B.4: Indexing setting

In pre-processing phase the user can choose between indexing systems, normal for normal and query expansion search or semantic for document expansion search. User can also choose to use both of them.

Appendix C

EPO Stop Words

In this appendix we list all the stop words for patent document from EPO: Here in Table C.1 there is a list of all English stop words:

a, also, Also, an, An, and, And, any, Any, are, Are, as, at, At
be, been, Been, being, Being, by, By
could, Could
Dwg
each, Each, etc, Etc, example, examples
for, For, from, From
had, Had, has, Has, have, Have, having, Having
if, If, in, into, Into, invention, inventions, is, Is, it, It, its, Its
may
no, No, not, Not
of, Of, on, On, onto, Onto, or, Or
SEP
should, Should, so, So, some, Some, such, Such
tb
that, That, the, The, THE, their, Their, then, Then, there, There, thereby, Thereby,
therefore, Therefore, therein, Therein, thereof, Thereof, these, These, this, This, those,
Those, thus, Thus, to, To
was, Was, were, Were, when, When, where, Where, wherein, Wherein, which,
Which, while, While, with, With, would, Would

Table C.1: English Stop Words

In Table C.2 there is a list of all Patent stop words.

Finally, all the stop words that we use in our system without considering the shape of the words are in Table C.3.

a, about, About, ABOUT, accordance, Accordance, ACCORDANCE, according, According, ACCORDING, along, Along, ALONG, also, Also, ALSO, an, An , AN, and, And, any, Any, ANY, are, Are, ARE, as, at, At, AT
be, because, Because, BECAUSE, become, Become, BECOME, becomes, Becomes, BECOMES, been, Been, BEEN, being, Being, BEING, between, Between, BETWEEN, brought, Brought, BROUGHT, but, But, BUT, by, By, BY
claim, Claim, CLAIM, composed, Composed, COMPOSED, comprise, Comprise, COMPRISE, comprises, Comprises, COMPRISES, comprised, Comprised, COMPRISED, comprising, Comprising, COMPRISING, consisting, Consisting, CONSISTING, consists, Consists, CONSISTS, could, Could, COULD
described, Described, DESCRIBED, desired, Desired, DESIRED, do, Do, DO, does, Does, DOES, due, Due, DUE, during, During, DURING
each, Each, EACH, effective, Effective, EFFECTIVE, embodiment, Embodiment, EMBODIMENT, etc, Etc, ETC, example, Example, EXAMPLE, excellent, Excellent, EXCELLENT
for, For, FOR, from, From, FROM
give, Give, GIVE, given, Given, GIVEN, good, Good, GOOD
had, Had, HAD, has, Has, HAS, have, Have, HAVE, having, Having, HAVING, here, Here, HERE, hereby, Hereby, HEREBY, herein, Herein, HEREIN, hereupon, Hereupon, HEREUPON, however, However, HOWEVER
if, If, in, into, Into, INTO, invention, Invention, INVENTION, is, Is, IS, it, It, IT, its, Its, ITS, just, Just, JUST, k, K, Kk, KK, like, Like, LIKE
made, Made, MADE, make, Make, MAKE, making, Making, MAKING, may, May, MAY, meanwhile, Meanwhile, MEANWHILE, more, More, MORE
no, No, NO, not, Not, NOT, now, Now, NOW
obtain, Obtain, OBTAIN, obtained, Obtained, OBTAINED, obtaining, Obtaining, OBTAINING, obtd, Obtd, OBTD, of, Of, OF, on, On, ON, onto, Onto, ONTO, or, Or, otherwise, Otherwise, OTHERWISE
part, Part, PART, parts, Parts, PARTS, possible, Possible, POSSIBLE, prescribed, Prescribed, PRESCRIBED, provide, Provide, PROVIDE, provided, Provided, PROVIDED, provides, Provides, PROVIDES, providing, Providing, PROVIDING, purpose, Purpose, PURPOSE
relatively, Relatively, RELATIVELY, required, Required, REQUIRED, respective, Respective, RESPECTIVE
said, Said, SAID, should, Should, SHOULD, since, Since, SINCE, so, So, SO, some, Some, SOME, such, Such, SUCH, suitable, Suitable, SUITABLE, suitably, Suitably, SUITABLY
than, Than, THAN, that, That, THAT, the, The, THE, their, Their, THEIR, them, Them, THEM, then, Then, THEN, thence, Thence, THENCE, there, There, THERE, thereafter, Thereafter, THEREAFTER, thereby, Thereby, THEREBY, therefore, Therefore, THEREFORE, therein, Therein, THEREIN, thereof, Thereof, THEREOF, thereon, Thereon, THEREON, thereto, Thereto, THERETO, therewith, Therewith, THEREWITH, these, These, THESE, they, They, THEY, this, This, THIS, those, Those, THOSE, through, Through, THROUGHT, thus, Thus, THUS, titled, Titled, TITLED, to, To, TO
until, Until, UNTIL, use, Use, USE, used, Used, USED, useful, Useful, USEFUL, using, Using, USING
was, Was, WAS, were, Were, WERE, what, What, WHAT, when, When, WHEN, where, Where, WHERE, whereby, Whereby, WHEREBY, wherein, Wherein, WHEREIN, whether, Whether, WHETHER, which, Which, WHICH, while, While, WHILE, who, Who, WHO, whose, Whose, WHOSE, will, Will, WILL, with, With, WITH, within, Within, WITHIN, would, Would, WOULD

Table C.2: Patent Stop Words

a , also , an , and , any , are , as , at , be , been , being , by , could , dwg , each , etc , example , examples , for , from , had , has , have , having , if , in , into , invention , inventions , is , it , its , may , no , not , of , on , onto , or , sep , should , so , some , such , tb , that , the , their , then , there , thereby , therefore , therein , thereof , these , this , those , thus , to , was , were , when , where , wherein , which , while , with , would , compare , see , about , accordance , according , along , because , become , becomes , between , brought , but , claim , composed , comprise , comprises , comprised , comprising , consisting , consists , described , desired , do , does , due , during , effective , excellent , give , given , good , here , hereby , herein , hereupon , however , just , kk , like , made , make , making , meanwhile , more , now , obtain , obtained , obtaining , obtd , otherwise , part , parts , possible , prescribed , provide , provided , provides , providing , purpose , relatively , required , respective , said , since , suitable , suitably , than , them , thence , thereafter , thereon , thereto , therewith , they , titled , until , use , used , useful , using , what , whereby , whether , who , whose , will , within , embodiment , trough , through , thought

Table C.3: List of all Patent Stop Words