SEMANTIC SEARCH ON TURKISH NEWS DOMAIN WITH AUTOMATIC QUERY EXPANSION

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

TUĞBA DEMİR

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

JULY 2016

Approval of the thesis:

SEMANTIC SEARCH ON TURKISH NEWS DOMAIN WITH AUTOMATIC QUERY EXPANSION

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

Prof. Dr. Gülbin Dural Ünver _____
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Adnan Yazıcı _____
Head of Department, **Computer Engineering**

Prof. Dr. Nihan Kesim Çiçekli _____
Supervisor, **Computer Engineering Dept., METU**

Examining Committee Members:

Prof. Dr. Ali Doğru
Computer Engineering Dept., METU _____

Prof. Dr. Nihan Kesim Çiçekli
Computer Engineering Dept., METU _____

Assist. Prof. Dr. Selim Temizer
Computer Engineering Dept., METU _____

Assist. Prof. Dr. Çiğdem Turhan
Computer Engineering Dept., Atılım University _____

Assist. Prof. Dr. Gönenç Ercan
Computer Engineering Dept., Hacettepe University _____

Date: _____

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

Name, Last Name: TUĞBA DEMİR

Signature :

ABSTRACT

SEMANTIC SEARCH ON TURKISH NEWS DOMAIN WITH AUTOMATIC QUERY EXPANSION

DEMİR, Tuğba

M.S., Department of Computer Engineering

Supervisor : Prof. Dr. Nihan Kesim Çiçekli

July 2016, 52 pages

In this thesis, semantic search on Turkish news domain with query expansion is proposed. Our aim is to provide the user with the most relevant documents related to their entered keywords. Our system uses data sources from Turkish news websites such as Hürriyet, Milliyet, Sabah, etc. Our system extends the user's query with word embeddings and semantic relatedness. Furthermore, named entities, containing precious information, are extracted from news sources and user query and ranked to return on top of the results. In the rest of search process, it relies on traditional information retrieval (IR) techniques. For Turkish language, to the best of our knowledge, our system is the first attempt to use such search and query extension techniques on news data.

Keywords: Information retrieval, semantic search, query expansion, keyword based search

ÖZ

OTOMATİK SORGU GENİŞLETMESİ İLE TÜRKÇE HABER İÇİN SEMANTİK ARAMA

DEMİR, Tuğba
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi : Prof. Dr. Nihan Kesim Çiçekli

Temmuz 2016, 52 sayfa

Bu tezde sorgu genişletme ile Türkçe haber alanında yapılan anlamsal aramalar sunulmaktadır. Hedefimiz kullanıcıya arama yaptıkları anahtar kelimeler ile en ilgili haberleri sunmaktır. Sistemimiz Hürriyet, Milliyet, Sabah, vb. haber sitelerinin veri kaynaklarını kullanmaktadır. Sistemimiz, kullanıcının sorgusunu kelime gömme ve anlamsal ilişkilendirme tekniklerini kullanarak genişletmektedir. Dahası, önemli bilgiler barındıran isimsel varlıklar, haber kaynaklarından ve kullanıcı sorgularından çıkarılmakta ve bu isimsel varlıklar, haberlerin en üstünde döndürülmek üzere sıralanmaktadır. Arama işleminin devamında ise geleneksel bilgi çıkarma (IR) tekniklerine bağlı kalınmaktadır. Türkçe dili için, bilinen kadarıyla, sistemimiz haber verileri üzerinde sorgu genişletme ve arama tekniklerini kullanan ilk girişimdir.

Anahtar Kelimeler: Bilgi çıkarımı, anlamsal arama, sorgu genişletme, anahtar kelime tabanlı arama

To my family...

ACKNOWLEDGMENTS

I would like to express my deep appreciation to my supervisor Prof. Dr. Nihan Kesim Çiçekli for her guidance, encouragement and for her valuable advices throughout this study. It was really a great chance for me to work such a thoughtful, friendly and motivating supervisor.

I would like to deeply thank to every member of my family, especially to my mother Hatice Demir and to my father Ahmet Demir for their supports.

I am thankful to my co-workers Fırat Erciş, Dilek Baysal and Fahriye Özge Ünel for their encouragement and friendships.

Finally, I wish to express my warmest thanks to my husband İlker Fındık for his endless love, support and patience. Without his love, this work would be harder for me.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

ANNIE	A Nearly-New Information Extraction System
API	Application Programming Interface
BORO	Business Object Reference Ontology
CBOW	Continuous Bag of Words
CRF	Conditional Random Field
CRM	Conceptual Reference Model
DIP	Data Information and Process Integration
GATE	General Architecture for Text Engineering
GUI	Graphical User Interface
HCI	Human Computer Interaction
HTML	HyperText Markup Language
IPTC	International Press Telecommunications Council
IR	Information Retrieval
JAPE	Java Annotation Patterns Engine
KB	Knowledge Base
LSI	Latent Semantic Indexing
NE	Named Entity
NER	Named Entity Recognition
NNLM	Neural Net Language Model
OWL	The Web Ontology Language
POS	Part of Speech
RDF	Resource Description Framework
RSS	Rich Site Summary
TOVE	Toronto Virtual Enterprise
t-SNE	t-distributed Stochastic Neighbor Embedding
UI	User Interface
VSM	Vector Space Model
WNF	World News Finder

WNO	World News Ontology
XML	Extensible Markup Language

CHAPTER 1

INTRODUCTION

1.1 Motivation

Search engines provide the user a quick way of reaching the information. People use search engines for many purposes such as entertainment, shopping, information gathering, etc. Most people prefer search engines like Google [1], which is a keyword-based Internet search engine, trying to cover all the web. When it comes to searching for a specific information belonging to a particular domain, search engines have some drawbacks. Their current states are updated irregularly in order to list the most popular information on the top of the results, and when it comes to detailed queries, they fail to provide high precisions.

When we want the information on a certain field, domain-specific search engines can be more powerful than web search engines. For Turkish news domain, there are a limited number of search engines covering all the news. One of them is Google news Turkey [2], which is a keyword-based news search engine. It provides a basic search using the exact words entered by user. Another option for searching news is using the search boxes provided in the news websites. People can use these boxes for searching for their keywords. However, user must apply a search on each website, which is time consuming. An important point in the news search is that the search engines must be user-friendly and have a broad coverage of information related to the user's search.

Our main motivation on this study is to design a search system for Turkish news which not only covers a broad range of the contents of Turkish news websites, but also provides the user with the most relevant results by using semantic enhancement techniques and a user-friendly interface.

1.2 Problem Definition and Our Approach

In basic keyword search, search engines mostly use a ranking system that measures the number of matching terms between the words in query and the words on a page. In the case of a page not containing one or more words specified by the user, the page is not ranked at all. On the other hand, semantic search tries to understand the user's intent to improve search accuracy. It extends the user's query with semantically related words. Even if a page does not contain exact words specified by the user, it can be in ranked results. Semantic search is also very helpful in Turkish news domain. In order to construct semantic search system on Turkish news domain the following issues should be dealt with: the extraction of knowledge from various news sources, extending the knowledge by using semantic web techniques and providing user with a user-friendly interface.

In the news domain, there exist a huge amount of news sources should be organized automatically for searching and data retrieval purposes. Although webpage metadata facilitates gathering of news, people can have difficulties in managing the metadata and mining useful information. Current metadata of news are heterogeneous and does not cover all the knowledge on these documents. Manual annotation is not practical and for Turkish news texts, there is no automatic annotation tool which is fully developed. To retrieve the latest contents from news websites easily, Rich Site Summary (RSS) feeds are created. Articles are collected daily using these RSS feeds provided by Turkish news websites. A Named entity recognition (NER) algorithm is applied on the collected articles. The named entities contain precious information like person names, locations and organizations. These named entities are used to improve the precision of a search.

We extend the user queries with query expansion techniques in order to improve the retrieval performance of our search system. In this way, more information that is related to user queries are returned. We use two methods for the query extension: a Neural Network Model, namely word2vec [3] and Wordnet for Turkish [4]. The Neural Network Model and the Wordnet produce semantically related words to user queries.

A user interface (UI) is created to present the relevant information to the user. It consists of a search box, a date picker, and the result part. For the sake of simplicity, in the result screen, we give the title and a brief description of news with a link to the origin of the news.

Towards our goal, we evaluated our system to show the effects of named entities and query expansion on the performance of the search.

1.3 Contributions

The main contributions of this thesis can be summarized as follows:

1. A keyword-based semantic search engine for Turkish news domain is proposed and implemented. The implemented components can be used to support other domains such as biographies, arts, etc.

2. A query expansion module is implemented by using a Neural Network Model and Wordnet on Turkish news text. A NER algorithm is used on our collected data to extract named entities which contain precious information for search.

3. Our system is evaluated by comparing the results with manually annotated test data, which contains about 600 articles and improvements on classical keyword-based approaches are achieved by applying NER and query extension techniques.

1.4 Organization of Thesis

The rest of the thesis is organized as follows. In chapter 2, a brief background information is given about the techniques used in this thesis. In chapter 3, the works related to our system are presented. In chapter 4, the details of the components in our system are given and implementation is explained. Firstly, data processor module is explained, which contains keyword extraction (NER), stemming, stop-word elimination and storage processes. Then indexing details are given. Chapter 4 continues with the query expansion details. Lastly, ranking and GUI implementations are explained. Evaluation results are presented in Chapter 5. Finally, in chapter 6, the thesis is concluded with a brief summary, discussions and possible future work.

CHAPTER 2

BACKGROUND INFORMATION

In this chapter, we first give general information about the existing query expansion techniques. Then, we explain the named entity recognizer used in our search system.

2.1 Query Expansion Techniques

The word mismatch problem is one of the most fundamental problems in information retrieval. People often tend to use different words while describing the contents of their queries instead of the words described by the authors for the same query. That is why query expansion techniques are proposed for dealing with this problem. Query expansion extends the user's query with the relevant words in order to increase the relevancy on searches. There are many different approaches for query expansion like interactive query expansion [5], relevance feedback [6], word sense disambiguation [7], search results clustering [8], and boolean term decomposition [9]. In this chapter, a brief information about the most popular query expansion techniques is given and the expansion techniques used in our system are explained.

2.1.1 Query expansion using Relevance Feedback

Rocchio [6] introduced "relevance feedback" technique to expand a query. In the relevance feedback technique, vector representations for documents and requests are constructed. Cosine similarity measures between documents and requests are used to find appropriate matches. Users are then asked to specify which documents are related to their query and this information is used to construct an optimal query, which would give the most relevant output to user's queries. Classic relevance feedback implementations are based on Rocchio's work [6].The process of relevance feedback is as follows:

1. User enters a query to search.
2. Initial set of retrieved results are returned to user.
3. User marks returned documents as relevant or irrelevant.
4. Using the user's feedback, a new query is constructed and pushed to the system.
5. A revised set of retrieval results are returned to the user.

Relevance feedback is not sufficient in the case of misspelled word or for when the user's word does not match the vocabulary of the collection.

2.1.2 Query Expansion using WordNet

WordNet [10] is a lexical database containing word definitions and their relationships with other words. WordNet consists of a set of synonyms (synsets). All sysnsets have semantic relations with other sysnsets. These relations for nouns include hypernym, hyponym, meronym and holonym relations. One of the early works using WordNet for query expansion is conducted by Voorhees [11]. Voorhees used semantic relations in WordNet to expand queries. A hand-selected list of sysnsets related to a particular topic is created to handle the situation where a query word occurs in multiple synsets like in "spider" word which can be an animal or a computer term. Then useful sysnsets are added to the query.

For Balkan languages, Balkanet project is created. The Balkan WordNet aims at the development of a multilingual lexical database comprising of individual WordNets for the Balkan languages [12]. For Turkish language, there exists Turkish WordNet developed in the scope of Balkanet project [4]. Turkish WordNet is distributed in the XML file format. The file consists of a series of synsets, each between <SYNSET> </SYNSET> tags. Each tag contains a single synset and its relation with other synsets. Fig 2.1 shows an example synset containing the synonymous words "adet" and "sayı".

```
<SYNSET>
        <ID>ENG20-04836174-n</ID>
        <POS>n</POS>
        <SYNONYM>
                <LITERAL>
                        adet
                        <SENSE>2</SENSE>
                </LITERAL>
                <LITERAL>
                        sayı
                        <SENSE>2</SENSE>
                </LITERAL>
        </SYNONYM>
        <ILR>
                ENG20-04826612-n
                <TYPE>hypernym</TYPE>
        </ILR>
        <DEF>Sayma, ölçme, tartma gibi işlerin sonunda bulunan birimlerin kaç olduğunu anlatan söz</DEF>
        <BCS>1</BCS>
</SYNSET>
```

Figure 2.1: A screen shot from Turkish WordNet XML

< **ID** > : unique synset identifier that links Turkish WordNet to all other WordNets built upon Princeton WordNet. ID beginning with ENG like ENG20-07172978-n means that this synset was taken from Princeton WordNet 2.0, has the Princeton WordNet 2.0 ID 07172978 and is a noun. Similarly, ID beginning with BILI like BILI-60000008 means that this synset is one of the Balkanet-specific concepts jointly developed by the six members of the Balkanet Consortium and has the Balkanet Inter-Lingual Index (BILI) identifier 60000008.

< **POS** > : contains part-of-speech information, where 'n' denotes a noun, 'v' a verb and 'a' an adjective.

< **SYNONYM** > : contains the synset members within <literal> tags and their automatically assigned sense numbers within < SENSE > tags.

$<$ **ILR** $>$: links the synset to another synset, where the $<$ TYPE $>$ tag indicates the type of the semantic relation. It can be hypernym, hyponym, near-antonym and holo-member. When we follow the given id ENG20-04826612-n in this specific example, we get "miktar" as hypernym of "adet".

$<$ **DEF** $>$: contains a brief definition of the concept.

$<$ **BCS** $>$: indicates whether the synset belongs to one of the three Base Concept sets defined during the Balkanet project [13].

Statistics about the number of synsets and word senses are given in Table 2.1

Table2.1: Statics about words in XML

PoS	synsets	word senses	words
Nouns	11226	15253	12443
Verbs	2736	3769	2232
Adjectives	792	1318	1086
Adverbs	40	79	71

We used Wordnet for Turkish in our IR system to find synonyms of entered query words.

2.1.3 Query expansion using ontologies

Ontology provides a vocabulary of terms which contains knowledge about some topic. The vocabulary provided by ontology also contains relationships between the terms. Resource Description Framework (RDF) [14] and the Web Ontology Language (OWL) [15] are the languages used in the construction of ontologies. Using ontologies for query expansion is one of the latest trends in the semantic search. Corese [16] is an ontology-based search engine which uses RDF(S) semantic metadata and uses the query language SPARQL based on RDF(S). For a user query, Corese produces approximate answers by computing semantic distance of classes or properties in the ontology hierarchies.

For an example search "A book written by a teacher" which is taken from Corese manual [17], Corese can produce "an article written by a researcher" as an approximate answer. For the given example, simple ontology, a query example and search results are shown in Fig 2.2 below.

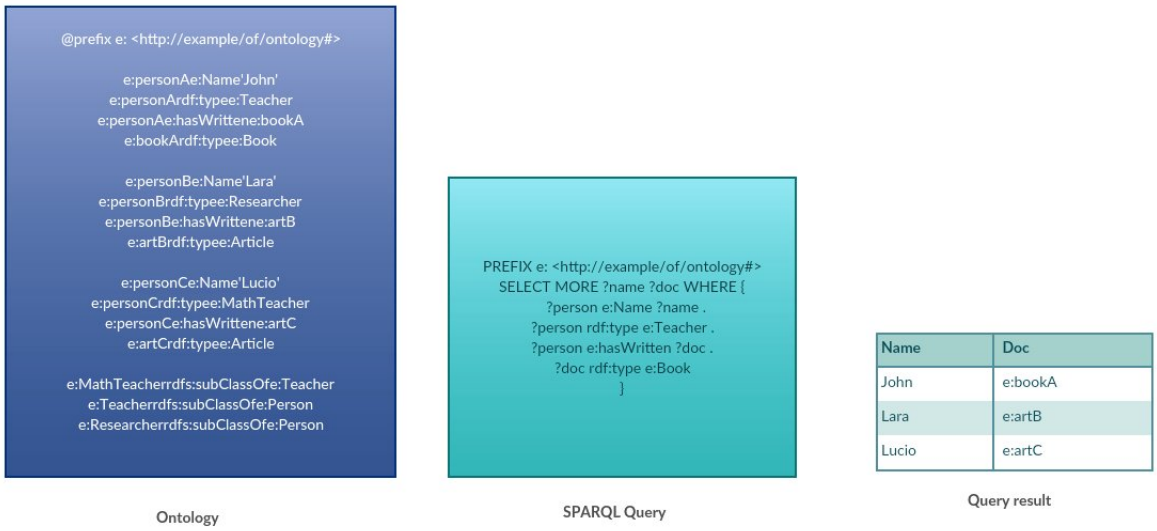

Figure 2.2: An example query process in the Corese search system

Another approach using approximation is presented in [18]. They propose a method for querying RDF datasets with SPARQL using Bloom filters to produce approximate answers to the user's query. There exist domain-specific approaches in the construction of ontology. In a domain-specific ontology, terms and concepts from a given domain are modeled. These ontologies have many different application areas including law, medicine, geography, business, history, news, etc. Some of the usage examples of domain-specific approaches are explained in [19], [20], [21] and [22]. We did not use an ontology in our search system.

2.1.4 Query expansion using distributed word representations

In recent years, neural network technologies are successfully integrated with word representation models. Mikolov introduced Word2Vec [23] which is a "shallow" neural network model that can process billions of word occurrences and create semantically and syntactically meaningful word representations. Word2Vec can be used for query expansion. To do so, firstly, Word2Vec is trained with a corpus to produce distributional word representations. The words are represented as multi-dimensional vectors. During the modeling, there is no need for human supervision. Only the context surrounding the words is considered. In multi-dimensional vector space, semantically related words are close to each other. Then, by using the proximity between two vectors, semantic similarity between user query and other words in vector space are found. Word2Vec returns semantically related words to user's query with their distance to the query. A threshold can be chosen such that the words with the distance above this threshold are used as extended words to the user's query. In [24] and [25], Word2Vec is used for query expansion. We also used Word2Vec for query expansion in our search system.

2.2 Turkish Named Entity Recognition

Named Entity Recognition (NER) is an important process in Information Extraction. With the help of NER, named entities such as persons, locations, organizations can be extracted from an unstructured text. There are different works conducted for Turkish NER. One of them is NER proposed by Bayraktar and Temizel [26]. They extract the person names from financial news text using a local grammar-based approach and they achieved 81.97 % performance on news data as they reported. Performance measure is F-Score, which is harmonic average between accuracy and coverage. Moreover, Küçük and Yazıcı [27] followed a rule-based approach with rote learning algorithm and achieved 90.13 % for NER on Turkish news articles.

Tatar and Çiçekli [28] proposed an automatic rule learning system and they achieved 91.08 % performance. There are two other approaches using conditional random fields (CRF) method for NER tasks in Turkish. One of them is conducted by Yeniterzi [29] and achieved 88.94 % performance. The other is the work of Şeker and Eryiğit [30] by using CRF morphological features and gazetteer lists on Turkish news data with the performance of 92 %.

We followed the approach of Tatar and Çiçekli [28]. Thanks to them for providing us the annotated Turkish corpus, which contains 5672 named entities with 1335 person names, 2355 location names and 1218 organization names. They also provided us an API for keyword extraction as a .jar executable. We added this library to our project and used it in keyword extraction.

CHAPTER 3

RELATED WORK

This chapter surveys the previous work on semantic search and query expansion techniques. For each work, we give the methods and how they are related to our search system.

Artequekt project [22] automatically produces biographies of artists from fragments of information extracted from the Web. The creators of Artequekt constructed the Conceptual Reference Model (CRM) ontology [31] to represent artifacts, their production, ownership, location, etc. In their system, information extraction, knowledge storage and narrative generation are applied in order. Information extraction process is done by using ontology and WordNet [10]. Crawled web documents are split into paragraphs. On these paragraphs, semantic and syntactic analyses are done. They use NER component of GATE (General Architecture for Text Engineering) framework [32] during the semantic analysis to find person names. In syntactic analysis, they divide sentences into one or more clauses, each containing a subject, verb and an object. After analyses are done, they find relations between clauses by using WordNet synonyms, hypernyms and hyponyms.

For an example sentence, extracted knowledge using ontology and WordNet is shown in Fig 3.1 (Alani et al.,2003,p.3)below.

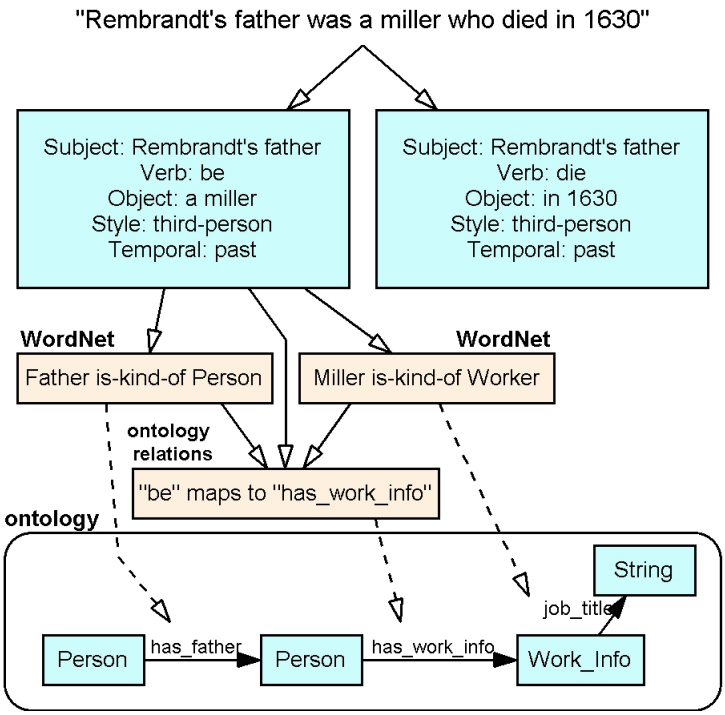

Figure 3.1: An example of knowledge extraction using ontology and WordNet

These extracted information are stored in a knowledge base (KB) in XML format. This XML file is sent to an ontology server to extend the knowledge for creating the relevant instances and relationships in the ontology. For the user's query, server renders a narrative from the information stored in the knowledge base and returns a story.In our search system, a Turkish WordNet is used to find synonyms and a Turkish NER is used for keyword extraction as in the Artequekt project.

Neptuno project [20] is one of the earliest semantic search systems developed to make semantic search on news domain. Their data was coming from a news media company (Diari SEGRE) archive. Reporters and archivists were annotating news documents. Neptuno creates an ontology for news, which has been created by using existing ontologies and vocabularies.

IPTC consortium standards [33] are also used to classify news archive contents and are integrated with their ontology. A knowledge base is constructed to hold archive materials which are described using the ontology. User is provided with a user interface (UI), where s/he selects the classes of content (news, photograph, graphics, or page) and enters keywords for the desired fields (heading, author, section, date, subject, etc.). These information is used to construct RDQL query [34], which is an RDF query language. Constructed query is run on KB and relevant results to user's query are returned. We did not use an ontology in our search system. The Neptuno project and our work only have domain "news" in common.

World News Finder (WNF) [35] performs a semantic search on news domain by using automatically extracted metadata files. System applies search on metadata files rather than on keywords. Daily articles are collected by using RSS (Rich Site Summary) news feeds provided by news agencies. With an HTML parser, news feeds are parsed and each article is stored as a plain text. All the irrelevant contents (advertisements, media content, presentation elements) are removed from the articles before giving to ANNIE (A Nearly-New Information Extraction System) pipeline, which is a GATE component [32]. ANNIE performs syntactical parsing, Named Entity Recognition and pattern matching with the following components: English tokenizer, onto gazetteer, sentence splitter, minipar parser, pos tagger , named entity (NE) transducer, orthomatcher and JAPE transducer.Each plain text document is given ANNIE as input and the annotated documents are taken as the output. In WNF, initial annotations are enriched with information World News Ontology (WNO) created. Then the annotated documents are given to News Meta Tagger which is also a GATE component. News Meta Tagger extracts useful metadata and puts in a file called "hat". Each hat contains the annotations of topics with the number of occurrences, persons and locations.

An example hat can be seen in Fig 3.2.

```
<LOCATION>
        <CONTINENT>Middle East</CONTINENT>
        <COUNTRY>IRAQ</COUNTRY>
        <REGION-CITY>BAGHDAD</REGION-CITY>
</LOCATION>
<LOCATION_ABOUT>
        <CONTINENT>Americas</CONTINENT>
        <COUNTRY>U.S.A.</COUNTRY>
</LOCATION_ABOUT>
<DATE>20090503</DATE>
<TOPIC>
        <NAME>troops_withdrawal</NAME>
        <WEIGHT>5</WEIGHT>
        <ATTR>
                <TYPE>home_country</TYPE>
                <VALUE>U.S.A.</VALUE>
        </ATTR>
</TOPIC>
<TOPIC>
        <NAME>armed_forces</NAME>
        <WEIGHT>5</WEIGHT>
        <ATTR>
                <TYPE>home_country</TYPE>
                <VALUE>U.S.A.</VALUE>
        </ATTR>
</TOPIC>
<TOPIC>
        <NAME>war</NAME>
        <WEIGHT>3</WEIGHT>
</TOPIC>
<TOPIC>
        <NAME>security_measures</NAME>
        <WEIGHT>1</WEIGHT>
</TOPIC>
<TOPIC>
        <NAME>act_of_terror</NAME>
        <WEIGHT>1</WEIGHT>
</TOPIC>
```

Figure 3.2: A screen shot from XML containing a hat

WNF provides the user an API for searching. Location, date range, subject and topic fields are used to construct a query. The query is searched on the metadata hats and relevant results is shown to the user. WNF is evaluated by posting 400 queries to the search system. Evaluation of results is done manually by direct examination of the database content. They achieved 97.56 % precision and 92.33 % recall on the average. WNF has much in common with our work. We search on news as well. In the collection of articles, we use RSS news feeds and remove irrelevant content as in WNF. We use a Turkish NER for named entity recognition while WNF uses a GATE component.

Lupiani-Ruiz et al. [21] proposed a semantic search system for financial news based on Semantic Web technologies. Their search system consists of the following three modules: financial ontology, ontology population and ontology-based search engine module. They used Ontology Web Language (OWL) [15] to show the extracted knowledge obtained from the text. They constructed

a financial ontology based on existing ontologies like TOVE (Toronto Virtual Enterprise) [36], BORO (Business Object Reference Ontology) [37], financial ontology of DIP (Data Information and Process Integration) consortium [38], etc. Ontology population module populates financial ontology with the relevant information. Their ontology-based search engine module makes semantic annotation, querying and searching. In semantic annotation process, they annotate RSS news and Web page sources with the domain ontology by using GATE. In query processing, they apply Part-of-speech (POS) tagging, lemmatization, NER on user query. User query is expanded by synonyms and given as input to ontology-based search engine module. Search engine module searches annotations related to expanded query on OWL based financial ontology and shows the results to user in ranked order. Query processing approach in this work is similar to ours. POS- tagging, lemmatization and NER are applied in Turkish news search system. We extend user queries with word synonyms.

Another search system most similar to ours is conducted by Nikitinsky et al [24]. They created technology analysis and forecasting information retrieval system for Russian language. Their data consists of patents, research papers and government contracts. In addition to classical full-text search methods, they apply query extension methods by using semantic analysis and word embeddings. Their system pipeline is shown in Fig 3.3 below.

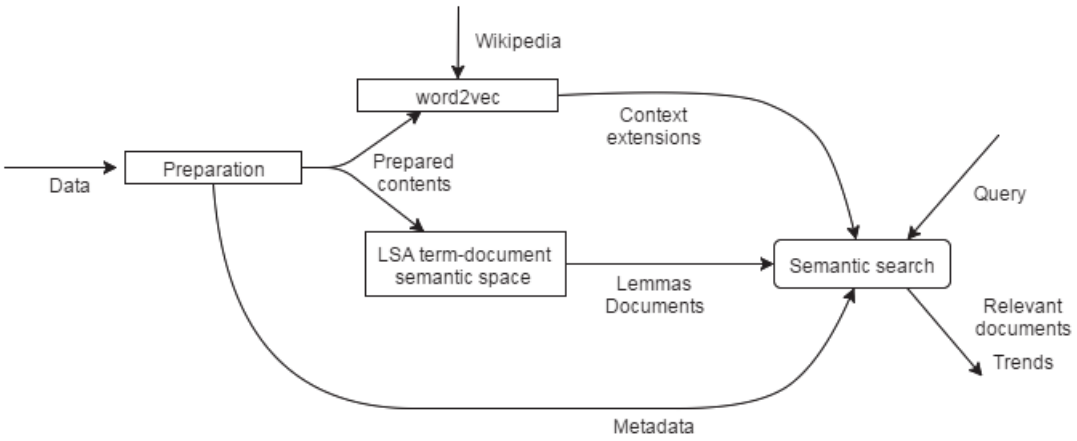

Figure 3.3: System pipeline of IR for Russian

They extract the metadata of documents for setting filters to search. Then tokenization, stop-word elimination and lemmatization are done on the content of documents. Latent Semantic Index(LSI) is constructed to compute a cosine similarity between the documents and the terms and then Word2Vec model is built on the collection of contracts, patents and research papers for later use in query expansion. On searching phase, the queries are extended using Word2Vec model and searched in LSI to retrieve relevant documents. The results are shown to user via a GUI. Most of the approaches on this work are followed in our news search system. Lucene indexer is used to construct our term-document space. Morever, Word2Vec model is built on the collection of news for query expansion purpose.

CHAPTER 4

TURKISH NEWS SEMANTIC SEARCH SYSTEM

In this chapter, the structure of the search system for Turkish news is explained. Before presenting the proposed system, a general overview of the search system is given. Then the components of the system are explained along with the implementation details.

4.1 General Overview

A general overview of our Turkish News Search Application is shown in Fig 4.1.

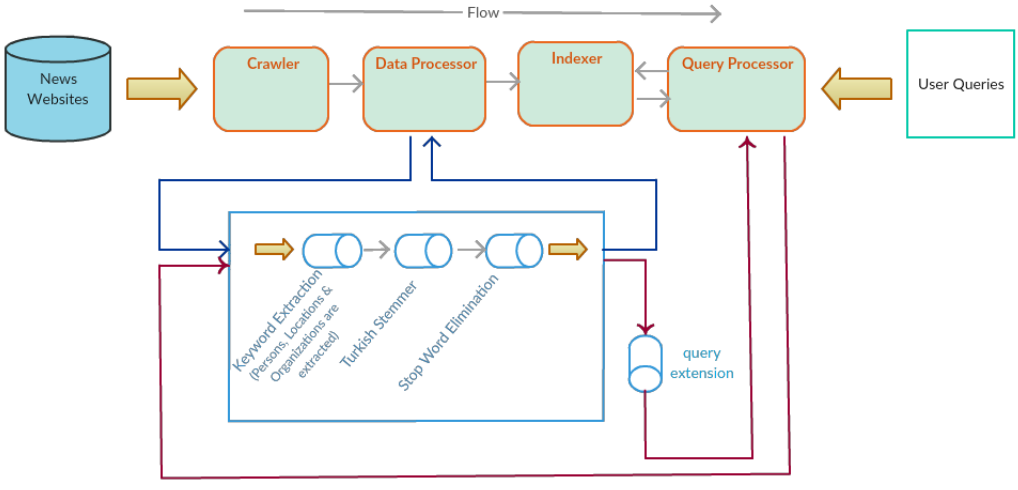

Figure 4.1: General overview of the system

Data from news websites are crawled by using RSS Feed facilities provided by News Agencies. Collected news fall into three categories, namely agenda, world and politics. Agencies used as news sources are: Hürriyet, Milliyet, NTV, Takvim, Sabah, CNNTürk, Radikal, Posta and Cumhuriyet. The articles of a webpage are extracted using XML parser [39]. It takes a URL as input and extracts XML tag filters based on our design for the presentation format used by each news website. The outcome of this procedure is a plain text containing only title, link, description and publication date. In addition to RSS parser results, we also added category and source information to the crawled news. These information is held in a FeedMessage class as seen in Fig 4.2. These FeedMessages are then given as input to the data processor module.

```
public class FeedMessage {

      String title;
      String description;
      String link;
      String category;
      String source;
      String publicationDate;
      String orgTitle;
      String orgDescription;
```

Figure 4.2: Feed Message Structure

Data processor applies tokenization, keyword extraction (by using NER) and stemming on the collected data. In the tokenization process, all non-word characters (e.g punctuations) and stop-words which have almost no sense for analysis (e.g pronuns and interjections) are removed. After that, keyword extraction process finds and classifies the named entities in the collected data. These named entities contain person names, locations and organizations. The extracted entities contain valuable information and they are indexed for searching. After NER, we apply stemming to reduce the words to a common base form. With the use of a stemmer, different forms of one word (“geldi”, “geliyor”) can be reduced to one conventional form (“gel”).

Indexer creates a term-document space to analyze the relationships between a set of documents and the terms they contain. Apache Lucene is used for indexing our pre-processed data. Lucene can be easily trained on big data sets and it provides fast and efficient search on the indexes. A graphical user interface (GUI) is provided to accept the user queries. The GUI consists of a search box and a date-picker. The user can set a filter on search by selecting a range of date. The search results are shown on the GUI in ranked order.

Query processor takes user's queries and gives the query to the data processor. In this way, all the steps of data processing (keyword extraction, stemming and stop-word elimination) are applied on the user's queries. After that, the query expansion process begins. In the query expansion, Word2Vec and WordNet for Turkish are used. The extended words are used in the search to improve the performance of retrieval. The search results are shown to the user via the GUI.

4.2 Data Processor Module

This module takes FeedMessages and processes the data in title and description part of a message. This process starts with the elimination of the numeric characters and punctuation. Then Named Entity Recognition (NER), stemming, stop-word elimination and data storage are applied in order.

4.2.1 Keyword Extraction

In the keyword extraction process, we used the approach of Tatar and Çiçekli [22]. We added the library provided to our system. The library contains a method for getting annotated words. Annotated words is a list of words in type Word, which is a class containing word itself and WordType. WordType can be abbreviation, person, location, organization, continent,city, etc. Using Keyword Extraction API, named entities from FeedMessages are extracted.These extracted keywords are not given into stemming process. For instance, when "Obama" is given, stemmer would return "oba", which is a generic name in Turkish.

4.2.2 Turkish Stemmer

Stemming is an important process in the indexing and search systems. Stemmer truncates the suffixes from a word and improves the recall by reducing words to their actual roots. The words "işlemciler", "işlemcilerin", "işlemciye" will all be indexed and searched only with their root form "işlemci". Turkish words consist of three parts: root, derivational suffixes and inflectional suffixes. We expect the stemmer to analyze the words and remove inflectional suffixes from the word. For the word “dengedeki”, expected result will be “denge” and for the word “dengelediğimizde” it will be “dengeledik”.

We used resha-turkish-stemmer for stemming purposes [40]. This stemmer uses a stem dictionary generated by Nuve [41], which is a Natural Language Processing Library for Turkish. Nuve makes morphological analysis, morphology generation, stemming, boundary detection and n-gram extraction. For a word, it gives the most possible stem without considering the neighbor words. For stemming purpose, complex morphological analysis will be needed. In that case, the word “aralığında” is stemmed and inflectional suffixes are removed, resulting in “aralığ”. Nuve can also handle a situation like this example, and gives “aralık” as the result. The dictionary generated by Nuve contains more than 1.1 million word-stem pairs. Users can also extend this dictionary by adding their custom word-stem pairs.
After keyword extraction, we give FeedMessages as an input to the stemmer. It updates the FeedMessages with stemmed words. At the end of this process, extracted keywords (named entities) are added to the related members of FeedMessages.

4.2.3 Stop-word Elimination

Tsz-Wai Lo,et al. [42] proposes a definition that words in a document that occur frequently but are meaningless in terms of Information Retrieval (IR) are called stop-words. In their article, it is claimed that a stop-word does not contribute towards the context or information of the documents and they should be removed during indexing as well as before being queried by an IR system. This stop-word concept dates back to early days of Information Retrieval. Luhn [43] labeled highly frequent words, such as "and" and "to" as common or noise words and named them as stop-words which can even be ignored. To increase the relevancy, we constructed a stop-word list, which contains common Turkish words, such as "ve", "için", "çünkü" and removed stop-words from the title and description of FeedMessages. This was the last data process step before storing the data. Below, there is an example title and description of FeedMessage before and after the data processing done.

Initial Feed Message:
title: Mardin'de 171 terörist etkisiz hale getirildi
description: Mardin´in Nusaybin ilçesinde sokağa çıkma yasağının 23. gününe girilirken, Mardin Valiliği kentte tüm hızıyla devam eden operasyonlara ilişkin açıklama yaptı. Valilik 3'ü sağ yakalanmak üzere toplam 171 terörist etkisiz hale getirilirken, 32 güvenlik görevlisinin şehit olduğunu bildirdi

After Data Processing:
title: terörist etkisiz hal getiril Mardin
description: ilçe sokak çıkma yasak gün giril valilik kent hız devam eden operasyon ilişkin açıklama yap valilik sağ yakalanmak üzere toplam terörist etkisiz hal getiril güvenlik görevli şehit olduk bildir Mardin Nusaybin Mardin Mardin Valiliği

4.2.4 Storage

In our prototype, we used MySQL relational database management system. We have one table in our database, named news. The structure of the table is shown in Fig 4.3 below.

Field	Type
id	bigint(20)
title	varchar(1000)
link	varchar(1000)
description	mediumtext
category	varchar(1000)
source	varchar(1000)
date	datetime
orgTitle	varchar(1000)
orgDescription	mediumtext

Figure 4.3: News table

id: primary key (necessary for uniqueness of object)
title: title of the news article (data processed title)
description: detailed explanation part of the news (data processed description)
category: news category (agenda,economy, world, politics or Turkey)
source: source of the news article (Hürriyet, Milliyet, NTV, Takvim, Sabah, CNNTürk, Radikal, Posta or Cumhuriyet)
date: The information oftime of news in the form of "YYYY-MM-DD HH:MM:SS" as in 2016-04-16 18:53:24
orgTitle: original title of the news article (without the data process)
orgDescription: The original, explanation part of the news (without data process)

FeedMessages are inserted into table "news" in our database.The title and description parts of FeedMessages are also saved into a text file, namely "newsDescriptions.txt" for later use in co-occurence calculations.

4.3 Indexing Module

We used Apache Lucene for indexing purposes. Apache Lucene is a high-performance, full-featured text search engine library written entirely in Java [44]. Lucene provides high performance in indexing and is efficient and accurate in search algorithms. In the following Fig 4.4, indexing process is illustrated.

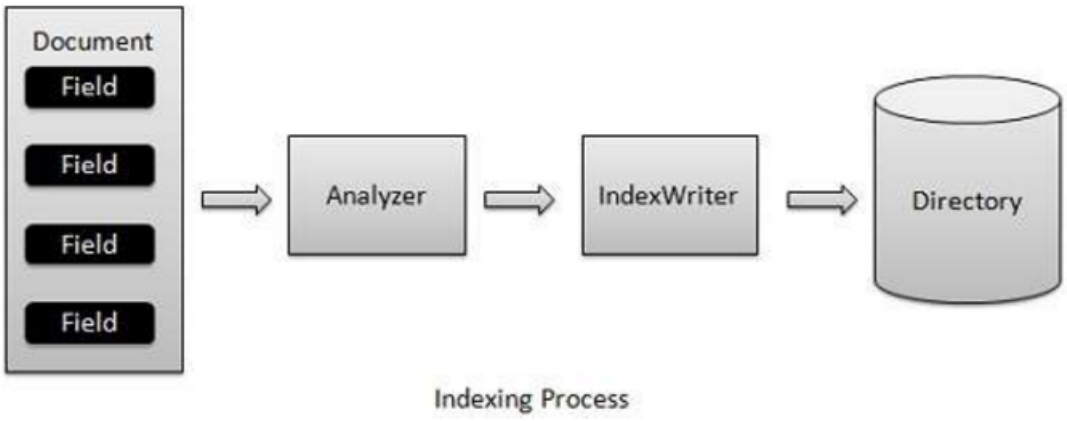

Figure 4.4: Indexing process

Field: Contains the content of the actual document. (for example, “title” is a field with terms “kar yağış Ankara”).
Document: Set of fields.
Analyzer: Gets the tokens from the text which is to be indexed.
IndexWriter: Inserts/updates the indexes during the indexing process.
Directory: Location where the indexes are stored.

We indexed the database "news". The indexed fields are “id”, title”, description”, “category” and “source”. Lucene creates inverted index which allows fast, full-text searching. Inverted index, at minimum, matches each word with a list of documents the word appears in.

Internal structure of the indexing is a group of files such as “Field names (.fnm)”, “Term dictionary (.tim)”, “Term frequencies(.frq)”, “Term positions (.prx)”, etc. Fig 4.5 , represents a slice of our sample news index.

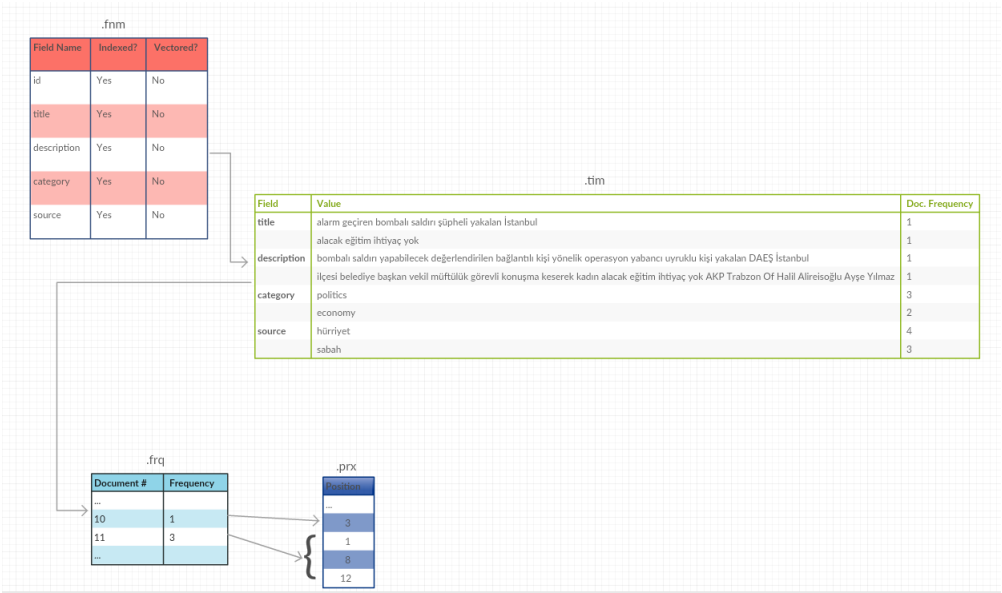

Figure 4.5: Lucene index format

For scoring, Lucene uses a combination of Vector Space Model (VSM) and Boolean Model to determine the relevancy of user query and document [45]. For a query “q”, for example, VSM scores the document “d” using the cosine similarity of weighted query vector V(q) and weighted document V(d). The weights are tf-idf values (term frequency - inverse document frequency).Cosine-similarity is calculated by the following Eqn. 4.1.

$$cosine - similarity = \frac{V(q) - V(d)}{|V(q) \times V(d))|} \tag{4.1}$$

Search results of Lucene can be influenced by boosting at "index time" or "query time".

Index-time boost: Done before adding document to index.

Query-time boost: Done by applying a boost to query at query time.

Lucene score calculation is done by the following Eqn. 4.2:

$$score(q,d) = coord(q,d) \times queryNorm(q) \times \sum_{t \in q} tf(t \in d) \times idf(t^2) \times t.getBoost() \times norm(t,d) \tag{4.2}$$

where,

coord (q,d): Search time score factor based on the number of query terms found in document.

queryNorm(q): Search time factor to make the scores from different queries comparable (calculated by Eqn. 4.3). It does not affect the document ranking.

$$\frac{1}{\sqrt{q.getBoost()^2 \times \sum_{t \in q}(idf(t) \times t.getBoost()^2)}} \tag{4.3}$$

tf(t in d): Term's frequency, the number of times the term appears in currently scored document d (calculated by Eqn. 4.4).

$$tf(t \in d) = \sqrt{frequency} \tag{4.4}$$

idf(t): Inverse document frequency, the number of documents in which the term t occurs (calculated by Eqn. 4.5).

$$idf(t) = 1 + log(\frac{numOfDocuments}{docFreq + 1}) \tag{4.5}$$

t.getBoost(): Search time boost of the term t in query q.

norm(t , d): Covers the indexing time boost and the length factors (calculated by Eqn. 4.6)
Field Boost: Boost added to the field before adding it to document.
lengthNorm: Computed when adding a document to index, shorter fields contribute more to score.

$$norm(t,d) = lengthNorm \times \prod_{field\ f\ in\ d\ named\ as\ t} f.boost() \qquad (4.6)$$

We indexed our news data using two indexing types, namely BASIC-INDEX and FIELD-BOOSTED-INDEX. In BASIC-INDEX, we made classic indexing on the news taken from our database. On the other hand, in FIELD-BOOSTED-INDEX, we give the title field a boost factor of 4.0f, to give the title terms more importance than other terms of the news at indexing time. S.Pant et al. [46] states in their work about relevance ranking that a search term which appears inside a <TITLE> </TITLE> tag pair might be given a greater relevance weighting than the same word in the same document but in normal body text. Because title gives more accurate and concise description of a page's content.

4.4 Query Expansion Module

We used Word2Vec [3] and WordNet for Turkish [4] to find the semantically related words to user's query.

4.4.1 Word2Vec

A corpus is given to Word2Vec as input and the vector representations of words are taken as output. Firstly, a vocabulary from the training text data is constructed and then the word vectors are learned. The resulting vector file can be used in the calculation of distance of a word to other words. Word2Vec has two learning models Continuous Bag of Words (CBOW) and skip-gram model which

are shallow neural models introduced in [23]. General architectures of the two models are given in Fig 4.6 below.

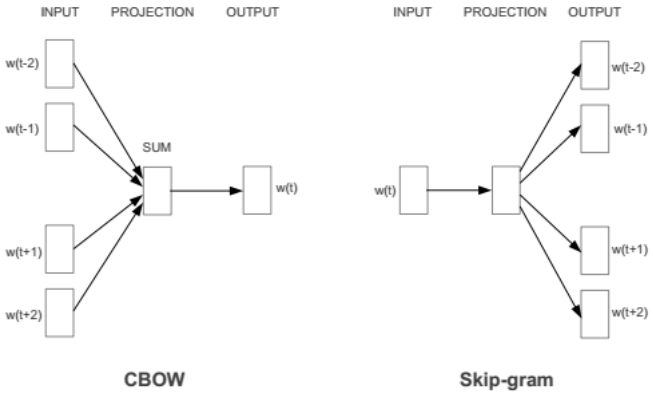

Figure 4.6: Architecture for the CBOW and Skip-gram method [23]

CBOW model is based on Feedforward Neural Net Language Model (NNLM) [47]. It consists of input, projection and output layers as in NNLM. It does not contain hidden layer of NNLM. NNLM predicts the probability of a word by considering the past n words, while CBOW predicts the probability of a word by considering past and future n/2 words. In CBOW model, mean vector of projections are calculated for the context words and then these vectors are used to predict the target word. Context means the window of words to the left and to the right of a target word. When the input to the model is the words "w_{i-2}, w_{i-1}, w_{i+1}, w_{i+2}", the output will be "w_i".

Skip-gram model predicts the context given a word. When the input to the model is "w_i", the output can be "w_{i-2}, w_{i-1}, w_{i+1}, w_{i+2}". We used skip-gram model while training as suggested by Mikolov et al. [48]. In the paper [48], it is explained that skip-gram is better for infrequent words.

For the sentence "Başbakan Davutoğlu NATO Genel Sekreteri telefonda görüştü", with the window size of 1, we have the dataset of (context, target) pairs as below:

([Başbakan, NATO], Davutoğlu), ([Davutoğlu, Genel], NATO), ([NATO, Sekreteri], Genel),...

Skip-gram predicts each context word from its target word by inverting the contexts and the targets. Our purpose is the prediction of "Başbakan" and "NATO" from "Davutoğlu", "Davutoğlu" and "Genel" from "NATO", etc. Therefore with skip-gram model, our dataset becomes the following (input, output) pairs:

(Davutoğlu, Başbakan), (Davutoğlu, NATO), (NATO, Davutoğlu), (NATO, Genel), ...

Word2Vec has two different training methods, namely with / without negative sampling. User has a choice for training the model. We have chosen the negative sampling. In [48], it was shown that skip-gram with negative sampling outperforms other combinations. Our "newsDescriptions.txt" file is given as the input to Word2Vec. It contains 450716 words and Word2Vec produces vocabulary of size 9709. The output of Word2Vec is the vector representations of words. These representations are saved to file "vectors.bin" and then this file is used in finding a semantic similarity between words. Word2Vec finds the semantically similar words to a user's query. It computes cosine distance (dot product) between the vector representation of a query word and the other words in vocabulary. It returns words in ranked order according to their distance to the query word. When we entered "Ankara", first five words returned are shown in Table 4.1.We append the words with the cosine distance 0.5 and above to the query word.

Table4.1: First five Word2Vec results for the word "Ankara"

Word	Cosine distance
Gar	0.602061
Nazlıaka	0.595110
Sincan	0.555095
Aylin	0.534214
Tunç	0.532991

Scikit-Learn's implementation [49] of a dimensionality reduction algorithm called t-SNE (t-distributed stochastic neighbor embedding) is used to visualize the word vectors. t-SNE is a machine learning algorithm developed by van der Maaten, L.J.P and Hinton, G.E for dimensionality reduction [50]. Dimensionality reduction provides embedding of high-dimensional data into a space of two or three dimensions, which can be visualized in a plot. We constructed three example word sets, namely politics,world and agenda. Word sets are taken from our database. In the database "news", we are holding a field named "category". We constructed word sets by selecting "description" and "title" parts of news filtered by related category. At the end of it, we had three text files, "politics.txt", "world.txt" and "agenda.txt". From these text files, we constructed the vectors of related category with each vector of size 1000. We then use t-SNE algorithm and matplotlib (python 2D plotting library) to visualize the clusters. Implementation is done in python, importing t-SNE (from sklearn.manifold) and plt (from matplotlib.pyplot). Reduced vectors can be easily constructed using these libraries. Using these reduced vectors, we can draw a 2 dimensional plot containing politics, world and agenda vectors. Resulting plot is shown in Fig 4.7. On this plot, x and y represents the coordinates of the words in N dimensional space, where N is the size of the word vectors obtained.

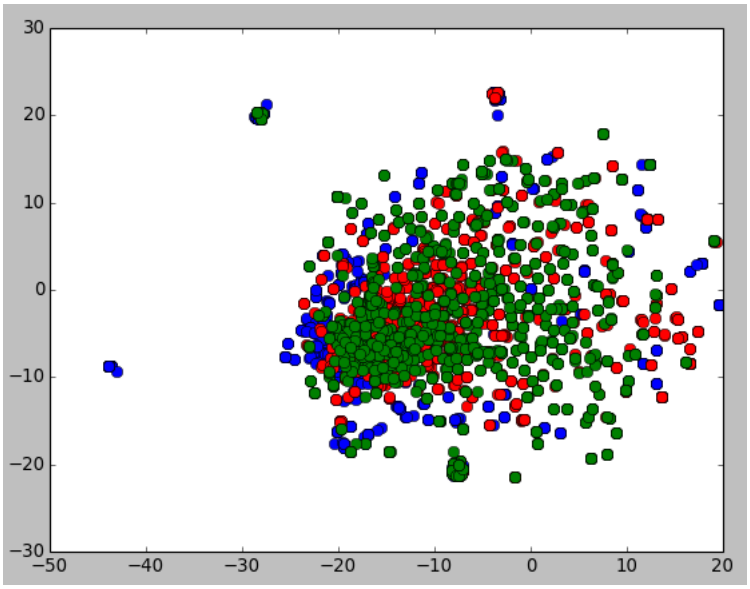

Figure 4.7: T-SNE projected clusters of politics words (blue), agenda words (red), and world words (green)

We also constructed a word set of Keywords from the "politics" category. We get only 500 keywords from our database in politics for showing purpose. Using word set of "politics", TSNE and matplotlib, the learned embeddings are visualized in Fig 4.8 and Fig. 4.9.

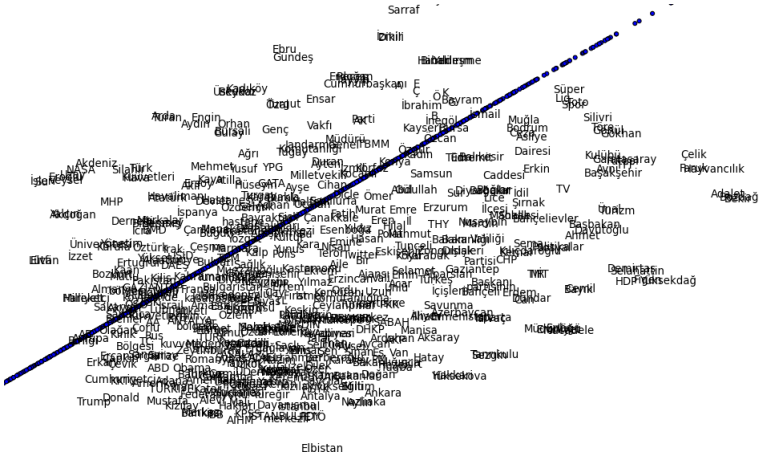

Figure 4.8: T-SNE projected cluster of politic keywords

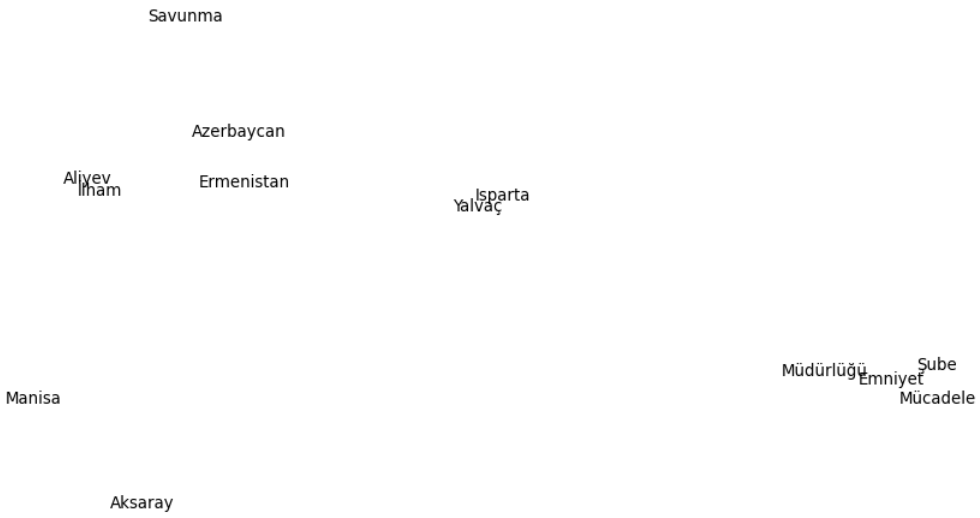

Figure 4.9: Zoomed version of Fig 4.8

We can see in Fig.4.9, words semantically related end up clustering nearby each other and unrelated words are separated.

4.4.2 WordNet for Turkish

We used Turkish WordNet developed by Oflazer [4].We modified Python3 API for WordNet XML [51] according to our needs. This API parses XML WordNet file and relates words to each other. We call the API in our Java-based application. For word "anarşist", the API returns :

ENG20-09170394-n anarşist:2 (an advocate of anarchism)

ENG20-02425170-a terörist:1, anarşist:1 (characteristic of someone who employs terrorism (especially as a political weapon))

We extract for example "terörist" from results as synonym of "anarşist" and use extracted synonyms in searching for query expansion purposes.

4.5 Searching Module

4.5.1 User Interface

We provided the user a keyword-based news search interface. In the GUI, there is a text-box where user can enter words to search (Fig. 4.10). User can also select a date interval for the search.

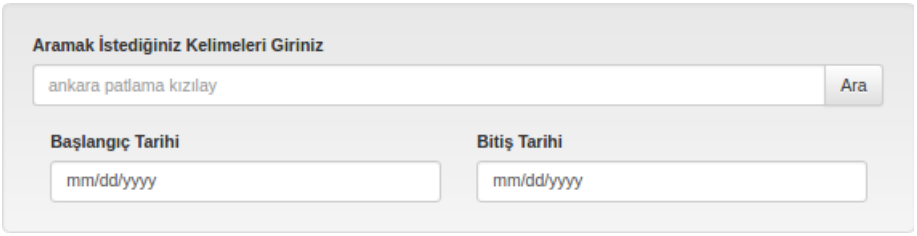

Figure 4.10: Screenshot of the search screen

After hitting the search button, a ranked list of news related to user search is returned, as shown in Fig 4.11. When the user clicks on one of the results, he/she will be redirected to the actual site where that news is published.

Figure 4.11: A few results returned for the example query

4.5.2 Searching and Ranking

Searching and ranking process starts with the entry provided by user. User can enter more than one word to search. Query word is processed by Data Processer Module. Firstly, keywords are extracted from query and then stemmed by Turkish stemmer to get rid of the suffixes in the Turkish word. Then, stop-word elimination is done. After the data processor module completes its job, query expansion process begins. Using Word2Vec and WordNet for Turkish, query expansion is done for all the words in user query. After the query expansion is done, searching and ranking on the expanded query begins. At the end of this process, related news to the user query is returned.

4.5.2.1 Pre-processing of User Query

We can explain the overall pre-process of a query on the following example:

1. User enters the words "başbakan Obama" as the search query with an option to select start and end date of news. If the user does not select a date range, all news related to the query will be searched. From these query words, keywords are extracted and Turkish stemmer is applied. Resulting

query words are held in the variable orgWords in the form of String.

2. Word2Vec is run to expand the query. It returns "Barack" ,"Angela", "görüşme", and "Fidel" as co-occurred words. Now the query word becomes “başbakan Obama Barack Angela görüşme Fidel”. It is held in the variable expandedWords in the form of String.

3. Synonyms of the words in the query are found using WordNet for Turkish. “başvekil” is returned as a synonym of "başbakan" and "müzakere", "istişare", "toplantı" are returned as the synonyms of "görüşme". The results are appended to expandedWords and it finally becomes "başbakan Obama Barack Angela görüşme Fidel başvekil müzakere istişare toplantı".

4. The keywords are extracted from expandedWords, which are "Obama", "Barack", "Angela" and "Fidel" and held in the variable "keywords" in the form of String.

4.5.2.2 Searching and Ranking of the Expanded Query

Firstly, orgWords are searched using FIELD_BOOSTED_INDEX, which was created by giving the title field a boost factor of 4.0f at indexing time. Increasing the boost factor by more than 4.0f did not change the returned results. Lucene search on this index returns the ranked documents related to orgQuery words. The ranked documents taken from Lucene is added to our resulted "news" array, which holds the type of "News".

We constructed RUN-TIME-BOOSTED-QUERY, which is used during the search. It consists of title and description queries. The title query contains the keywords. We give it a boost factor of 4.0f to give a higher rank to the news which contain these keywords. The description query contains all the expanded words. The RUN-TIME-BOOSTED-QUERY is used by Lucene to search on FIELD-BOOSTED-INDEX’ed documents and the ranked documents are returned. These documents are appended to the “news” array.

Ranked results of an example query search “Beşiktaş stat açılışı” can be seen on Fig. 4.12 below.

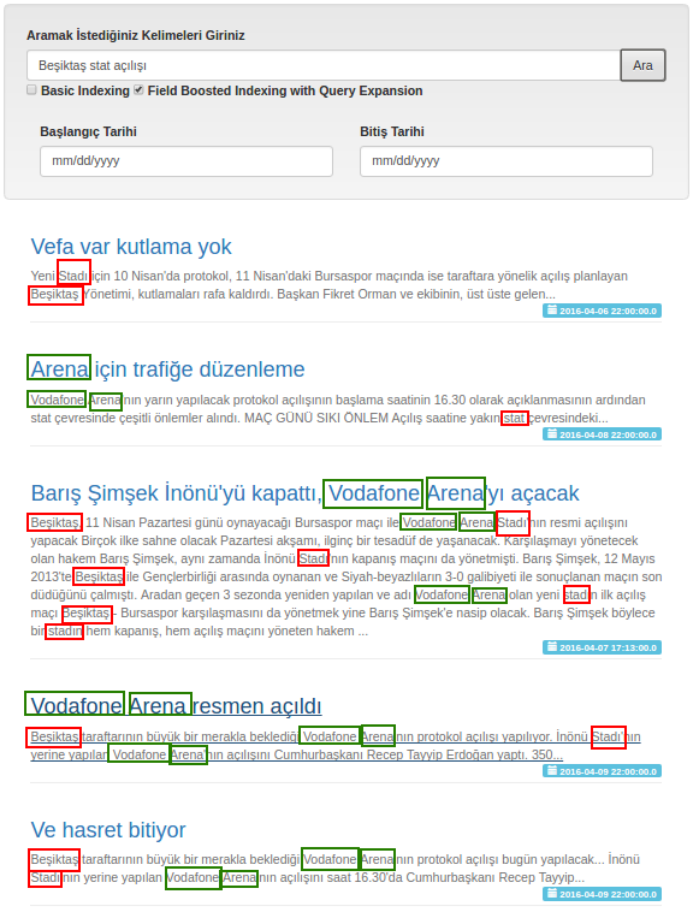

Figure 4.12: search result of "Beşiktaş stat açılışı"

In this figure, red squares are showing the words user entered and green ones are showing the words after query expansion process.

CHAPTER 5

EVALUATION

We evaluated our Semantic News Search Engine through two different experiments. The first one is conducted by comparing our search engine with the basic search done by Lucene. Lucene makes basic search on the user entered words, while our search engine makes semantic enhancement on the entered words. The second one is the evaluation of the complete system. In the evaluation of the complete system, the advantage of query expansions in retrieving relevant documents are shown.

5.1 Evaluation based on a baseline search system

We constructed a dataset to compare our search system with a baseline search system. This set contains the query words "Zerrab", "anarşist fiiller" and "besteci yaşamını yitirdi". For testing purposes, we added a choice for indexing type to our UI (Fig. 5.1).

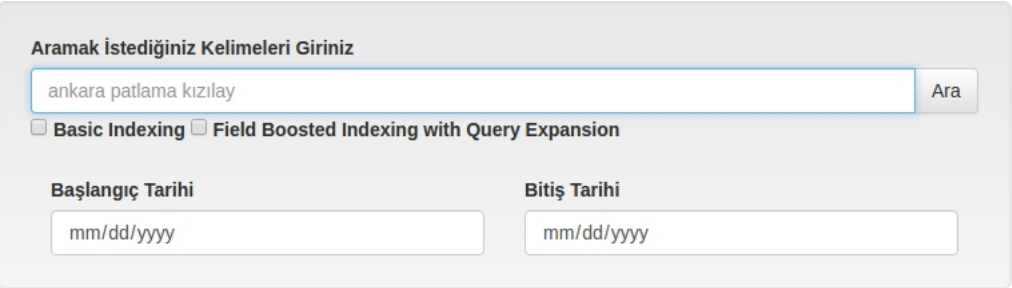

Figure 5.1: Index type added UI

When we enter "Zerrab" to the search bar, basic indexing returns the news containing word "Zerrab" in only "title" or "description" parts of news. It returns us only 4 results as shown in Table 5.1.

Table5.1: Search results for "Zerrab" in the basic search

Title No	Title
1	Reza Zerrab, kefaletle serbest kalmak için başvurdu
2	Reza Zarrab'ı tutuklatan ABD'li Savcı Bharara konuştu: Öğrendiğim ilk Türkçe sözcük adalet
3	Sarraf'ı tutuklatan savcı Bharara öğrendiği ilk Türkçe kelimeyi açıkladı
4	Zarrab Miami'den New York'a 4 farklı cezaevinde yatarak gidecek

When we search "Zerrab" in our semantically enhanced search engine, it returns more than 30 results related to the query word. The titles of first 15 news returned is shown in Table 5.2.

Table5.2: Search results for "Zerrab" in the semantic search

Title No	Title
1	Reza Zerrab kefaletle serbest kalmak için başvurdu
2	Reza Zarrab'ı tutuklatan ABD'li Savcı Bharara konuştu: Öğrendiğim ilk Türkçe sözcük adalet
3	Sarraf'ı tutuklatan savcı Bharara öğrendiği ilk Türkçe kelimeyi açıkladı
4	Zarrab, Miami'den New York'a 4 farklı cezaevinde yatarak gidecek
5	CHP heyeti Reza Zarrab duruşması gezisini erteledi
6	Reza Zarrab kefalet hakkından vazgeçti
7	"Zencani'nin paraları Zarrab'ın elinde" iddiası
8	Zarrab'ın savcısından Türkiye açıklaması
9	Reza Zarrab'ın tutuklanması dünya basınında
10	İşte Reza Zarrab'ı tutuklatan ABD'li savcı Preet Bharara
11	United States of America v. Rıza Sarraf
12	Reza Zarrab'ın avukatı Şeyda Yıldırım: Kefalet talebi ret veya kabul edilmedi
13	Sarraf'ı tutuklayan Savcı Bharara: Kimseden korkmadan işimizi yapacağız
14	Zarrab'lı darbe iddiası saçma
15	Reza Zarrab'ı tutuklatan ABD'li Başsavcı Preet Bharara: Sindiremezler

Our search engine returns more results when compared to the basic search. Query expansion module gives us the words "işadamı", "Preet", "Reza", "tutuklatan", "Şeyda", "karapara", "Sarraf", "Zarrab" and "Miami" as the extension to the query word. These words are also added to the searching phase.

Even if the title or description of news do not contain the query word, our search engine returns news related to the user's query. An example result for the word "Zerrab" is shown in Fig. 5.2.

Figure 5.2: A few news returned for the search "Zerrab"

When we enter "anarşist fiiller" for searching, basic indexing returns us only 2 results with news containing either the words "anarşist" or "fiil" as shown in Fig. 5.3.

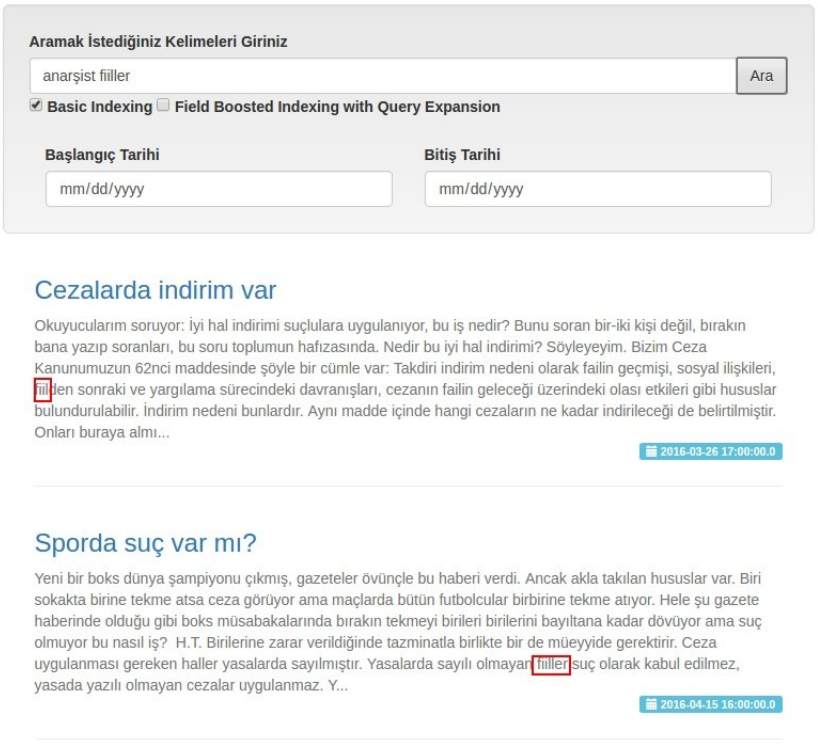

Figure 5.3: Search results for "anarşist fiiller" in the basic search

Our semantically enhanced news search engine returns more than 30 results related to the user's query "anarşist fiiller". These query words are expanded by the query expansion module with the words "terörist", "eylem" and "hareket". Some of the results returned are shown in Fig. 5.4.

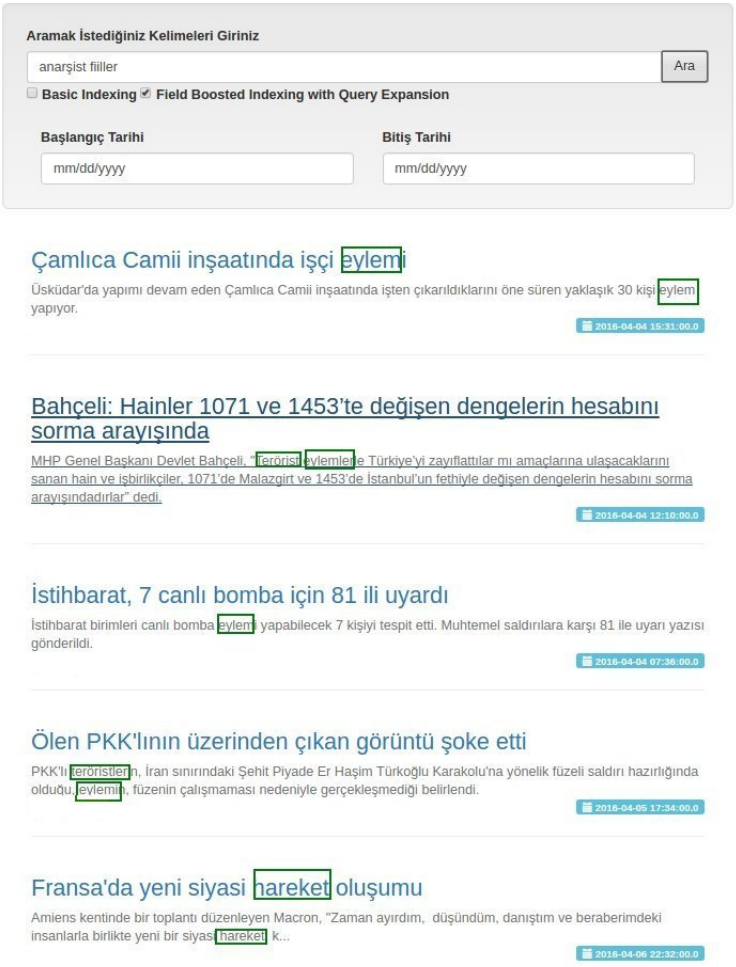

Figure 5.4: Search results for "anarşist fiiller" in the semantic search

When we enter "besteci yaşamını yitirdi" as the query, basic search returns us the results below as the first five results, which are shown in Fig. 5.5.

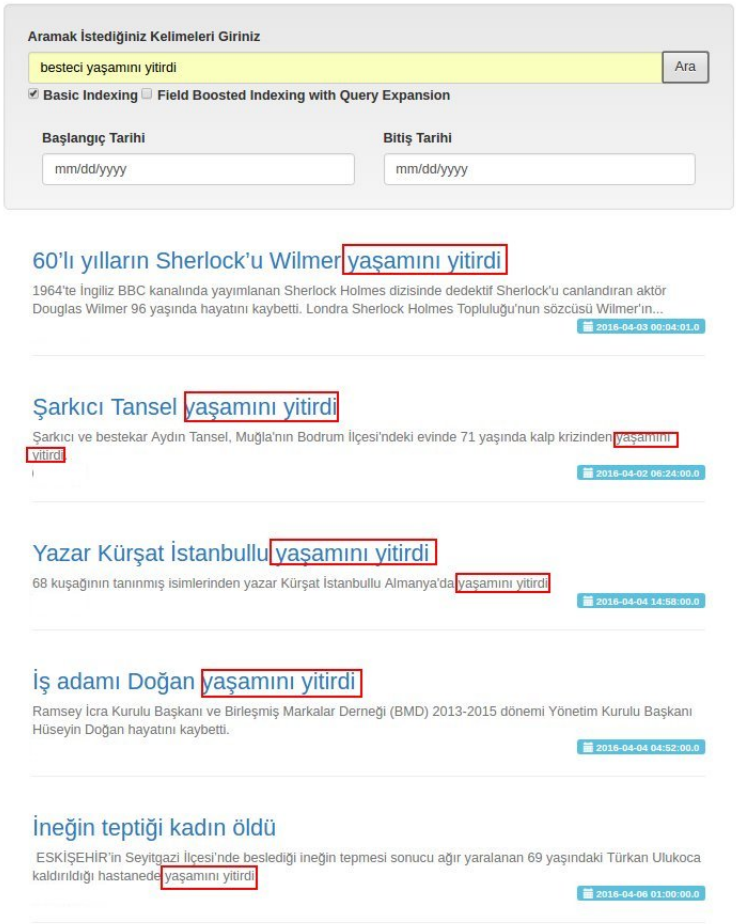

Figure 5.5: Search results for "besteci yaşamını yitirdi" in the basic search

In our search engine first five results for "besteci yaşamını yitirdi" are shown in Fig. 5.6. Our search engine puts more related results in the first five news. The user query is expanded with the words "bestekar", "müzisyen", "Özdemiroğlu", "kompozitör" and "hayat".

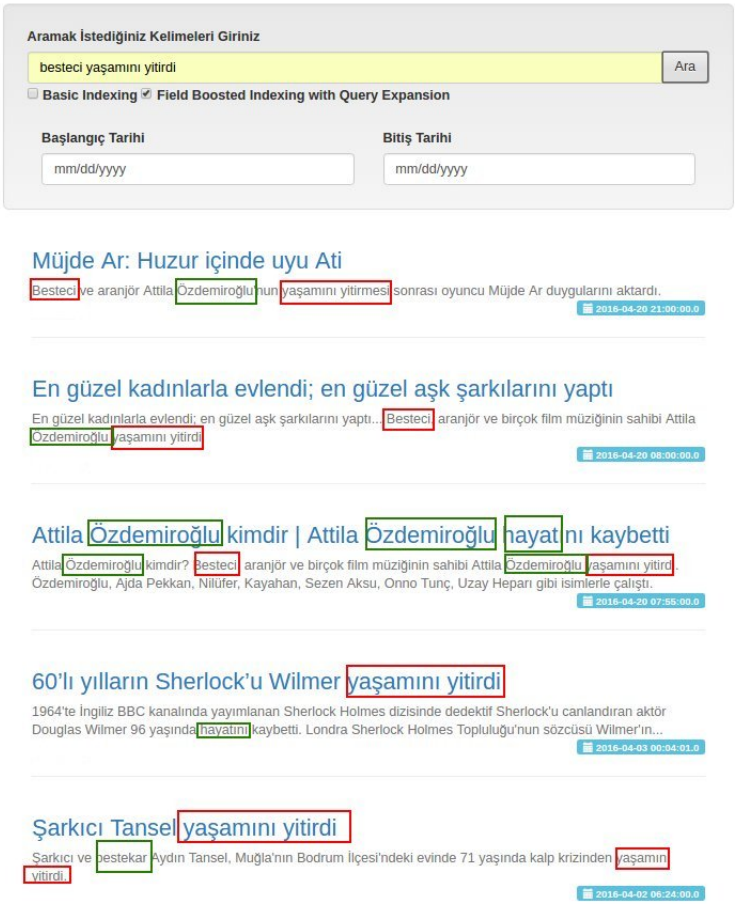

Figure 5.6: Search results for "besteci yaşamını yitirdi" in the semantic search

5.2 Evaluation of the Turkish News Search System

We evaluated our search system by following a similar approach used in the work [24]. For each category, five words are selected as a test query. Some of the selected words are "Zaventem", "Mossack" and "Angela". These words are selected among hot topics at the time of the data collected. For each word, we manually annotated 50 news related to the word to construct our gold standard dataset. To show the effect of context extension on search, we searched the news in five different ways:

1. Search by using a one-word initial query without context extension.

2. Search by applying context extension to a query and then removing the initial query. In this way, the performance of context extension can be shown.

3. Search by applying context extension to a query, removing the initial query and manually deleting the most inappropriate terms from extension.

4. Search by applying context extension to an initial query and using initial query and extension together.

5. Search by applying context extension to an initial query, removing the inappropriate terms from extension and using manually edited extension along with initial query.

For a test query, we compared the most relevant 20 and 40 documents retrieved by news search system with our gold standard dataset. Then we computed the mean average precision for the first 20 and 40 retrieved results. In the calculation of precision we used classic precision measure [52].

$$P = \frac{|D_{rel}| \cap |D_{ret}|}{|D_{ret}|} \tag{5.1}$$

where, D_{rel} is the number of relevant documents and D_{ret} is the number of retrieved documents. For the precision-recall plots, we calculated recall by using the following Eqn. 5.2.

$$P = \frac{|D_{rel}| \cap |D_{ret}|}{|D_{rel}|} \tag{5.2}$$

Recall of 100% can be achieved by returning all the news related to the user's query. That is why we used recall only for precision-recall plots. For average precision-recall graph, we calculated interpolated precision at 11 standard recall levels by using the following Eqn. 5.3:

$$P(R) = max\left\{P^{'} : R^{'} \geqslant R \wedge (R^{'}, P^{'}) \in S\right\} \tag{5.3}$$

where R is recall level, P is precision and S is the set of observed (R,P) points. The resulting average graphs for the agenda and politics at first 40 documents are shown in Fig. 5.7 and 5.8 respectively.

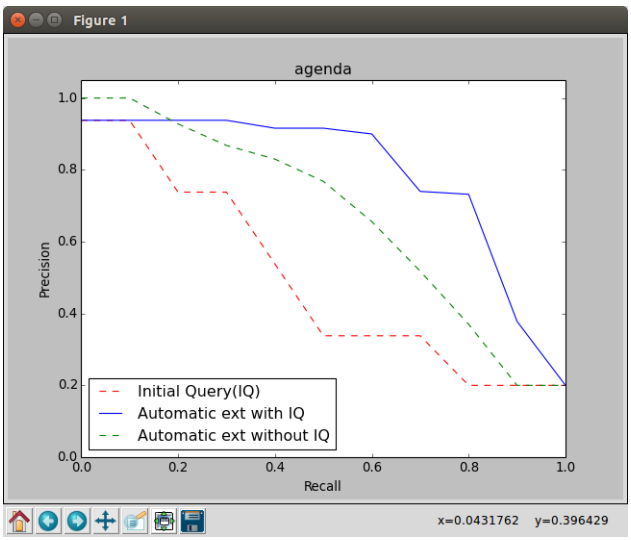

Figure 5.7: Precision-recall plot for the agenda on P@40

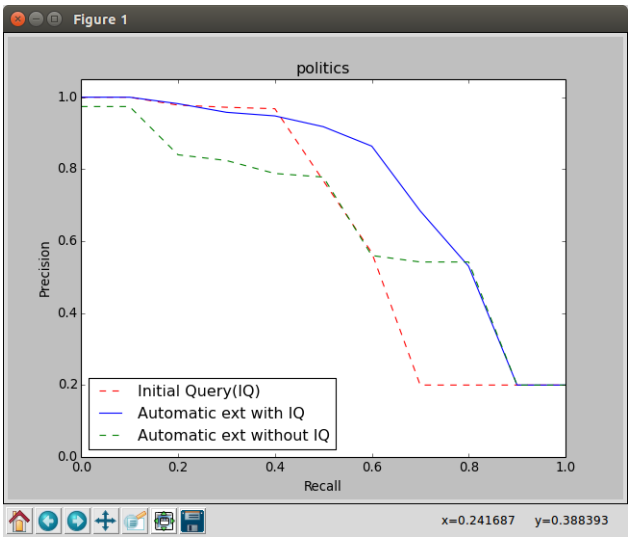

Figure 5.8: Precision-recall plot for the politics on P@40

The resulting average plot for the category world is shown in Fig. 5.9.

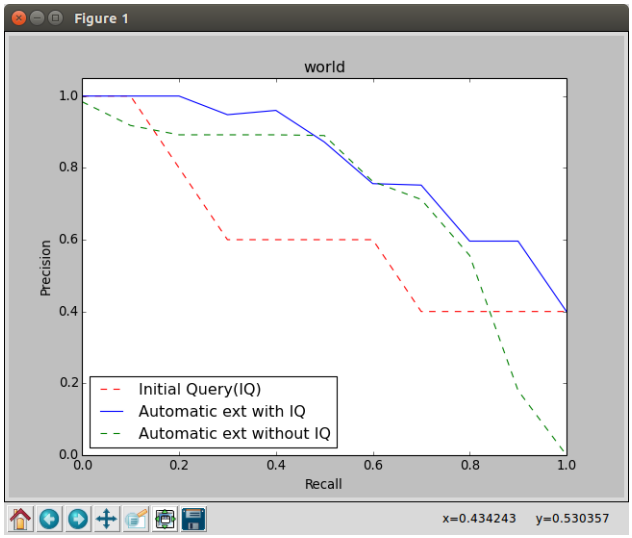

Figure 5.9: Precision-recall plot for the word on P@40

Table 5.3 and 5.4 show the statistics about the average precision at the 20 and 40 retrieved results under different categories.

Table5.3: P@20 under different categories

	Agenda	World	Politics	Mean
Initial Query (IQ)	0.6	0.74	0.91	0.75
Automatic ext. without IQ	0.7	0.81	0.7	0.74
Manually edited ext. without IQ	0.98	0.88	0.93	0.93
Automatic ext. with IQ	0.82	0.9	0.91	0.88
Manually edited ext. with IQ	1	0.94	0.95	0.96

Table5.4: P@40 under different categories

	Agenda	World	Politics	Mean
Initial Query (IQ)	0.25	0.61	0.63	0.5
Automatic ext. without IQ	0.66	0.67	0.62	0.65
Manually edited ext. without IQ	0.91	0.85	0.88	0.88
Automatic ext. with IQ	0.77	0.82	0.75	0.78
Manually edited ext. with IQ	0.99	0.89	0.91	0.93

From the Table 5.3 , we see that our semantic news search system gives higher precision at first 20 retrieved documents for the cases we applied extensions (both auto created and manually edited): 88% and 96% respectively.

In Table 5.4, it can be seen that initial query with automatically created extensions and along with manually edited content give the average precision of 78% and 93% respectively, while mean precision on the first 40 retrieved results is only 50% for the initial one-word query.

When we compare our approach (automatic extension with IQ) with the direct one word search (initial query only), we see that our approach shows 13% and 28% precision improvements at first 20 and 40 documents respectively.

CHAPTER 6

CONCLUSIONS

In this thesis, a semantic search engine on Turkish news domain is proposed and a web-based prototype for the search system is implemented. The system consists of four components: news crawler, data processor, indexer, and searcher. Firstly, the data from news websites are crawled using RSS news feeds. Next, data processor applies keyword extraction (by using NER), stemming and stop-word elimination on the crawled data. Then, the processed data is given to indexer to create a term-document space. The last component, searcher, takes the user's input via a GUI, extends the user's query with semantically similar words and performs a search on the index created by the indexer component. The result of the search is shown to user via the GUI.

The most important parts in our system are the query expansion and keyword extraction. User queries are expanded with WordNet synonyms and Word2Vec similarity relations. We also used NER to extract named entities to give higher rank to these entities during search. These approaches produce results more related to the user query when compared to the basic search system.

To conclude, we would like to say that our search system shows an acceptable performance in terms of information retrieval. While evaluating our system under different news categories, we found out that the best performance is achieved by using automatically created and manually edited expansions along with the initial query.

For a future study, this work can be extended by :

1. Using more categories of news (e.g economy, sport, magazine).
2. Using a broader range of date for news collection.
3. Adding category selection to the system (to improve relevance of information).
4. Adding data sources from other domains of Turkish like arts, papers, soccer or cinema (to show how our system will work with other data sources).

REFERENCES

[1] Google. `http://www.google.com`. Accessed: 2016-04-04.

[2] Google news turkey. `http://www.news.google.com.tr`. Accessed: 2016-04-04.

[3] word2vec. `https://code.google.com/archive/p/word2vec/`. Accessed: 2016-04-05.

[4] Orhan Bilgin, Özlem Çetinoğlu, and Kemal Oflazer. Building a wordnet for turkish. *Romanian Journal of Information Science and Technology*, 7(1-2):163–172, 2004.

[5] Efthimis N Efthimiadis. Query expansion. *Annual review of information science and technology*, 31:121–187, 1996.

[6] Joseph John Rocchio. Relevance feedback in information retrieval. 1971.

[7] Roberto Navigli. Word sense disambiguation: A survey. *ACM Computing Surveys (CSUR)*, 41(2):10, 2009.

[8] Claudio Carpineto, Stanislaw Osiński, Giovanni Romano, and Dawid Weiss. A survey of web clustering engines. *ACM Computing Surveys (CSUR)*, 41(3):17, 2009.

[9] S. K. Michael Wong, Wojciech Ziarko, Vijay V. Raghavan, and PCN Wong. On modeling of information retrieval concepts in vector spaces. *ACM Transactions on Database Systems (TODS)*, 12(2):299–321, 1987.

[10] George A Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J Miller. Introduction to wordnet: An on-line lexical database*. *International journal of lexicography*, 3(4):235–244, 1990.

[11] Ellen M Voorhees. Query expansion using lexical-semantic relations. In *SIGIR'94*, pages 61–69. Springer, 1994.

[12] Balkanet - design and development of a multilingual balkan wordnet. `http://www.dblab.upatras.gr/balkanet/`. Accessed: 2016-03-05.

[13] Dan Tufis, Dan Cristea, and Sofia Stamou. Balkanet: Aims, methods, results and perspectives. a general overview. *Romanian Journal of Information science and technology*, 7(1-2):9–43, 2004.

[14] Rdf 1.1 concepts and abstract syntax. https://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/. Accessed: 2016-04-04.

[15] Owl 2 web ontology language. https://www.w3.org/TR/owl2-overview/. Accessed: 2016-04-04.

[16] Olivier Corby, Rose Dieng-Kuntz, and Catherine Faron-Zucker. Querying the semantic web with corese search engine. In *ECAI*, volume 16, page 705, 2004.

[17] Corese user manual. https://www-sop.inria.fr/acacia/soft/corese/manual/. Accessed: 2016-04-04.

[18] Eyal Oren, Christophe Guéret, and Stefan Schlobach. *Anytime query answering in RDF through evolutionary algorithms.* Springer, 2008.

[19] Debajyoti Mukhopadhyay, Aritra Banik, Sreemoyee Mukherjee, Jhilik Bhattacharya, and Young-Chon Kim. A domain specific ontology based semantic web search engine. *arXiv preprint arXiv:1102.0695*, 2011.

[20] Pablo Castells, Ferran Perdrix, E Pulido, Mariano Rico, R Benjamins, Jesús Contreras, and J Lorés. Neptuno: Semantic web technologies for a digital newspaper archive. In *The Semantic Web: Research and Applications*, pages 445–458. Springer, 2004.

[21] Eduardo Lupiani-Ruiz, Ignacio GarcíA-Manotas, Rafael Valencia-GarcíA, Francisco GarcíA-SáNchez, Dagoberto Castellanos-Nieves, Jesualdo TomáS FernáNdez-Breis, and Juan Bosco CamóN-Herrero. Financial news semantic search engine. *Expert systems with applications*, 38(12):15565–15572, 2011.

[22] Harith Alani, Sanghee Kim, David E Millard, Mark J Weal, Wendy Hall, Paul H Lewis, and Nigel R Shadbolt. Automatic ontology-based knowledge extraction from web documents. *Intelligent Systems, IEEE*, 18(1):14–21, 2003.

[23] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[24] Nikita Nikitinsky, Dmitry Ustalov, and Sergey Shashev. An information retrieval system for technology analysis and forecasting. *Proceedings of the AINL-ISMW FRUCT/Ed. by Sergey Balandin, Tatiana Tyutina, Ulia Trifonova*, pages 52–59, 2015.

[25] Travis Goodwin and Sanda M Harabagiu. Utd at trec 2014: Query expansion for clinical decision support. Technical report, DTIC Document, 2014.

[26] Özkan Bayraktar and Tuğba Taşkaya Temizel. Person name extraction from turkish financial news text using local grammar-based approach. In *Computer and Information Sciences, 2008. ISCIS'08. 23rd International Symposium on*, pages 1–4. IEEE, 2008.

[27] Dilek Küçük and Adnan Yazıcı. A hybrid named entity recognizer for turkish. *Expert Systems with Applications*, 39(3):2733–2742, 2012.

[28] Serhan Tatar and Ilyas Cicekli. Automatic rule learning exploiting morphological features for named entity recognition in turkish. *Journal of Information Science*, 37(2):137–151, 2011.

[29] Reyyan Yeniterzi. Exploiting morphology in turkish named entity recognition system. In *Proceedings of the ACL 2011 Student Session*, pages 105–110. Association for Computational Linguistics, 2011.

[30] Gökhan Akin Seker and Gülsen Eryigit. Initial explorations on using crfs for turkish named entity recognition. In *COLING*, pages 2459–2474, 2012.

[31] N Crofts, M Doerr, T Gill, S Stead, and M Stiff. Definition of the cidoc objectoriented conceptual reference model and crossreference manual (2003), 2000.

[32] Hamish Cunningham, Diana Maynard, Kalina Bontcheva, and Valentin Tablan. A framework and graphical development environment for robust nlp tools and applications. In *ACL*, pages 168–175, 2002.

[33] Iptc standards. `https://iptc.org/standards/`. Accessed: 2016-05-05.

[34] Rdql - a query language for rdf. `https://www.w3.org/Submission/RDQL/`. Accessed: 2016-04-05.

[35] Leonidas Kallipolitis, Vassilis Karpis, and Isambo Karali. Semantic search in the world news domain using automatically extracted metadata files. *Knowledge-Based Systems*, 27:38–50, 2012.

[36] M Fox, M Barbuceanu, M Gruninger, and J Lin. An organization ontology for en4 terprise modelling. simulating organizations: Computational models of institutions and groups, m. pritula, k. carley & l. gasser (eds), menlo park ca: Aaai, 1996.

[37] Chris Partridge and Milena Stefanova. A synthesis of state of the art enterprise ontologies. *Lessons Learned. The BORO Program, LADSEB CNR*, 2001.

[38] LS Alonso, LJ Bas, Sergio Bellido, Jesús Contreras, Richard Benjamins, and MJ Gomez. Wp10: Case study ebanking d10. 7 financial ontology. *Data, Information and Process Integration with Semantic Web Services, FP6-507483*, 2005.

[39] Java dom parser - parse xml document. http://www.tutorialspoint.com/java_xml/java_dom_parse_document.htm. Accessed: 2016-04-04.

[40] resha-turkish-stemmer. https://github.com/hrzafer/resha-turkish-stemmer. Accessed: 2016-03-05.

[41] Nuve. https://github.com/hrzafer/nuve. Accessed: 2016-03-05.

[42] Rachel Tsz-Wai Lo, Ben He, and Iadh Ounis. Automatically building a stopword list for an information retrieval system. In *Journal on Digital Information Management: Special Issue on the 5th Dutch-Belgian Information Retrieval Workshop (DIR)*, volume 5, pages 17–24. Citeseer, 2005.

[43] Hans Peter Luhn. The automatic creation of literature abstracts. *IBM Journal of research and development*, 2(2):159–165, 1958.

[44] Apache lucene core. https://lucene.apache.org/core. Accessed: 2016-04-04.

[45] Tfidfsimilarity. https://lucene.apache.org/core/5_4_0/core/org/apache/lucene/search/similarities/TFIDFSimilarity.html. Accessed: 2016-03-05.

[46] Sangam Pant, David L Andre, Gray Watson, Richard M Green, and Michael J Schiegg. Computer system with user-controlled relevance ranking of search results, January 4 2000. US Patent 6,012,053.

[47] Yoshua Bengio, Holger Schwenk, Jean-Sébastien Senécal, Fréderic Morin, and Jean-Luc Gauvain. Neural probabilistic language models. In *Innovations in Machine Learning*, pages 137–186. Springer, 2006.

[48] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[49] sklearn.manifold.tsne. http://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html. Accessed: 2016-04-05.

[50] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(2579-2605):85, 2008.

[51] Python3 api for wordnet xml (hungarian wordnet / balkanet / visdic format). https://github.com/ppke-nlpg/pywnxml. Accessed: 2016-03-05.

[52] Christopher D Manning, Prabhakar Raghavan, Hinrich Schütze, et al. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.