**NEWS INFORMATION RETRIVEAL OVER ALBANIAN LANGUAGE DOCUMENTS**

**Master Degree**

**Berru QAZIMI**

**Eskişehir 2018**

# NEWS INFORMATION RETRIEVAL OVAR ALBANIAN LANGUAGE DOCUMENTS

**Berru QAZIMI**

**MASTER DEGREE**

Computer Engineering Program

Supervisor: Assoc. Prof. Dr. Özgür YILMAZEL

Eskişehir
Anadolu University
Graduate School of Science
June 2018

## FINAL APPROVAL FOR THESIS

This thesis titled "**News Information Retrieval Over Albanian Language Documents**" has been prepared and submitted by **Berru QAZIMI** in partial fullfillment of the requirements in "Anadolu University Directive on Graduate Education and Examination" for the Degree of MSc in Computer Engineering Department has been examined and approved on 22/06/2018.

| **Committee Members** | **Title Name Surname** | **Signature** |
|---|---|---|
| Member (Supervisor) | : Assoc. Prof. Dr. Özgür YILMAZEL | ..... |
| Member | : Asst. Prof. Ahmet ARSLAN | ..... |
| Member | : Asst. Prof. Muammer AKÇAY | ..... |

**Prof.Dr. Ersin YÜCEL**

**Director of Graduate School of Sciences**

# ÖZET


ARNAVUTÇA DİLİNDE HABER ARAMA VE ERİŞİM SİSTEMİ

Berru QAZIMI

Bilgisayar Mühendisliği Anabilim Dalı
Anadolu Üniversitesi, Fen Bilimleri Enstitüsü, Haziran 2018

Danışman: Doç. Dr. Özgür YILMAZEL

Bu araştırmada Arnavutça Dilini ve toplumun gelişmesi için belirli bir dilde bilgi edinme ve sistem uyarlamalarına erişme ilkeleri incelenmiştir. Serbest metin aramalarının dil kullanımına bağlı olduğu bilinmektedir. Sonuç olarak, Standart Analizör'ün içine Arnavutça Dilinin bağlaçları eklenmiş ve standart analizörde değişiklik yapılmıştır.

Bu bağlaçlar eklendikten sonra, tam metin indeksleme imkanı sunan Lucene adlı açık kaynak kodlu bir metin kütüphanesi kullanılarak veriler indekslenmiştir. Lucene kullanılarak değişiklik yapılmış standart analizörün üç farklı analizörle karşılaştırılması yapılmıştır. Verileri indekslemek için Arnavutça konuşulan üç farklı ülkedeki çok okunan beş farklı gazeteden elli farklı konuyu içeren ve belirli bir zaman aralığında toplanan verilerle test edilmiştir. Toplanmış verilerden edilen sonuçlara göre her bir analizör doğruluk düzeyine göre sıralanmış ve değerlendirilmiştir.

Arnavutça dökümanlardan elde edilen verilere dayanarak çıkarılan sonuclara göre, bu dildeki bağlaçlar eklendiğinde sistemin daha iyi sonuç verdiği ve Arnavutça verilerini diğer dillerle karşılaştırarak elde ettiğimiz verilere göre İtalyan analizörün en iyi performansı gösterdiği sonucuna varılmıştır.

Bu araştırmayla ilgili veri kümesi, başlıklar ve diğer tüm ilgili konulara https://github.com/berruqazimi/InformationRetrivealInAlbanian adresinden erişilebilir.



**Anahtar Sözcükler:** Bilgiye Erişim, Kıt Kaynak Diller, Veri Kümesi Oluşturulması, Arnavutça Dili

## ABSTRACT

NEWS INFORMATION RETRIEVAL OVER ALBANIAN LANGUAGE DOCUMENTS

Berru QAZIMI

Department of Computer Engineering
Anadolu University, Graduate School of Science, June 2018

Supervisor: Assoc. Prof. Dr. Özgür YILMAZEL

Throughout the examination of thesis we screened the Albanian language and its principles to permit information retrieval for community development and adaption of system in a specific language. We already know free text searches are highly dependent in use of language. As an outcome we modified Standard Analyzer for Albanian language where all Albanian stopwords were added inside the analyzer.

We accomplished the objective by using open source text retrieval library named Lucene which requires full text indexing afterwards, searching capability. Exploiting Lucene, the distinction of modified standard analyzer with three different analyzers were made. For indexing data we exploited our collection of data which is collected by five different most frequented newspapers in Albanian speaking countries at specific time period followed by fifty topics. As a result of retrieval, each analyzer were judged by relevance and later were evaluated.

We marked that retrieved relevant documents for Albanian results, are getting improved by adding stopword list in that language, besides we noticed that Italian analyzer performed better for retrieving Albanian documents compared with other languages.

Dataset, topics, qrels and other stuff related to this research are available at this link https://github.com/berruqazimi/InformationRetrivealInAlbanian.



**Keywords:** Information Retrieval, Resource-Scarce Languages, Dataset Creation, Albanian Language

22/06/2018

## STATEMENT OF COMPLIANCE WITH ETHICAL PRINCIPLES AND RULES

I hereby truthfully declare that this thesis is an original work prepared by me; that I have behaved in accordance with the scientific ethical principles and rules throughout the stages of preparation, data collection, analysis and presentation of my work; that I have cited the sources of all the data and information that could be obtained within the scope of this study, and included these sources in the references section; and that this study has been scanned for plagiarism with "scientific plagiarism detection program" used by Anadolu University, and that "it does not have any plagiarism" whatsoever. I also declare that, if a case contrary to my declaration is detected in my work at any time, I hereby express my consent to all the ethical and legal consequences that are involved.

.....

Berru QAZIMI

## TABLE OF CONTENTS

**ACKNOWLEDGEMENTS**

I would like to thank my advisor Assoc. Prof. Dr. Özgür YILMAZEL for his guidance and support during my study. It was my pleasure to work with him during this study.

I am also thankful to Asst. Prof. Ahmet ARSLAN for giving me some useful tips throughout the research.

Berru QAZIMI

.

## LIST OF FIGURES

**Page**

## LIST OF TABLES

**Page**

## LIST OF EQUATIONS

**Page**

## ABBREVIATIONS

| | |
|---|---|
| **IR** | : Information Retrieval |
| **TREC** | : Text Retrieval Conference |
| **TF-IDF** | : Term frequency - Inverse document frequency |
| **SMART** | : System for the Mechanical Analysis and Retrieval of Text |
| **NIST** | : National Institute of Standards and Technology |
| **MAP** | : Mean Average Precision |
| **MRR** | : Mean Reciprocal Rank |
| **P@K** | : Precision at k |
| **RFID** | : Radio Frequency Identification Devices |

## 1. INTRODUCTION

Data volume initiated to increase in both structured and unstructured data, exclusively unstructured data is matter of course in finding an appropriate information is a main concern. Reacquisition of this number and this sort of data, the Information Retrieval was created just to retrieve and to satisfy the user needs.

Mobile phones, social media, imaging technologies and lots of these create new data. The created data must be stored somewhere for some purpose [1]. Data is all around us. Everything that you do online, the clicks you make, ads you see, any billing events, video watching, every request you make to a server, transaction you perform, the network message you either send or receive, any faults which occurs in the network, or in the applications you use in your devices. All of these information could be recorded.

It represents just the tip of the iceberg in terms of data that could be collected and afterwards analyzed. Big Data also comes from content generated by user and this may be on the web and mobile applications, from Facebook, Instagram, Twitter, or YouTube. These applications have user generated content, like posts, videos, and pictures.

Another source of Big Data is health and scientific computing. In scientific computing, the large hadron collider is generating petabytes of data. Similarly, in protein applications, we are generating gigabytes, terabytes, and petabytes of data. Graph data also is a source of Big Data and involves too many interesting data type that has a graph structure. For instance social networks, our friend's relationships, telecommunication networks, computer networks, collaborations, and relationships. Some of these graphs can be absolutely huge. If let's think about Facebook user graph, all of those friend's relationships across billions of users indicates us the how huge the data size is.

In addition the sources of Big Data are web server machines which generate plenty of information. Every interaction we do with an application on the machine generates a record in a file.

Furthermore source is RFID tags. Some models are in countries like Europe and USA they use RFID tags to pay tolls. It is fast track electronic toll collection system. All of these make big data world.

On the other hand, we have unstructured data. Examples of this are plain text and media. This type of data does not have a schema also there is no types associated with the data. In the middle, we have semi-structured data. This includes items like documents,

extended markup language documents, tag texts, and media. With these type of data we may be able to infer the types of information stored in specific file. Structured data [7] has its pre-defined model of data. For instance, relational databases and formatted messages.

Unstructured data neither have pre-defined model nor organized specific structure. In contrast unstructured data information role-plays its significance on language dependence. As a result of rapid escalation, IR system grew into a great boost momentum. There are plenty source of information retrieval libraries available for each with a different feature [13].

IR is set up a collection of data for retrieval using indexing process. During indexing process archives are tokenized into words and a file of inverted index is created. The record has a rundown of each ordering highlight demonstrating the reports that contains it. There are extra advances that may happen during either at indexing and retrieval. One of them are stopwords which are the words Stopwords are words with no prejudicial esteem. If these values are discarded during indexing the goal is achieved so it spares space and speeds up the process.

Collected data text documents in Albanian language involves top 5 newspapers in Albanian speaking countries such as: Albania, Macedonia and Kosovo.

In comparison, the retrieval performance and its effectivity in a different language analyzers, choices are made by the cause of similarities that exists in a target languages and in our modified standard analyzer. The exclusive reason why the modified this analyzer is because Lucene does not have a specific analyzer for Albanian language and Albanians usually use English for retrieving information.

We have evaluated the effectiveness of each approach in our text collection with distinct measurements based on relevancy of retrieved document.

### 1.1. Objectives

The main objective of this work is to study Lucene[1] technology for both indexing and searching techniques of text data and implement them for 2000 xml files form our data collection. Our Collection consists data collected from 5 most frequently visited

[1]https://lucene.apache.org/

newspapers in Albanian regions. Collected data in this set contains only textual information.

The purpose of this study is to reveal the best language for retrieving Albanian documents and for exploring improvement of the system only by adding stopwords in a proper language.

This study compares Lucene analyzers with each other afterwards, it compares all of these with our modified standard analyzer. The aim is to analyze the retrieval performance of documents in Albanian. In addition, we wanted to imply the significance of language in information retrieval and how we used our data collection for built-in analyzers and our modified standard analyzer.

Outputs of this work include: This paper describes the theoretical background of Information Retrieval systems and technical notes, a text index of our data collection including removal of frequently used words in a language and suitable foreign language for retrieval.

### 1.2. Thesis Outline

The organization of entire thesis is along these lines. The second section gives summary about the usage of grammar in Albanian language and also gives some extra information about the language in general. Further, presents the detailed information of our collected data, sample of indexed documents, queries, structure of information needs and system overview for both indexing and searching processes. The next chapter gives general concepts of Information Retrieval history, strategy and implementation.

Lucene chapter introduces this technology starting from architecture, to its most crucial processes indexing and searching and the description of the tasks that have to be undertaken during the Information Retrieval. The fifth chapter presents results of our experiments by using different measures for all pre-defined analyzers.

The last chapter summarizes the outcomes of the whole thesis, putting emphasis in the most important aspects of our study, and concludes the findings. The thesis ends by encompassing remarks and future work in the field of Information Retrieval.

### 1.3. Literature Review

Information need is something that has become an important issue in today's world. Retrieving the relevant information for distinct languages and to acquire good outcome is also important manner.

On the other hand, Information Retrieval has a long history and evaluation is in the center field of this approach starting form beginning years of its development. There are many algorithms, models and systems in literature in order to choose most efficient one among many, evaluation is needed. Christopher D. Manning [8] who is the main source of our study states that for measuring ad hoc IR effectiveness in a standard way there is need for collection consisting three things: Data collection, test suite of information needs and set of relevance judgments (binary assessments for relevant and non-relevant for each document-query pair). After fulfilling these ingredients the evaluation can be done in collection with two most frequent and basic measures of IR precision and recall.

The human language contains function words that have less meaning in a document comparison with others. Those words are undivided parts of our texts. As W. Bruce Croft [15] states those words should be treated in a special way in text processing because of two properties. Fist, these function words are extremely common and second these words rarely indicate anything about document relevance on their own. Usually these words are called stopwords the reason is because text processing stops when one of them is occurred.

With a specific end goal to assess our calculation, we required an accumulation of records (writings) in Albanian. Since we couldn't locate any current corpus-based computational phonetics asset for the language, we chose to make our own particular one. Hence, we gathered documents from the Internet.

Thus, Albanian language is not widely spoken language, we do not have too many resources in this area. But, several researches are made by Nikitas N. Karanikolas.

In his study Nikitas N. Karanikolas [14], summarizes the concept of creating stems and evaluation of the stemmers against actual stemmer. He revealed that this approach gave quite well improvement, caused only by using a subset of the experts.

In addition, Nikitas N. Karanikolas [10] study briefs rule-base stemmer. Stemmers are modules that utilize text processing tasks like classifying and summarizing

documents. He created simple rule-based stemmer for the Albanian language. Their ability is replacing suffixes using some conditions.

In this paper we contribute on the field of text retrieval, especially by improving relevant retrieved documents in Albanian. With this study we achieved better results for some measurements. As well as it figures out nearest similar language for retrieving Albanian texts.

### 1.4. Method

The examination includes mix of both qualitative and quantitative research techniques. First, to ensure the reliability of data in our collection we chose to collect data form different newspapers especially from those Albanian speaking countries, labels containing various topics in such a long amount of time. After the accomplishment of this task, we chose full-text search platform called Lucene for indexing and searching through our data. This platform has built-in analyzers and their responsibility is to covert given data into its fundamental representation called tokens [6]. Their main task is to deal with grammar and vocabulary parts of language.

In Lucene each language has its own analyzer, unfortunately Albanian does not have one. Based on this fact we used languages similar to Albanian for retrieving text from our data collection throughout our research. Languages are selected based on their similarities to Albanian language. The aim is to encounter the optimal built-in analyzer for retrieving Albanian documents.

However, based on Bruce Croft [14] most commonly used words or stopwords are words that do not have any influence on performance of retrieving relevant documents. Taking advantage of this information we modified standard analyzer by adding Albanian stopwords in it then, performance of this analyzer is compared with other selected analyzers.

At the same time the used stopword list is assumed and presented to be changed in future. Using stopwords in Figure 1.1. indexing and searching processes are made in our corpus. While we have some extraordinary letters like Ç and Ë we will compose the two adaptations of words contain these letters as presented in Figure 1.1.

| Albanian Language Stopwords | | | | |
|---|---|---|---|---|
| a | i | kemi | pasi | të/te |
| apo | jam | këtë/kete | për/per | ti |
| asnjë/asnje | janë/jane | mes | prej | tek |
| ata | jemi | më / me | që/qe | tij |
| ato | jeni | mu | sa | tonë/tone |
| ca | ju | në/ne | së/se | tuaj |
| deri | juaj | nëse/nese | se | ty |
| dhe | kam | një/nje | seç/sec | tyre |
| do | kaq | nuk | si | unë/une |
| e | ke | pa | saj | veç/vec |

**Figure 1.1.** *Stopword list*

In our approach for making a stopword list first we curled through distinct sources on the web. Thus, Albanian is not widely used language and it is hard to make a giant corpus.

Besides, there are a few words that are not extremely frequent but they are totally unfit to segregate amongst documents. Therefore, these words cannot be expelled by a measurable approach. On the other hand, as language resource we used newspapers in Albanian speaking countries. We have considered the following newspapers:

- Telegrafi is one of most frequently visited online portals in these areas founded on 2006 in Kosovo, is the leading portal in Albanian language. It is the dominant web site in Kosovo, Albania, Macedonia, Montenegro, Presevo Valley, and wherever Albanians live. In 2011/2012 has won the award of Super brands Kosovo's Choice.
- GazetaExpress is a portal owned by MediaWorks founded on 2005 in Kosovo by Berat Buzhala, Petrit Selimi, Dukagjin Gorani, Ilir Mirena, Astrit Gashi, Arlinda Desku, Andrew Testa, Gjergj Filipaj, Bul Salihu. GazetaExpress is the leading Portal in the Balkans. Express brings to her readers the latest news from Kosovo, Albania, Macedonia and all around the World.
- TetovaSot is an independent Albanian-language portal, based in Tetovo, founded in 2013.
- Tetova1 is news portal based in Tetovo founded in 2011 which delivers news mostly from Macedonia especially Tetovo.
- OraInfo is an informative portal in Kosovo founded in 2012 which brings news mainly from Kosovo.

We have gathered a sum of 2000 documents from the above areas. This gathering of 2000 documents involves our (little) corpus. The subjects of documents fluctuate

between various fields like economy, politics, sport, magazine and technology. Mostly taken form fields of politics and economy.

The sample file consists DOC field which is the root element and DOCID which is an identifier of each content. Its structure is shown in Figure 1.2.

```
<DOC>
<DOCNO>Number of document </DOCNO>
<SOURCE>Name of the newspaper</SOURCE>
<URL>URL of newspaper</URL>
<TITLE>Headline of the content</TITLE>
<TEXT>The content </TEXT>
<DATE>Curled date</DATE>
</DOC>
```

**Figure 1.2.** *Structure of files*

The entire study is done by making relevance judgments of every retrieved document based on created information needs. The relevance judgments are a crucial piece of a test collections which helps us evaluating them. The meaning of relevance is then you are composing a write about some subject and would utilize the data contained in the collection or in the report, at that point when the record is important or fulfills user's needs.

First the binary judgments are made for evaluation the document using 1 to relevant document and 0 to non-relevant one .Afterwards the record is judged as significant if any bit of it is applicable without paying attention how much relevant information contains the document.

The method of enhancing retrieval effectiveness considering relevance judgments provided by user is called relevance feedback. In the system users make a query then the system retrieves ranked list achieves based on its judgments. System can modify the information need based on its relevance.

Later on the query can be executed and a new ranked list will be created. Relevance feedback ordinarily gives an enhanced ranking in contrast with retrieval based on given query. The Table 1.1. shows qrels example of Spanish analyzer for first query where first column is query id, second one is document id and last column is its relevancy.

**Table 1.1.** *Qrels*

| Query | Document ID | Relevance |
|---|---|---|
| 1 | 829 | 0 |
| 1 | 1202 | 0 |
| 1 | 309 | 1 |
| 1 | 338 | 0 |
| 1 | 1808 | 1 |
| 1 | 1776 | 1 |
| 1 | 814 | 1 |
| 1 | 1205 | 1 |
| 1 | 882 | 1 |
| 1 | 503 | 0 |
| 1 | 488 | 1 |
| 1 | 826 | 1 |
| 1 | 282 | 1 |
| 1 | 635 | 0 |
| 1 | 783 | 1 |

The disk space requirement for our entire collection which consists 10 MB XML data, and other index files generated by Lucene. These numbers are slightly smaller compared to other collection sizes. The reason is because we do not have collection in Albanian as we motioned before we created our own. Our objective is to index the XML files that store data. The statistics state that the number of words are between 20 and 2940 per file and their size varies from 1Kb to 19Kb.

In the process of adding stopwords we have used a CharArrayset class of Apache Lucene to add stopwords to the standard analyzer which is a simple class that stores Strings as char's in a hash table. It cannot expel things from the set, nor does it resize its hash table to be smaller. Its aim is to test the presence of char in a set without converting it into String.

```
org.apache.lucene.analysis.CharArraySet stopset=
CharArraySet.copy(StandardAnalyzer.STOP_WORDS_SET);
stopset.add();
```

**Figure 1.3.** *Method of adding*

All this is done by utilizing add method. This method is utilized for adding a component to the CharArrayset. The Figure 1.3. shows the utilized method for adding stopwords of Albanian mentioned in Figure 1.1.

To implement the search activity into the pointed collection containing the file, first the parameter should passed to the Index Reader class. Thus the information is static in our collection the user can perform only read in index. During the searching process we

used term query for retrieving data through our collection. After acquiring the query, it is parsed and then goes into the searcher. Afterwards, the searcher looks through the record and returns the highest ranked documents based on its information need. The search process is done by using term query.

In addition, during the evaluation the preferred measure plays significant role in effectivity of the system. Finlay we would analyze the collected results then, use them in order to answer this research question.

### 1.4. Importance of Weighting

Reflection of its duty is a frequency of the term, measurement of IR in which people refer to, is a collection frequency of the term. In this way, the collection frequency of the term is an aggregate number of times specified word shows up in the corpus. This measurement is often used to build a unigram language models or deals with spam classifiers. Normally is not used in IR ranking systems. The reason is explained down below in a Table 1.2.

**Table 1.2.** *Weighting in IR*

| Word | Collection Frequency | Document Frequency |
|---|---|---|
| punë | 11753 | 3911 |
| shkoj | 11832 | 8923 |

Words such as “work” in Albanian “punë” and “go” in Albanian “shkoj”. Both of them have virtually identical collection frequency and both occur more than eleven thousand times in a corpus. Although, the document frequency in the word “shkoj” is broadly dispersed crosswise over documents. The word “punë” tends to occur 2 up to 3 times in document whether we incline to search for word “punë” we don’t want to include all documents consists the word “shkoj". This means we give higher weighting to instances of the word “punë”.

Making query “I go to work” in Albanian “Unë shkoj në punë”. The most important word query need is to find and to retrieve documents in a certain collection containing “punë”, the second in command is the word “unë” next, comes the verb “shkoj” before the stopword “në”.

Seems like weighting is captured by looking at Document frequency and not captured by Collection frequency which would score” punë” and “shkoj” equally.

## 2. ALBANIAN LANGUAGE IN GENERAL

Albanian[2] or "Shqip" in local language is spoken by 5 million native speakers and 10 million people around the world. It is an independent branch of Indo-European family primarily spoken in Albania, Kosovo, Macedonia and Montenegro which is official language inside these borders however, it is spoken in other areas like Southern Europe venues where Albanian populations live Preševo Valley in Southern Serbia and Epirus in Northern Greece. There is two characteristical Albanian dialects named Gheg which is spoken in the northern and the other one Tosk which is spoken in the southern Europe. They both have been diverging for at least a millennium. Distinctions [10] were made in the enhancement of vowel sounds in Gheg, in comparison to the Tosk. The existence of nasal vocals in Gheg is not characteristic in Tosk.

Communities who speak Albanian [1]dialects can be found in Italy called Arbëreshë, in Croatia (Arbanasi), Romania and Ukraine. Albanian is not a language scrutinized in IR system. Is based on Latin alphabet but presents exceptional imprints for a few letters.

Albanian alphabet has total 36 letters, the letters in alphabetical order are: a, b, c, ç, d, dh, e, ë, f, g, gj, h, i, j, k, l, ll, m, n, nj, o, p, q, r, rr, s, sh, t, th, u, v, x, xh, y, z, zh. Of which 6 letters vowels (a, e, ë, I, o, u, y) and other 30 letters are consonants. It uses 25 letters out of 26 letters in English, only the letter W is out of usage. In exclusion it uses letters Ë and Ç besides, it includes the so called couple letters and when they are gathered together, the way of using is as single letter. Total number of this kind of letters are 9 in Albanian language. Shown in Table 2.1.

**Table 2.1.** *Couple letters used as one letter*

| Letters |
|---|
| dh |
| gj |
| ll |
| nj |
| rr |
| sh |
| th |
| xh |
| zh |

[2]https://www.wikipedia.org/

Albanian [9] language does not have defined page code but is uses ISO-8859 part 16 which covers south eastern European languages like Albanian, Italian, Romanian, Hungarian Croatian Polish and Slovenian.

In Albanian [10] grammar nouns have three genders (masculine, feminine and neuter). There are 6 cases (nominative, accusative, genitive, dative, ablative and vocative) in which vocative is not broadly utilized. Cases are utilized as a part of both kind of nouns defined and undefined. The article is placed after the noun and it can be in type of suffixes, which fluctuate with sex and case. Verbs in Albanian language has six types of modes consisting 8 tenses which are five complex and three simple plus two voices active and passive. A verb has bounty number of varieties to help sentence structure leads in the dialect.

In this thesis we used Albanian stopwords for analyzing the relevance of documents. Stopwords in Albanian language are: a, apo, asnjë , asnje, ata, ato, ca, deri, dhe, do, e, i, jam janë ,jane jemi, jeni, ju, juaj, kam, kaq, ke, kemi, kete, këtë, më , me, mu, në , ne nëse, nese, një, nje, nuk, pa, pas, pasi, për, per, prej, që, qe, sa, së, se, seç , sec, si, saj, të, te, ti, tek, tij, tone, tone, tuaj, ty, tyre, unë, une, veç ,vec.

As we mentioned earlier Albanian alphabet have some special letters like Ç and Ë. In our application both versions of words contain these letters are used.

All information needs, dataset and qrels of analyzed languages are published in Github[3] which is web based hosting service mostly used for computer code.

### 2.1. Test Collection

In this study we made a collection of new stories grabbed form five newspapers in Albanian speaking countries from 2017-03-05 till 2017-12-16. The newspapers are Telegrafi[4], Tetova1[5], TetovaSot[6], GazetaExpress[7] and OraInfo[8]. The collection consists 2000 documents.

[3]https://github.com/

[4]https://telegrafi.com/

[5]http://www.tetova1.com/

[6]http://www.tetovasot.com/

[7]http://www.gazetaexpress.com/

[8]http://orainfo.net/

### 2.2. Documents

Documents in our collection consists of six fields: DOCNO, source, URL, title, text and date. An example of collection is given in Figure 2.1. Among this fields we will search only text field and get textual information [17]. In our retrieval system we used DOCNO as unique identifier and text as textual field.

<DOC>
<DOCNO>**977**</DOCNO>
<SOURCE>**TetovaSot**</SOURCE>
<URL>**http://www.tetovasot.com/2017/03/nisi-mbledhja-e-kuvendit-a-do zgjidhet-talat-xhaferi-kryeparlamentar-video/**</URL>
<TITLE>**Nisi mbledhja e kuvendit, a do zgjidhet Talat Xhaferi kryeparlamentar?**
</TITLE>
<TEXT>**Sapo kanë nisur të vin deputetët për të mbajtur seancën e radhës ku pritet të zgjidhet kryeparlamentari.**
**Deri tani i vetmi kandidat është Talat Xhaferi shqiptari i parë i cili do të ulet në karigen e parlamentit të Maqedonisë**</TEXT>
<DATE>**2017-03-27**</DATE>
</DOC>

**Figure 2.1.** *Document 977 in our collection*

### 2.3. Information Needs

In TREC each information need is referred to a topic. Information needs in our collection consists three parts the title, description and narrative similar to TREC topics. Title field consists short query, description field describes a content in a few words and narrative part gives an extra information about topic [17]. An example of information need in our collection is shown below on Figure 2.2. Topics consist a specific data structure and queries, tested through usage of excel formulas.

<top>
<QID>**1**</QID>
<Title>**Incidentet në Tetovë**</Title>
<Description> **Incidentet që ndodhin në Tetovë dhe rrethinë**</Description>
<Narrative>**Këtu përfshihen të gjithë incidentet që ndodhin në Tetovë rahjet, plagosjet dhe vjedhjet. Duke përfshirë të gjitha detajet e tyre nga ndodhja e rrastit deri te zbardhja e tij.**<Narrative>
</top>

**Figure 2.2.** *Information need 1 in our collection*

## 2.4. Relevance Judgments

Relevance judgments are correct answers of information where documents are relevant for specific information need. For information need and document you need to create a relevance assessment. Thus, evaluating all retrieved documents form collection requires Information Retrieval experts' judgments which takes time and is an expensive process [8].

## 2.5. System Overview

Figure 2.3. summarizes the system of indexing and Figure 2.4. summarizes querying and searching process of our collection using 4 analyzers for retrieval in Albanian documents.

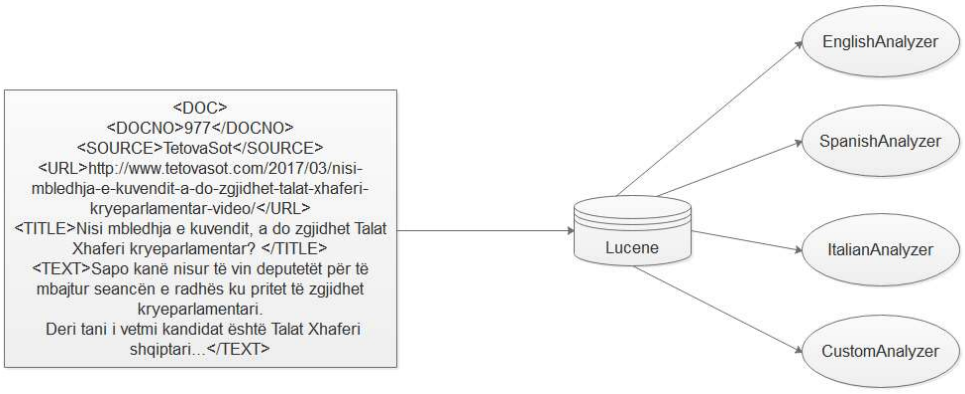


**Figure 2.3.** *Indexing*

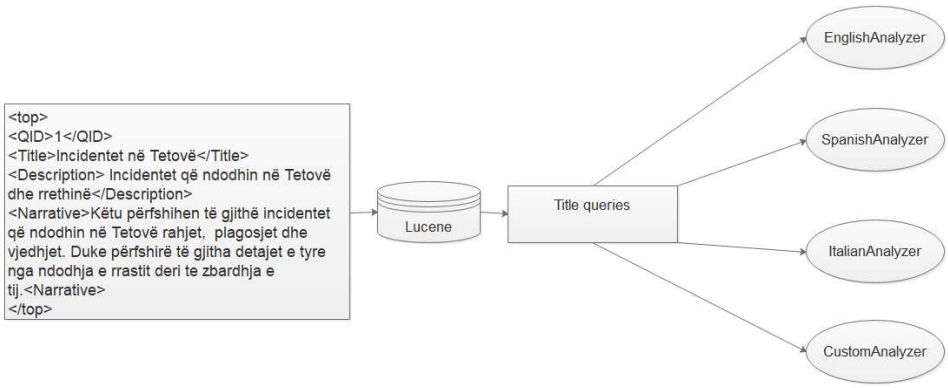


**Figure 2.4.** *Searching*

## 3. INFORMATION RETRIEVAL

The term information retrieval [5] alludes to an inquiry that may cover any type of data: structured data, content, video, picture, sound scores and numerous more data. In this section we will depict the IR history, systems, and existing executions of those techniques. An IR system regularly finds data that is significant to client's inquiry. It has capacity to seek over accumulations of structured and unstructured information. The need of the system happens when accumulation achieves the size where customary inventorying procedures can no longer cope.

### 3.1. Information Retrieval History

Long time ago people have been realized that archiving and finding information is significant. The principal strategy of doing data information retrieval is known as brute force, taking a gander at each item in dataset and deciding whatever it fulfills the required data.

First advanced technique appeared in libraries. The attempt in this technique was to locate the documents by some criteria. Numerous catalogues were built to go about as access focuses to the accumulations, and to encourage the retrieval process.

The IR field, as it is today, was conceived in the 1950s out of this need. It was after 1945 when an American engineer called Vannevar Bush [3] published an article titled *As We May Think*. After this event first automated information retrieval system introduced later on.

After several researches were made in information retrieval area .The most significant one came from Hans Peter Luhn [3] in 1957. His study was KWIC (Key Words In Context) which is a type of word index where each occurrence of the keyword is shown together with encompassing words in a list of string. Then later he also worked with techniques like full-text processing, hash codes, and auto-indexing.

Afterwards in 1960 a German engineer and professor in Cornell University Gerard Salton [3], with his students, developed the SMART (System of Mechanical Analysis and Retrieval of Text) in Harvard University. This SMART system allowed researchers to experiment with ideas which improved the search quality. The TF-IDF (term frequency-inverse document frequency) model for scoring documents has also introduced by him.

In the 1970s the researches formalized the retrieval process, the main role acted by Gerard Salton who gathered materials of his group on vectors to produce the vector space model [4]. This way to deal with retrieval process consolidated numerous search systems and a ton of research in coming two decades..

In 1992 with cooperation of US Government and NIST (the National Institute of Standards and Technology) the TREC is organized, the aim of TREC is to encourage research of IR Systems from large text collections [3]. With TREC some new techniques are developed to do effective retrieval over large documents. When the search engines came in 1990s increased the need for searching in large collection of data. They become most common of information retrieval models, search and implementation. They now are used every day by a million of people all around the world. After some period of time there is need for other requirements including the need for search in non-textual data like pictures, video and audio files.

### 3.2. Information Retrieval Strategies

After history we will present thoroughly the most important strategies and models over the years. It will refer to some strategies and models in Information Retrieval systems and to mention most important ones in the field.

Information retrieval strategy [2] is an calculation in which a query Q and set of documents$D_1$,$D_2$,…$D_m$ distinguishes likeness between finite sample sets (Q,$D_i$) for each documents $1 \leq i \leq n$. Later on we will list some strategies that exits to identify how these documents are ranked.

#### 3.2.1. Boolean model

Boolean model [6] is first IR model. In the inside of this model the user must specify the information need by using specific query in normal form. This method is often used in search engines on the Internet hence, it is fast and therefore can be used online. Since the model coefficient in pure Boolean is 0 or 1 records will either fulfill or don't the provided query, there are no importance scores related with these records and the documents are unordered.

### 3.2.2. Vector Space model

Vector space model [6] is algebraic model where documents and queries are represented as vector in high dimensional vector space where each unique term is dimensional and the relevance of information is calculated by distances between vectors. If it occurs in the document its values are non-zero. Way of computing these values are weights and best noun is IT-IDF weighting.

### 3.2.3. Probabilistic model

Probabilistic model [6] is used to compare probability between documents and query match using a full probabilistic approach. It is a bit of all documents that is favored by the user as the appropriate response set for query q, if we consider set R as perfect answer we expand the general possibility of relevance to that user. The expectation of these documents in set R are important to the query, else they are non-applicable.

### 3.2.4. Metric Space model

Metric space is another field which concerns in terms of weighting. Metric space is general model, but still used for indexing in IR system. This model treats the dataset as unstructured data objects with its distance for every object. Distances taken all together are called metric on a specific set. No additional data about the items' inward structure or properties are required. The created function [8] simply measures the comparability or uniqueness of any two objects and it can depend just on the different metric hypothesizes.

## 3.3. Evaluating the Performance of IR

An IR system [12] usually returns a ranked list of the documents to user's query. Widely used measure in IR system is relevance-based measure of recall and precision. Considering the query entire space of documents is divided in four sets which are: Relevant and retrieved, relevant not retrieved, not relevant retrieved and not relevant not retrieved. Those formulate the evaluation measurements of IR systems Recall and Precision Figure 3.2.

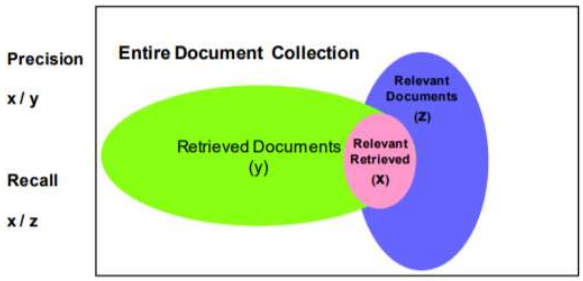


**Figure 3.2.** *Recall and Precision [8]*

$$\text{Recall}=\frac{number\ of\ retrieved\ relevant\ documents}{total\ number\ of\ relevant\ documents} \tag{3.1}$$

$$\text{Precision}=\frac{number\ of\ retrieved\ relevant\ documents}{total\ number\ of\ retrieved\ documents} \tag{3.2}$$

Recall is fraction of relevant documents from entire corpus however, Precision is fraction of retrieved relevant documents. An additional measure rarely used is fallout, besides it examines the probability of queries' non-relevant retrieved documents. Another additional measurement is F measure, which stands as weighted harmonic mean of precision and recall.

$$\text{Fallout}=\frac{number\ of non\ retrieved\ relevant\ documents}{total\ number\ of\ non\ relevant\ documents} \tag{3.3}$$

A related approach [8] has been used more frequently in recent times is (MAP), where the accuracy is estimated at each point where an relevant record is gotten and then taking average over every single document to come up with the precision for a given query.

# 4. LUCENE

## 4.1. Introduction to Lucene

Apache Lucene is highly performed text retrieval library .It is purely written in java and is a single Jar file with a small size, with no dependencies can be integrated from the simplest java stand-alone console program to the most advanced application that requires full-text search especially cross-platform.

It is free and open source platform which is written by Doug Cutting and now supported by Apache Foundation. Has several built-in analyzers which are able to compound words, spell correction and case sensitivity.

Lucene has ability to index any data and make it searchable. It can index and search data stored in all files like simple text, word document, HTML documents, XML documents and many more files.

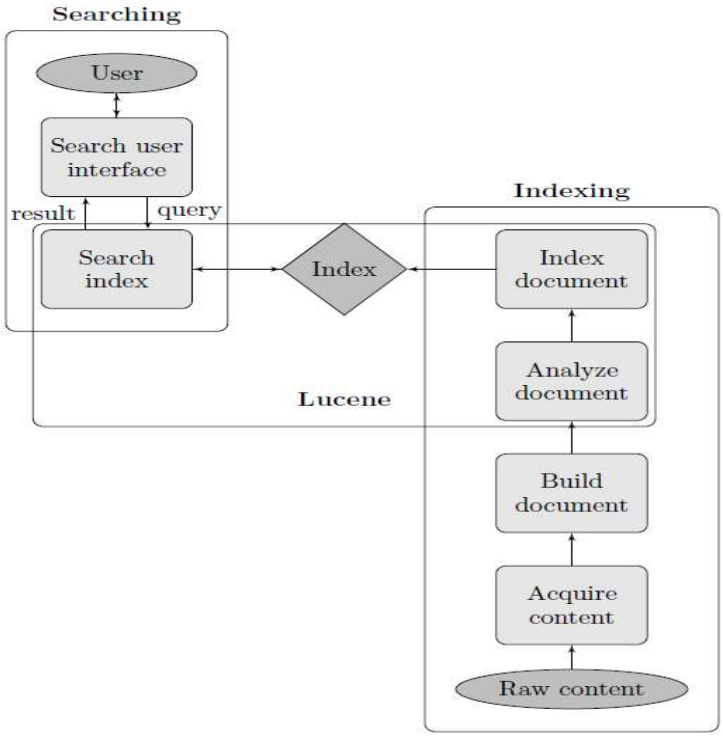


**Figure 4.1.** *Data flow in Lucene [6]*

Each sequence of the steps is important in searching. The Figure 4.1. shows the steps how data flows while using the open source library of Lucene. For better

understanding of this flow chart, we will explain deeply the following parts such as indexing, searching and analyzing process.

### 4.2. Indexing

Indexing is core function provided by Lucene. Index [6] is actually processing information into exceptionally effective cross-reference lookup in order to facilitate rapid searching. Indexing is conversion process and its output is called index. Lucene is like database, it stores information and retrieves it later. But there are some differences between traditional databases and Lucene even though, they retrieve information need. The Distinctive feature between Lucene and relational databases is schema. Databases have schema at which per every table have constrains, domains and name for attributes. In Lucene a single index can hold documents that represent different entities.

Another difference between relational databases and Lucene while indexing does not permit neither recursion nor nesting, so content must be flattened. Relational databases usually have arbitrary number of joins with both keys primary and secondary, linking the tables with each other using concept named foreign key. Occurred recursions and joins must be denormalized while indexing although, the result of content might be replicated so many times. This never occurs in Lucene because while indexing it denormalizes the content.

#### 4.2.1. Inverted index

To find the term as fast as possible in specific documents, an efficient cross-reference lookup is needed. Fist, Lucene scans the entire data collection for specific information need which is identified as unique terms. Inverted index is processes of converting free text into its most fundamental index that is represented by building the structure [6].

A set of unique terms usually refers to dictionary. For each term in dictionary a unique posting list is created. The posting list actually is a list of records which contains the term and the position of the document.

Structure is inverted because it uses unique terms from input documents as lookup keys inserted on using documents as central entities. Inverted index is same model like content's page in books, you look for specific term in a table when the term occurs, and you directly jump into corresponding page.

### 4.3. Analyzers

Task of the analyzer is to convert given data into its fundamental indexed representation called token [6]. Whole process starts with tokenizer and carries on with series of tokens. Each token carries itself text value and some metadata. Combining tokens with their associated field name we get terms. Its responsibility is to deal with problems like spelling, synonyms, compound words, case sensitivity. Keeping in mind that resulting tokens really depend by the analyzer. Analyzer has full control over tokens extracted while analyzing and next tokens are inserted into index, only indexed tokens became searchable.

Creating built-in analyzer in Lucene is simple, at first you need to import analyzer and then create it. In Lucene version 6.6 this is how you create an analyzer Figure 4.2. Also IndexWriterConfig holds all configuration used to create an index.

```
Analyzer analyzer = new StandardAnalyzer();
IndexWriterConfig iwc = new IndexWriterConfig(analyzer);
```

**Figure 4.2.** *Creating Analyzer*

Lucene library provides core built-in analyzers and also gives you an opportunity to customize your own analyzer by extending analyzer class. As we mentioned before Lucene has built-in analyzers and the description will appear in detail.

#### 4.3.1. Standard analyzer

Standard Analyzer is Lucene's most sophisticated core analyzer. Its ability is to recognize patterns of company names, hostnames and e-mail addresses, also provides grammar based tokenization and works well for most languages. This analyzer applies logic inside tokens, removes stopwords, punctuation and lowercases.

#### 4.3.2. Whitespace analyzer

Whitespace Analyzer neither makes any normalization in tokens nor any tasks like lowercase, stopwords removal. It only performs a single operation likewise splits the text into tokens on whitespace characters.

### 4.3.3. Stop analyzer

Stop Analyzer does the same job as Standard Analyzer except it removes common words. The analyzer removes common words in English language. Removed common English words with this analyzer are: a, an, and, are, as, at, be, but, by, for, if, in, into, is, it, no, not, of, on, or, such, that, the, their, then, there, these, they, this, to, was, will, with.

Using this type of analyzer, we can decrease the size of index but it may result to a negative impact on precision.

### 4.3.4. Simple analyzer

Simple Analyzer like its name is the simplest one. This analyzer lowercases each token. Compared to whitespace it uses non-alphabetic characters instead of whitespace. This means that the numeric characters are not included.

### 4.3.5. Keyword analyzer

This analyzer does not take any operation in specific test, only considers it like a single token. Keyword Analyzer is usually used when we have a special field and those fields must be untouched, as an example of these are identifiers.

## 4.4. Searching

Searching [6] is a process of looking up to indexed words and retrieve it based on given criteria. There are measurements which describes quality of search using precision and recall. Precision is the number of retrieved documents and recall is the number of retrieved relevant documents.

An ideal system will retrieve only relevant documents however, we do not have perfect system which will retrieve only relevant documents since the relevance relies upon subjective supposition of the user.

Standard search method walks over indexed documents and accepts those which fulfill the criteria of the given query and retrieves them. To obtain better results, we have to have a well-prepared index. In Lucene the indexed documents usually have bunch of fields. During the search, we have to be explicit about our aim. When we make a query our search will retrieve only indexed and stored documents instead of indexed and not stored ones.

### 4.5. Queries

Dealing with the user's query is primary task of actual search engines. Queries may be simple with single term, very rich and complex depending on user's information needs.

Lucene [20] provides some built-in queries for searching through indexed documents. These queries include searching in specific term, by phase, by prefix, by range and by fuzzy term matching. Lucene offers standard syntax which users are familiar with. Syntax is shown in Table 4.1.

Lucene has some characters with special functions. If we want to use them in a query, we need to use the backslash character (\). The characters that require escaping are as follows: +: -! \ ( ) ^] { } ~* ?

**Table 4.1.** *Query types in Lucene*

| Search | Syntax |
|---|---|
| Single Term | Information |
| Phrases | "Information Retrieval" |
| Fields | filed_name:"Information Retrieval" |
| Fuzzy | roam ~ |
| Proximity | "Albanian retrieval" ~ 9 |
| Range | field_name:[1 TO 100], field_name:{Berru TO Qazimi} |
| Wildcard | te?t, test* |
| Boosting | Albanian ^4 information |
| Boolean | AND, OR, NOT, +, - |

### 4.6. Scoring

When document is matched while searching Lucene computes numeric value of relevance called score and assigns it into documents. Taking high score means good match, and stronger similarity which corresponds to a better results. Lucene utilizes: Boolean model, vector space model and probabilistic model. This approach combines vector space model and Boolean model, and during the search it lets you to choose between these two models.

The similarity scoring formula [20] uses TF-IDF to measure similarity between query and documents.

Lucene's similarity formula is computed for each document (d) in a query (q) that matches its term (t).

$$score(q,d) = term(q,d) * qNorm\ (q,d)$$
$$\sum(tf\ (t \in d) * idf(t)^2 * boost(t.field \in d) * lenNorm(t.field \in \text{d})) \quad (4.1)$$

## 5. EXPERIMENTS AND RESULTS

The aim is to encounter the optimal language for retrieving Albanian documents and how the retrieved documents are improved by using Albanian language stopwords.

Leading any IR test the utilization of a corpus must be arranged and choices should concern the trial outline [16]. The objectives of the assessment must be characterized, an appropriate test collection must be chosen from those in presence, or one must be made particularly for the issue being tended to either various systems or techniques produced for analyzing.

It is normal to compare retrieved results against each other (competitive evaluation) where the most noteworthy scores are most relevant ones. Furthermore, the collection of results created by running a similar systems in a various circumstances with distinct parameter settings from distinct systems are utilized to compare them with each other. Usually the utilized measures are calculated over numerous queries and averaged to fabricate the final result of evaluation.

The retrieval of documents are done by using open source IR library Apache Lucene. However, the Albanian does not have default analyzer and the comparison is made through four built-in analyzers such as English, Spanish, and Italian which have similarities with Albanian language and our modified standard analyzer.

Two steps that have a main role in performance of IR are stems and stopwords. Stopwords are commonly used words like dhe, në, sa in Albanian language which does not affect in efficiency of retrieval and slows down the retrieval process.

The second one is stemming known as process of removing inflectional suffixes from words in order to bring them in their base form. Like words shkollës, shkollat, shkollave, shkollën, shkolla to shkollë which means school in English, escuela in Spanish, scuola in Italian.

In this part we introduce the measures used for evaluation of test collections that are utilized for this thesis. The concept of importance in documents and the formal appraisal logic has been created for surveying unranked systems. This consolidates clearing up the sorts of evaluation measures that are standardly used for retrieval documents and related endeavors like fitting and content characterization.

The key utility measure is user satisfaction .Speed of indexing and the extent of the record are factors in user satisfaction. It has all the earmarks of being sensible to

acknowledge that the most crucial variables like quick making and pointless answers which do not satisfy user needs. In any case, user recognitions do not generally match with system designers' ideas.

Preferred measurement has an important role thus, it directly affects the effectiveness and efficacy of the system in Information Retrieval. Many old traditional systems usually use [18, 19] precision and recall as a measure. Precision measures relevant retrieved documents and recall measures relevant documents in dataset. In analyzes we used MAP, P@K, MRR for our datasets. MAP (Mean Average Precision) is average precision of each P@K where precision scores are calculated for each query. In this thesis we calculated MAP for top 20 retrieved documents and this metrics formula is as bellow.

$$MAP = \frac{\sum_{q=1}^{Q} AveP(q)}{Q} \tag{5.1}$$

Computed precision at some specified rank is called P@K or Precision at K. For analysis we used K on top 5, 10, 15, up to 20 retrieved documents and the reason is because users are really interested to find the relevant information on top two pages of their search so, they want to get information they need as soon as they can.

Queries that did not retrieve enough documents for given number of K are not calculated, for instance if query retrieved 17 documents we compute only P@15. The results are shown in Figures 5.1.-5.5.

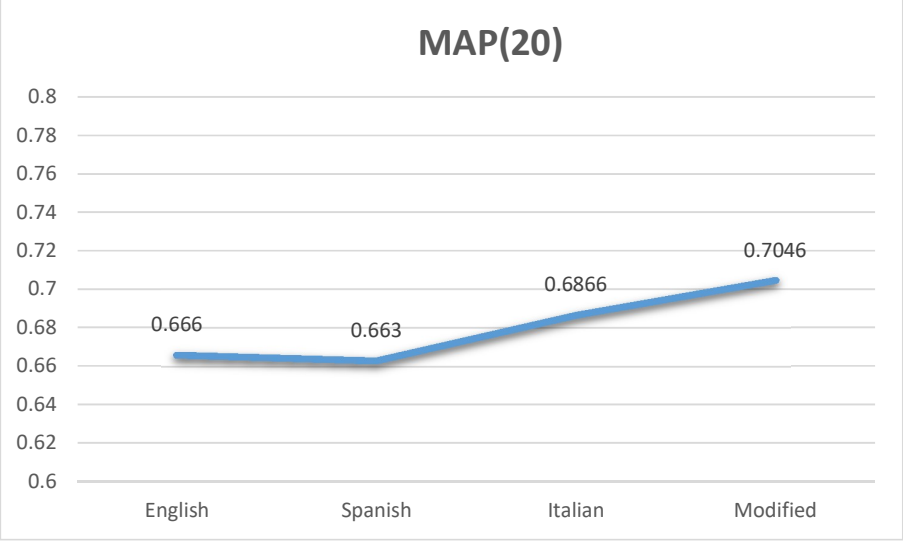


**Figure 5.1.** *Mean Average Precision for top 20*

The most effective results are retrieved by our modified and the worst preformed one is English language for MAP evaluation measurement based on top 20 retrieved documents (MAP20).

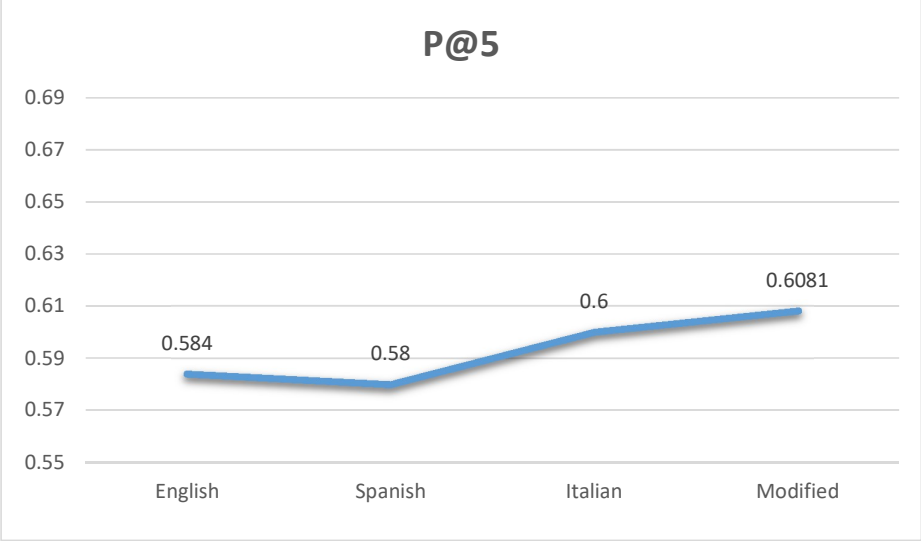


**Figure 5.2.** *Precision at 5*

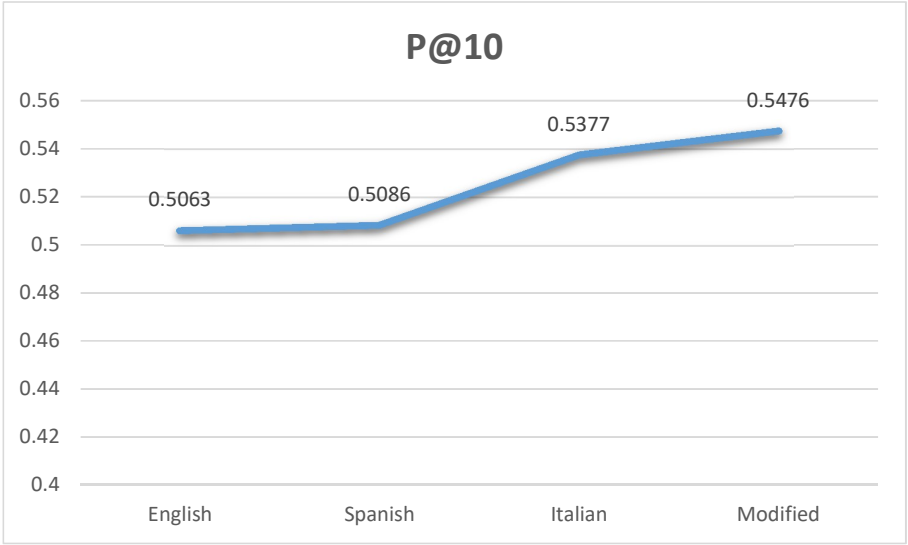


**Figure 5.3.** *Precision at 10*

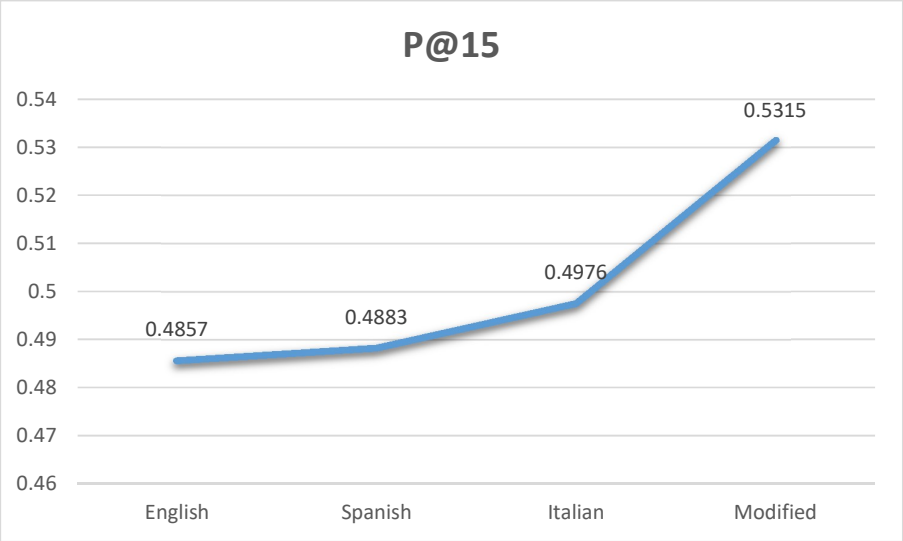


**Figure 5.4.** *Precision at 15*

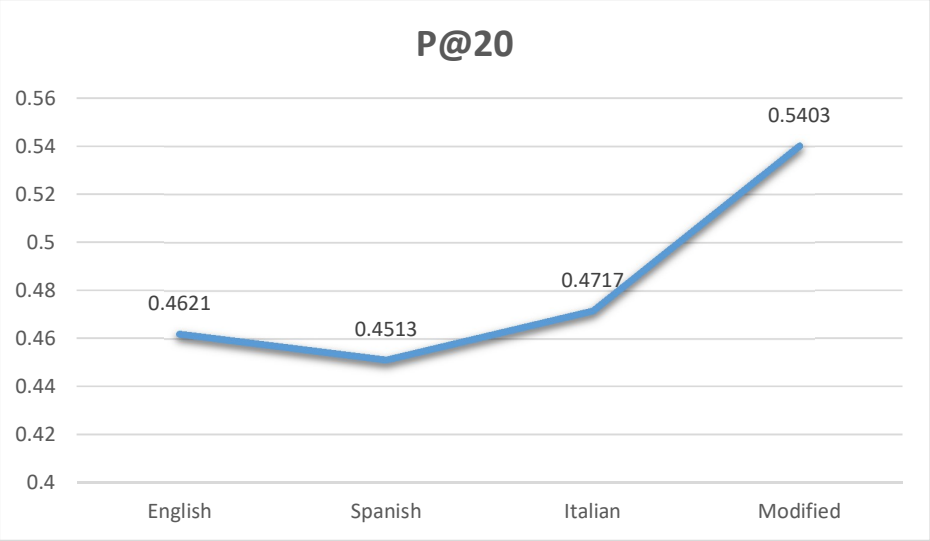


**Figure 5.5.** *Precision at 20*

We also recognized that Italian analyzer performed better for P@5-P@20 compared to other languages where English language became the worst contrariwise modified analyzer became highly ranked among all the others in its performance.

The RR (Reciprocal Rank) is measurement which calculates the first retrieved documents in a query. So if the retrieved relevant document is in 1st position its rank is 1, if it is in 2nd position its rank is 0.5 and if there are no documents retrieved the rank is 0.

The average of RR is taken from the set of queries is called MRR (Mean Reciprocal Rank). This measurement is used when user's intent for first retrieved relevant document and assume that users will keep searching till they find the relevant one. When the document is in some rank $n$ the quality of search is measured reciprocal of its rank, for instance $\frac{1}{n}$. The Figure 5.6 shows the values of MRR for four analyzers. The MRR formula is:

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_i} \tag{5.2}$$

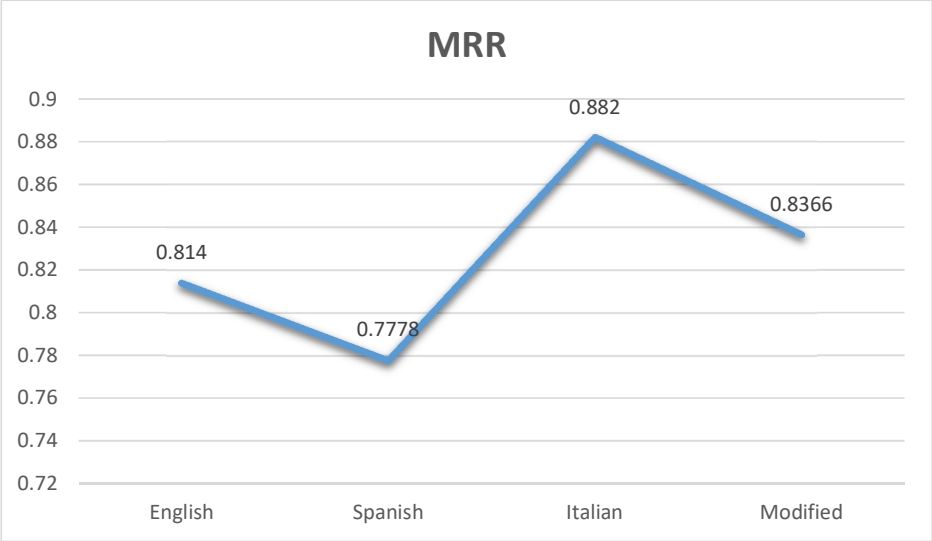


**Figure 5.6**. *Mean Reciprocal Rank*

Italian analyzer performed better in MRR measurement and the worst performance belongs to Spanish language. We additionally uncovered that the most suitable language for retrieving Albanian documents is Italian language.

The Table 5.1. shows the percentage (%) of languages for each evaluation metric based on our modified standard analyzer, values which contain minus (-) performed atrocious compared to our analyzer and these which do not performed better.

**Table 5.1.** *Percentage (%) of evaluation matrices*

| Analyzer/Metric | MAP | P@5 | P@10 | P@15 | P@20 | MRR |
|---|---|---|---|---|---|---|
| English | -3.86 | -2.41 | -4.13 | -4.58 | -7.82 | -2.26 |
| Spanish | -4.16 | -2.81 | -3.9 | -4.32 | -8.9 | -5.88 |
| Italian | -1.8 | -0.81 | -0.99 | -3.39 | -6.86 | 4.54 |
| Modified | 0.7046 | 0.6081 | 0.5476 | 0.5315 | 0.5403 | 0.8366 |

After running these analyzers for each query, we calculated the total number of retrieved documents for four analyzers and the average retrieved documents per query for 50 queries over dataset. In the Table 5.2. retrieved field contains analyzed languages, documents field shows total number of retrieved documents from our corpus and the field document per query (Doc/Query) reveals the average retrieved documents in a respective language.

**Table 5.2.** *Total Retrieved documents*

| Retrieved | Documents | Doc/Query |
|---|---|---|
| English | 28425 | 568.5 |
| Spanish | 21407 | 428.14 |
| Italian | 21232 | 424.64 |
| Modified | 3967 | 79.34 |

We observed that when the query length is extended (more stopwords are used), the number of retrieved documents for three built-in analyzers are increased drastically expect our analyzer (contains Albanian stopwords). We noticed that the highest number of documents were retrieved by English language and the lowest documents were retrieved by our modified standard analyzer.

# 6. CONCLUSION

## 6.1. Conclusion and Discussions

In summary, the thesis indicates distinctness between Lucene built-in analyzers such as English, Spanish, Italian and their performance in retrieving Albanian documents besides, these analyzers are compared with our modified standard analyzer which encompasses all the stopwords of Albanian language. Results are obtained after performing tests with each four analyzers in a fifty different query specifications and lengths through our collection, consisting data from newspapers in Albanian speaking countries.

We implied that information retrieval library along with modified analyzer including specific language improvements provided better efficiency for Albanian evolution set. The results of our analyzer which is presented in this thesis performed better in measurements like MAP (20) and P@K for each value of K for documents in our dataset.

We revealed, if users want to get a great number of relevant documents in their search they can refer to our modified analyzer on the contrary, if they are looking for relevant content to show up as fast as possible they need to refer in Italian analyzer.

For MAP measurement, modified analyzer showed 3.86% enhancement from English respectively 4.16% from Spanish and 1.80% from Italian language. Contrariwise, Italian language worked out in definite superiority for MRR with 4.54% in comparison with our modified analyzer.

Although we remarked that for every analyzed value of K in P@K our modified analyzer performed better immediately after Italian language occurs. The research showed up that the best built-in analyzer in Lucene for retrieving Albanian documents is Italian language.

## 6.2. Future Work

In addition, we do not guarantee that we have constructed the perfect analyzer for Albanian language. Then again, we trust that there is far to build up the ideal one.

In opposite, we uncovered that the Italian is the best language for retrieving Albanian documents. In any case, this commitment is a stage for future improvement of Information Retrieval Systems adjusted for Albanian.

The way we have drawn closer to the evaluation of our analyzer gave as significant clues for its improvement.

However, we will expand our studies to build up a complete analyzer containing all stems of Albanian language afterwards, comparing this analyzer with Lucene's remainder analyzers and other text retrieval libraries in market.

## REFERENCES


[1] *Marz, N., & Warren, J. Big data. (2015). Principles and best practices of scalable realtime data systems. Greenwich, CT, USA.*

[2] Grossman, D. A., & Frieder, O. (2012). *Information retrieval: Algorithms and heuristics* (Vol. 15). Springer Science & Business Media.

[3] Singhal, A. (2001). Modern information retrieval: A brief overview. *IEEE Data Eng. Bull.*, *24*(4), 35-43.

[4] Sanderson, M., & Croft, W. B. (2012). The history of information retrieval research. *Proceedings of the IEEE*, *100*(Special Centennial Issue), 1444-1451.

[5] Büttcher, S., Clarke, C. L., & Cormack, G. V. (2016). *Information retrieval: Implementing and evaluating search engines*. Mit Press.

[6] McCandless, M., Hatcher, E., & Gospodnetic, O. (2010). Lucene in action: covers Apache Lucene 3.0. Manning Publications Co.

[7] Drake, P. (2006). *Data structures and algorithms in Java*. Pearson/Prentice Hall.

[8] *Schütze, H., Manning, C. D., & Raghavan, P. (2008). Introduction to information retrieval (Vol. 39). Cambridge University Press.*

[9] Batko, M., Novak, D., & Zezula, P. (2007). MESSIF: Metric similarity search implementation framework. In *Digital Libraries: Research and Development* (pp. 1-10). Springer, Berlin, Heidelberg.

[10] Karanikolas, N. N. (2009, September). Bootstrapping the Albanian information retrieval. In Informatics, 2009. BCI'09. Fourth Balkan Conference in (pp. 231-235). IEEE.

[11] Çabej, E. (1982). Studime etimologjike në fushë të shqipes (Vol. 3). Akademia e Shkencave e RPS të Shqipërisë, Instituti i Gjuhësisë dhe i Letërsisë.

[12] Zhou, W., Smalheiser, N. R., & Yu, C. (2006). A tutorial on information retrieval: basic terms and concepts. *Journal of biomedical discovery and collaboration*, *1*(1), 2.

[13] Middleton, C., & Baeza-Yates, R. (2007). A comparison of open source search engines.

[14] Karanikolas, N. N. (2015). Supervised learning for building stemmers. *Journal of Information Science*, *41*(3), 315-328.

[15] Croft, W. B., Metzler, D., & Strohman, T. (2010). *Search engines: Information retrieval in practice* (Vol. 283). Reading: Addison-Wesley.

[16] Tague-Sutcliffe, J. M. (1996). Some perspectives on the evaluation of information retrieval systems. *Journal of the American Society for information science*, *47*(1), 1-3.

[17] Arslan, A., & Yilmazel, O. (2008). *A comparison of Relational Databases and information retrieval libraries on Turkish text retrieval*. 2008 International Conference on Natural Language Processing and Knowledge Engineering, 1-8.

[18] Arslan, A., & Yilmazel, O. (2010, October). Quality benchmarking relational databases and Lucene in the TREC4 adhoc task environment. In *Computer Science and Information Technology (IMCSIT), Proceedings of the 2010 International Multiconference on* (pp. 365-372). IEEE.

[19] Igawa, R. A., Kido, G. S., Seixas, J. L., & Barbon, S. (2014, October). Adaptive distribution of vocabulary frequencies: A novel estimation suitable for social media corpus. In *Intelligent Systems (BRACIS), 2014 Brazilian Conference on* (pp. 282-287). IEEE.

[20] Białecki, A., Muir, R., Ingersoll, G., & Imagination, L. (2012, August). *Apache lucene 4*. In SIGIR 2012 workshop on open source information retrieval (p. 17).