LEBANESE AMERICAN UNIVERSITY

An Innovative Automatic Indexing Method for Arabic Text

By

Nour K. Masri

A thesis
Submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Science

School of Arts and Sciences
January 2020
THESIS APPROVAL FORM

Student Name: Nour Masri  I.D. #: 201203406

Thesis Title: An Innovative Automatic Indexing Method for Arabic Text

Program: Master of Science in Computer Science

Department: Computer Science and Mathematics

School: Arts and Sciences

The undersigned certify that they have examined the final electronic copy of this thesis and approved it in Partial Fulfillment of the requirements for the degree of:

Masters of Science in the major of Computer Science

Thesis Advisor's Name: Dr. Ramzi Haraty

Signature:  Date: 29 / 01 / 2020

Committee Member's Name: Dr. Samer Habre

Signature:  Date: 29 / 01 / 2020

Committee Member's Name: Dr. Azzam Mourad

Signature:  Date: 29 / 01 / 2020
THESIS COPYRIGHT RELEASE FORM

LEBANESE AMERICAN UNIVERSITY NON-EXCLUSIVE DISTRIBUTION LICENSE

By signing and submitting this license, you (the author(s) or copyright owner) grants the Lebanese American University (LAU) the non-exclusive right to reproduce, translate (as defined below), and/or distribute your submission (including the abstract) worldwide in print and electronic formats and in any medium, including but not limited to audio or video. You agree that LAU may, without changing the content, translate the submission to any medium or format for the purpose of preservation. You also agree that LAU may keep more than one copy of this submission for purposes of security, backup and preservation. You represent that the submission is your original work, and that you have the right to grant the rights contained in this license. You also represent that your submission does not, to the best of your knowledge, infringe upon anyone’s copyright. If the submission contains material for which you do not hold copyright, you represent that you have obtained the unrestricted permission of the copyright owner to grant LAU the rights required by this license, and that such third-party owned material is clearly identified and acknowledged within the text or content of the submission. IF THE SUBMISSION IS BASED UPON WORK THAT HAS BEEN SPONSORED OR SUPPORTED BY AN AGENCY OR ORGANIZATION OTHER THAN LAU, YOU REPRESENT THAT YOU HAVE FULFILLED ANY RIGHT OF REVIEW OR OTHER OBLIGATIONS REQUIRED BY SUCH CONTRACT OR AGREEMENT. LAU will clearly identify your name(s) as the author(s) or owner(s) of the submission, and will not make any alteration, other than as allowed by this license, to your submission.

Name: Nour Masri

Signature: ____________________________ Date: 29/01/2020
PLAGIARISM POLICY COMPLIANCE STATEMENT

I certify that:

1. I have read and understood LAU’s Plagiarism Policy.
2. I understand that failure to comply with this Policy can lead to academic and disciplinary actions against me.
3. This work is substantially my own, and to the extent that any part of this work is not my own I have indicated that by acknowledging its sources.

Name: Nour Masri

Date: 29/01/2020

Signature:
ACKNOWLEDGMENT

This work would not have been possible without the support of my advisor Dr. Ramzi Haraty whom ensured producing a high level quality of work and was always providing constructive feedback. My close friends Lara, Nadine, Noura, Sabine, Aya and Mohamad who never stopped listening to my constant nagging and kept on encouraging me. My dear team leader Sandra and MDSL colleagues especially Anoud, Mabelle and Freddy who made this experience much more bearable and helped in every possible way. My family who has been there since day one throughout my whole journey for the past 25 years; my dad who never stopped believing in me, my mom where no words can describe her dedication and sacrifices, my little sister Dounia who brings so much joy to our hearts and at last to your beautiful soul grandma, this degree was attained and is specially dedicated to you.

I am grateful for LAU and for this opportunity it has presented to me where I got the chance to work with brilliant people in our field and experiment with new ideas and concepts. Such work would have not been possible elsewhere and without the previous extensive work done by three LAU alumni Walid K. Daher, Manal Houri and Rouba Nasrallah.

A special appreciation also goes to Dr. Samer Haber and Dr. Azzam Mourad for taking the time in revising my work and always providing me with valuable feedback.

This work is indeed accomplished thanks to everyone’s help and support.
An Innovative Automatic Indexing Method for Arabic Text

Nour K. Masri

ABSTRACT

Automatic indexing and texts retrieval methods for languages have been studied for a long time. Compared to other languages, there is still limited research which has been conducted for the automated Arabic Text Categorization. In this work, we present an innovative method to reinforce the accuracy of automatic indexing of Arabic texts by introducing a Thesaurus. Our model extracts new relevant words by referring to the introduced thesaurus which identifies words correlation. The Thesaurus is built through an NLTK toolkit which contains a library that lists the synonyms of a certain word available in WordNet library. The words having the same meaning and that frequently appear together were grouped under one umbrella using a JSON dictionary making it easier to identify the texts topic. Our results exhibit notable improvement in accuracy and efficiency compared to previous works.

Keywords: Automatic Indexing, Arabic Text, Building Thesaurus, Frequent Sets, Synonyms, JSON Dictionary.
# TABLE OF CONTENTS

1. **Introduction**  
   1.1. Indexing Documents ........................................ 2  
   1.2. Importance of the Arabic Language ......................... 4  
   1.3. Methodology and Contribution ............................... 6  
   1.4. Thesis Organization ....................................... 8  

2. **Related Work**  
   2.1. Text Processing ............................................ 10  
   2.2. Automatic Indexing characterized by Frequency and Words Spread 12  
   2.3. Document Tagging and Rule Based Data Mining ............... 14  
   2.4. Thesaurus Based Approach .................................. 15  

3. **Stem Word Extraction and Weight Calculation**  
   3.1. Stemming Definition ....................................... 20  
   3.2. Identification Process ..................................... 21  
   3.3. Extracting Stem Words ..................................... 23  
   3.4. Weight Calculation ....................................... 26  

4. **Proposed Solution**  
   4.1. Phases of our solution .................................... 30  
   4.2. Outcome of the proposed solution .......................... 37  
   4.3. Index selection ........................................... 37  

vii
5. Experimental Results 38
   5.1. Previous Implementations ............................... 39
   5.2. Our Implementation ................................. 42

6. Conclusion 46
   6.1. Further Work ....................................... 47

Bibliography 49

A. List of Prefix and Suffix 55

B. Case Study 57
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Verb and Noun decision Pseudo-code</td>
<td>22</td>
</tr>
<tr>
<td>2. List of Ten Derivations</td>
<td>25</td>
</tr>
<tr>
<td>3. Example of the Synonym Count Effect on Weight</td>
<td>29</td>
</tr>
<tr>
<td>4. JSON Element Example</td>
<td>32</td>
</tr>
<tr>
<td>5. NLTK and Wordnet import</td>
<td>33</td>
</tr>
<tr>
<td>6. Sample Thesaurus Element</td>
<td>34</td>
</tr>
<tr>
<td>7. Thesaurus Building Algorithm</td>
<td>34</td>
</tr>
<tr>
<td>8. New Counting Method</td>
<td>36</td>
</tr>
<tr>
<td>9. Full results of [26] and [27] respectively</td>
<td>40</td>
</tr>
<tr>
<td>10. Summary of the results of [26] and [27] respectively</td>
<td>42</td>
</tr>
<tr>
<td>11. Summary of our results using the Thesaurus Integration Method</td>
<td>43</td>
</tr>
<tr>
<td>12. Comparison of our results using the Thesaurus Integration Method with the previous methods</td>
<td>45</td>
</tr>
<tr>
<td>13. List of prefix pronouns</td>
<td>55</td>
</tr>
<tr>
<td>14. List of suffix pronouns</td>
<td>55</td>
</tr>
<tr>
<td>15. List of the combination of prefix and suffix pronouns</td>
<td>56</td>
</tr>
</tbody>
</table>
A recent study conducted by Statista Research Department revealed that as of year 2019, around 4.4 billion people were actively using the internet. Many countries, including Arab ones such as the UAE and Qatar are on top of the list, being able to cater for almost their whole residents reaching a 99 percent online utilization rate. With this remarkable technological advancement especially in the Arab countries in the last century, Arabic language and literature remains of significant importance having around 420 million Arabic speaking population [14].

From an academic perspective, such a prodigious number of Internet users requires a meticulous process to manage, categorize and identify different types of articles, documents, conferences and social media interaction. In his article, Micheal Bergman has identified that public information on the deep web, which is part of the World Wide Web but with un-indexed contents, is sized and estimated to be five hundred times larger than the typically defined Web. In addition of not being indexed by conventional search engines, it is also currently considered the fastest growing and emerging category of information on the Internet [8]. In the era of machine learning and AI, the essence of smart knowledge relies on keywords and relevant terms; for that, indexing texts facilitates retrieval of the needed and specific information the user is seeking.

In fact, indexing text documents lies on investigating its content to identify
its topic. Moreover, building indices organizes a reference to your data which makes it easier and faster to look up and retrieve relevant information. Previously, the indexing process was conducted manually. Mainly wordsmith, linguists and eloquent people were assigned to perform this task that required being an expert in vocabulary, grammar and analysis. Manual indexing is a tedious and exorbitant process. Moreover, the results have a tendency to be subjective due to human interference where the person choosing the indexes may have personal choices that can affect the objectivity of the text in certain sensitive topics such as politics and history. As the world advances, this manual task started to require a tremendous time and humans started facing a challenge to identify the most relevant information of such a big number of documents that is growing exponentially. Adding to that, indexing is no longer limited to academics articles but expanded to archiving, spam detection, automatic message routing, page content filtering and many other aspects where indexing showed process improvement. This led to looking into Automatic techniques, which are now in great demand to perform indexing.

1.1 Indexing Documents

Researchers identified two different types of indexing each based on a distinct concept: Full-text Indexing and Thesaurus Indexing [36].

1.1.1 Full-Text Indexing

This type focuses on picking terms that are only found in the presented document. It does not allow to refer to words that may be more commonly used or researched for if not available in the text. For example, given a sample text about Adolf Hitler, usually the following terms are associated with Hitler: the Third Reich, Nazi dictator, Führer, etc.. if the latter words are not mentioned in the article, even if highly representative of the topic, cannot be used as an index in this
category of indexing. For that, this approach is simpler to implement and adopt since it only relies on the available terms and words in the text being studied. However, presents limitation and weakens text enrichment advantage.

1.1.2 Thesaurus Indexing

This type on the contrary, allows the use of words that are not specifically available in the presented text, but are highly representative of the topic. In some cases, the document controller opts to include the synonym instead of the word presented in the document as it is widely used by people and leads to a more accurate index as the synonym is more common among users. Given the same example above, people tend to search for an article related to Hitler and his reign by searching for “Nazi Dictator” or “Third Reich”. The difficulty of Thesaurus base indexing is demonstrated in its implementation as it depends on lexicon understanding. From a technical perspective, it requires having a file we will call Thesaurus which contains words and their correspondent synonyms. The file should be regularly updated to ensure new words and concepts are added. This maintenance may require human intervention at first as it does not just group words of the same meaning (synonyms) together but also words belonging to the same context.

This approach is harder to implement since it requires content and sentiment knowledge to succeed. This type will be exploited further in our work.

1.1.3 Building Subject Heading

Regardless of the indexing method adopted, the outcome is the same: Set of keywords produced to identify the text’s subject(s). Subject heading accuracy is crucial as it is the main factor to retrieve relevant information and increase hit rate. As a standard, documenters must include the following fields in the subject heading:

- **Name**: The proper name of the individual, organization or corporation the
text is tackling, i.e. “Adolf Hitler”.

- **Position**: The social or political rank. i.e. “Führer” or “Gefreiter”.

- **Location**: Any Geographic Location: Country, City, Place, etc.. i.e. “Germany” or “Austria”.

- **Activity**: Reason behind the document. i.e. “Secret Meeting”.

The above information will be concatenated to produce the following sample subject heading: *Adolf Hitler > Führer > Germany > Secret Meeting*. The produced should at least contain one keyword to facilitate and accelerate the lookup of the browser or search engines by referring to the heading instead of going over the full text. Now that the basics have been covered, the complication lies in the specification and details of the language in question, in our case, the Arabic language.

## 1.2 Importance of the Arabic Language

### 1.2.1 Origin and History Overview

In its birthplace in the northwestern region of the Arabian Peninsula, Arabic was the main dialect spoken by the people of Quraysh, the tribe to which the Prophet Muhammad (PBUH) belonged to [12]. Even prior to the rise of Islam, Arabic literacy was of extreme importance as it was a medium of oral poetry. In fact, the poet was socially considered from the top ranked statuses one could earn. He was the spokesman and orator for the tribe. He was highly praised throughout his life and was considered a guide when in peace, and a champion and leader when in war [43].

Following the rise of Islam, Arabs and Muslims in particular started giving much more importance to Arabic as it is regarded as God-given language. It in fact carries the miracle of the holy Quran and was also pointed out as a divine purpose. Referring to Verse 2 in Surat 12: “Verily, We have sent it down as an
Arabic Qur’an in order that you may understand”. Arabic is majestic and unique in beauty, and is the most eloquent and expressive of all languages for conveying thoughts and emotions [53].

Even to more recent years, the pietists (a 17th century religious and biblical movement originating in Germany) along with the nationalists consider Arabic the mainstay of the faith and the pillar of nationalism being the differentiating factor among people who otherwise have much in common [12]. The west population, whom are mostly neither Arabs nor Muslims, still refer to books written by Arabs in the field of Medicine, Science and Philosophy in most important universities [50], [53].

1.2.2 Arabic Text Indexing

Arabic belongs to a group of languages known as the Semitic language. The United Nations recognizes it as one of its six official languages [42]. It consists of twenty eight letters and is written from right to left. The letters change their shapes and form depending on the place where they occur in. It is grammatically flexible to an extent where even if the words are arranged in different ways the meaning could still be preserved [41], [53]. This rich language has the ability to construct complex and varied words from basic roots. It is enough to have three letters such as ‘d-r-s’ to be the essence of many terms in the semantic field of 'studying' leading to many derivatives, such as the word ‘dirasa’ which means a study or research and ‘moudarriss’ which means instructor [48]. Since the Arabic Language is not solely bounded to Arabic Literacy but its role also expands to being:

1. A contrivance of artistic and correct expression;

2. A pillar of religion;

3. An anchor of culture; and

4. A centerpiece of contemporary nationalism.
We realize that the 420 million Arabic speaking population’s culture, manuscripts and activities has its weigh and is of great importance. This volume and striking complexity requires indeed a meticulous and diligent technique in order to have proper indexing which will lead to having an organized and easy access to the available data ranging from the first Century until today. Consequently, efforts to building an automatic indexer is introduced to facilitate and strengthen this loaded area.

1.3 Methodology and Contribution

Our proposed approach - Thesaurus Integration - is an enhanced and innovative method for extracting more accurate indices. The main idea behind our method is to take into consideration synonyms available within the text and key terms that are not necessary used in the text, but are highly representative of the document being studied. As a summary, the below contribution is presented in this work:

1. Building an Automatic Thesaurus,
2. Integrating the Thesaurus with the studied texts,
3. Excluding some terms from being stemmed; and
4. Adding new terms to the Stop List

In fact, introducing a thesaurus presented a remarkable improvement. And to enhance further, we have worked on adding more words and elements to the “stop words list” which were not previously considered and were leading to false-positives such as days of the week, month, year and numbers. At last, we also identified words that should be excluded from being stemmed and we added them as an exception while processing the related algorithm such as common and proper nouns. i.e “The United States of America”, “Sheikh Zayed”, etc... We will present in details the effect and value of the above contribution in the upcoming chapters.
1.3.1 Motivation

The Arabic language, being our mother tongue as Arabs living in the Middle East, in addition to being a very complex and rich language, led us to be eager to tackle this topic. Feelings aside, and from a technical perspective, the below were also crucial arguments behind investing in our research:

1. The extensive need to index Arabic documents available in both the global network and the un-indexed Deep Web,

2. The emerging of Artificial Intelligence and Internet of Things nowadays and benefiting from lower error rate,

3. To benefit from the rich Arabic research and documents, and

4. To enhance Knowledge and Language Understanding,

According to IBM, it is estimated that around 80 percent of all information is unstructured, with text category being on top of this list [49]. Such a large percentage requires special treatment, leading researchers to rely on new technologies such as AI and IOT to acquire better, automated and faster results. In addition and from a linguistic perspective, many words may have more than eight synonyms which will not be detected within the text, unless a thesaurus is introduced that contains all the synonyms, degradation of the word and related meanings. The word "Love" has more than ten words that refer to it, each conveying a different stage of love: Al wid, Al Fouton, Al Gharam, Al Ishik, Al Shaghaf, Al Shajan, Al Jawa, etc... If in a certain text each word is treated separately, probably none would be considered as an index due to being distinct words, each having a different root and cannot be grouped together based on morpheme (word origin). But if those synonyms are defined in a file, grouped under a certain term, then the counting method will be able to identify all these synonyms under one bucket and extract more relevant indexes. All leading to a better understanding of the text being studied.
1.4 Thesis Organization

In this work, we highlight the importance of the work previously done in automatic-indexing of Arabic text and present our work which will be detailed in Chapter Four.

The Second Chapter tackles the related work done on Automatic indexing Arabic documents including in specific the work done by the authors of [26] which assigned a weight to the words according to how frequent the word is found along with its spread factor in the text. In addition, this chapter presents the work done in [27] where the author was able to expand the number of relevant words by finding a relation between words belonging to a texts of the same category.

Chapter three will highlight the importance of Stemming and Weight assignment which is the pre-processing phase to be able to proceed with Thesaurus Integration. Chapter Four will introduce our proposed solution and the procedure implemented to choose the relevant words. Chapter five will demonstrate the results, analyze them and compare the improvements with previous work. At last, chapter six gives a conclusion and a proposition for future work.
Chapter 2

Related Work

The aim of our work is to build on and enhance the Automatic-Indexing System for Arabic texts previously built by the authors of [26], [36] and then updated by [27]. The software previously developed extracts representative indices from the text based on weight calculation and then applies an association rule, built on a data mining approach. Our proposed technique was inspired by the latter since improvement was shown when words association technique was introduced. However, presented some constraints where the pre-requisite texts have to be from the same category in order to extract the association ratio. For that, and to enhance the work we decided to integrate a thesaurus that can incorporate and index a text regardless of its category. Hence, eliminating the previous constraint. We will discuss in this chapter the work done in Text Processing in general and then in specific the work done on Automatic Indexing built on identifying words occurrences along with the words spread factor in the text, Text Classification through Data Mining rules and at last, Thesaurus Based Approaches.
2.1 Text Processing

2.1.1 Natural Language Processing

Natural Language Processing (NLP) is a field of research responsible for inspecting how computers could be employed to recognize and exploit natural language text for useful research. Researchers in this area aim to assemble and apprehend how human beings interpret and utilize languages to try and build the corresponding tools and invent techniques to solicit the same on computers and machines to carry out the appropriate tasks [13]. Currently, this is being applied in a varied fields of studies, including but not limited to machine translation, text processing and summarization. In specific, one crucial area of NLP application in our case is multilingual text processing that seeks to make us of the advantages of the WWW and online libraries [11], [13]. Many analyst such as the authors of [39] and [40] have proposed to employ WordNet libraries to enhance the statistical analysis results of natural language texts. The development of those libraries was initiated and carried out at Princeton University. Considered as one of the best NLP references, Wordnet is an online lexical system initially containing nouns, verbs, adjectives and adverbs organized into synonym sets for English language, each representing one underlying lexical concept. The research was then expanded to contain several languages such as Italian, Dutch, Spanish, German and French during the late 20th century. Arabic was introduced to Wordnet during 2006 and later on expanded in 2015 [2], [10].

The author of [21] lists a number of NLP packages that are widely used, such as:

1. **ConQuest**, a part of Excalibur, that incorporates a lexicon that is implemented as a semantic network

2. **InQuery** that parses sentences, stems words and recognizes proper nouns and concepts based on term co-occurrence
3. **LinguistX** parser from XEROX PARC that extracts syntactic information, and is used in InfoSeek

4. **NetOwl** from SRA, a text mining system.

Since Arabic language is considered to be more complex than others, further details related to Arabic text processing will be discussed in the next section.

### 2.1.2 Arabic Text Processing

The authors in [24] presented the issue of recognizing Arabic handwritten characters which still poses a difficulty to the scientific society. In order to be able to arrange the previously segmented handwritten Arabic characters, the authors built two neural networks to achieve that. Their approach correctly recognized seventy three percent of the characters. The experiment was conducted on 10027 training sets and tested on 2132 samples. The main problem presented was that handwritten character classification does not only depend on topographic features extracted but also on contextual understanding.

Stemming has shown a remarkable effect on Arabic text processing. For that, the authors in [25] presented an advanced Arabic stemmer called “Al-Monnakeb”. They were able to reach a remarkable accuracy improvement in both precision and recall, both being above 90 percent. This was due to adding more grammatical rules and introducing a temporal references extractor. The algorithm was able to extract almost all those references from the documents and give special priorities and ranks to know its importance as a temporal reference. Rules that decide whether this word is a temporal reference by itself, a catalyst for creating a temporal reference, or a part of a temporal reference were introduced in this paper.

In order to additionally improve stemming techniques, the authors in [23] tackled the issue of diacritization of Arabic text. Diacritization is the procedure of restoring the diacritical marks (short vowels) of words. Their proposed model
was to categorize Arabic words in order to figure out the function of each word in a sentence so it would be diactritized correctly. Proper diacritization would lead to better sentiment understanding. The implementation was done with the use of Hidden Markov Model and a rule based approach. No previous work was done on diacritizing Arabic text references to compare with the result of this work at the time of publication. However, accurate outcome was presented.

In a more recent paper, the authors in [6] proposed a reduced automatic diacritization process of Arabic texts. The system re-establishes the diacritical markings only if it minimizes ambiguity. Hence, where it is mostly needed by combining morphological analyzers and context similarities. It generates all candidates for the diacritics, and then is able to eliminate word ambiguity through statistical approaches. The results were found useful in 57 texts out of 80.

The authors of [28] present a tool called Arabic Duplicate Detection. It is responsible for the adaptation of the k-way sorting algorithm and is specifically tailored for Arabic input. The benefit of the presented work lies in presenting clean data that will lead to more accurate results.

2.2 Automatic Indexing characterized by Frequency and Words Spread

As previously mentioned, our technique is based on the work done by [27], which is a continuation of the work initiated by [36]. In this section, we aim to briefly present the model implemented by [36], any additional detail is found in his work.

The proposed incorporates four layers, each of which is developed in a way to operate as a standalone layer, if needed. Layers are interchanging information and serving as an input for the next layer. The implementation was done in this way in order to provide the system with the possibility to test any other algorithm that serves the same topic (i.e., any stemming technique can be injected and the flow will not be interrupted).
Four Layers Model:

1. **Read Whole Document and exclude Stop Words/Stop Phrases**
   This layer will read the document and exclude all unwanted words or phrases that do not present any added value to the context of the file. Remaining valid words, will be considered as an input for the next layer, which is responsible for stemming.

2. **Apply Algorithm to Extract Arabic Stem Words**
   This layer will take the output provided by the previous layer and apply the stemming algorithm. It will return the word to its stemmed form (root) and the word count in the text to feed as an input for the following layer. Almost all results analysis confirm that stemming along with spelling normalization remarkably improve both indexing and retrieval for the mere fact that the many variation of the words will be all treated as one word and will give better hits [56].

3. **Perform Weight Calculation**
   At this stage, the algorithm will calculate the ideal distance, average ideal distance, and average distance between the words. Once done, the weight of each word is produced and saved. Weight assignment is the core of producing accurate results since the difficulty of automatic indexing lies in determining words relevance.

4. **Select Appropriate Keyterms**
   At last, when all weights are presented, the algorithm will identify the highest number to select the best ranked terms as representative indices to the text. In the next chapter, the third and fourth layer will be described further.
2.3 Document Tagging and Rule Based Data Mining

Text classification, also known as text tagging, aims to assign and determine a category to un-categorized texts. In the past, this process was done manually, it was a difficult and expensive process since it needed time and resources to manually sort the data and handcraft rules that are difficult to maintain among indexers. Text classification remains an important part of businesses in our day as it provides insights on data. In consequence, automating this process became an interest for many researchers. Some of the methods adopted are listed below:

1. Naïve Bayes (based on Bayes’s Theorem) proposed by [37],

2. Decision Trees by [46],

3. Neural Networks by [30], and

4. Support Vector Machine (SVM) by [17].

In specific for Arabic language, the author of [4] suggested an approach relying on linguistic characteristics by identifying the feature frequency at first. Then proceeded with calculating the importance of each one for every class based on Chi Square (factor determining if a notable difference exists among the expected and observed frequencies in one or more categories). Seven datasets of different classes were chosen such as writers, poems, websites and forums. The corpus contained 17,658 texts and around twelve million words. Using both SVM and C5.0 (classification algorithm) a tool named (ATC) was developed to extract features to be able to sort the texts and determine to which category or class each belongs to. The author of [33] implemented a tagger (tool that produces tags) that uses rule-based techniques along with statistical methods to automate the process. Similar to English, Arabic word types are categorized into several divisions: verbs, nouns and particles. For better accuracy and grouping facilitation affixes were also removed.
In work done by [44], a set of texts from three different categories were pre-processed and led to building a representative and distinct grouping of terms for every document. Then, they applied an algorithm responsible for identifying set frequency which is the Apriori algorithm. The result is sets of words that frequently occur together. Now that the base file containing tags and representative words was built, a new document was processed in order to classify and extract from it the recurrent terms sets. The possibility of having a text belonging to a certain subject is determined by multiplying the probabilities of the frequent sets in each category. The same approach with some modification was also implemented by the authors of [27].

2.4 Thesaurus Based Approach

Within the framework of information retrieval (IR), a thesaurus is a mean of words arrangement or what is also called controlled vocabulary. It serves to minimize linguistic ambiguity by enforcing uniformity and consistency in the way objects are being stored [52]. From a linguistic perspective, the thesaurus is a dictionary of synonyms that helps with the assignment of desired words to fetch semantic metadata related to its content within the object. Constructing a thesaurus is a desired and effective method in Information Retrieval Systems, it boosts precision and control of idioms [35]. It should ideally consist of a list of essential words related to some subject or keyterm. The use of thesaurus has improved IRS by 10 to 20 percent [31]. Using concepts and synonyms rather than just the available words in the text for automated indexing adds specificity to the document representation [29]. Some work has been done to automatically construct a thesaurus for English Language while very little effort has been put for Arabic Language. In the upcoming sections we will present the different Thesaurus building methods done for both languages.
2.4.1 Building Thesaurus for English Language

In their work, the authors in [54] introduced keyphrase indexing by combining two approaches for thesaurus based automatic indexing: Keyphrase extraction and term assignment. Keyphrase extraction method analyses the phrases appearing in the text in order to identify its significance based on certain characteristics suchlike recurrence and length. While in term assignment, keyphrases are picked from a controlled vocabulary of terms. Documents and texts are classified into classes that correspond to their vocabulary elements according to their content [17]. The proposed algorithm, called KEA++ which is an enhancement of KEA, operates in 2 phases: (1) Identify the candidate and (2) filter the candidate. The authors were able to identify thesaurus terms that are relevant to the text in hand and then relied on machine learning techniques to chose the most significant terms [55].

The authors in [18] also built text classifiers trained on exerts categorized as either positive or negative based on machine learning technique. However, the authors faced issues when adopting the same strategy in different domains.

The authors in [38] proposed a new and enhanced method that overcomes the drawbacks presented in [18] by merging previous lexical knowledge with supervised learning. Two models are constructed, one for lexicon of pre-defined words referred to as sentiment-laden words, while the training model was built on tagged documents or also called as labeled texts. The combined model produced a multinomial Naïve Bayes classifier. The amount of training data required which was considered as drawback was remarkably reduced by inspecting and benefiting from previous lexical knowledge. The results showed an improvement for all datasets studied.

The authors in [29] compared the retrieval effectiveness of five different methods for automated (machine-assigned) indexing using three test collections. In their comparison experiment the authors utilized SWORD (Statistical Word-Oriented Retrieval from Database) a software created by BICC for information
retrieval search [29], [47]. In SWORD all words in documents which do not appear on a chosen Stop-List of 250 common words are stemmed and are given a weight according to their abundance in certain documents and infrequency in others. The best result was presented when the combination of concept-based indexing and word-based indexing methods was employed.

2.4.2 Building Thesaurus for Arabic Language

The Arabic language, being one of the only remaining complex languages that is still commonly used, has many properties. One of which is the grammatical malleability, where terms may be arranged in varied and distinct ways making it harder to determine polarity of the text. According to [51], developing automatic Text Categorization (TC) for Arabic is a demanding, complex and requires a considerable amount of time to perform. Also, TC technique for Arabic documents is not as efficient as it is for English due to its linguistic structure. Such reasons justify the lack of research in this domain.

The authors in [31] constructed a Thesaurus to ameliorate Arabic IRS. The study included 242 texts retrieved from the National Computer Conference held in Saudi Arabia. The authors study revealed that referring to a thesaurus will amplify the accuracy of the Arabic retrieval system when referring to the roots or stem of the words [53]. The authors in [51] proposed a new classifier for Arabic text categorization called FRAM. They divided the work into two stages: in the first phase, they pre-processed the texts that were already categorized following that, they extracted the relevant keywords. In the second stage, they built a database from the feature terms. During testing, uncategorized documents were classified using FRAM and compared accuracy with other three Bayesian learning classifiers. FRAM outperformed the mentioned techniques as it was estimating the appropriate category by calculating the Frequency Ratio for each feature of the new document based on the candidate features of the training set instead of having to carry out several feature selection (FS) and eliminating the
lowest frequency of the presented. The authors in [45] constructed an Arabic dataset comprising of five hundred movie reviews using SVM and NB. Manual spelling correction was performed in the pre-processing phase in addition to removing stop-words, words stemming and N-Grams tokenization. This technique presented almost 90 per cent accurate results. However, the size of the dataset is considered small.

The authors in [1] implemented a semantic indexing and query method for IRS. The authors relied on Arabic Wordnet as its semantic reference to define and inspect the effect of single words indexing in comparison to concept indexing. Wordnet has a library containing vocabulary. Synsets is a data structure employed by the latter and is presented as a pair of synonyms and pointers. The pointer identifies the relation found between the words and other synsets. Words can belong to a variety of categories. Results show that semantic indexing precision at different document used was higher in all measures. On average, they obtained a 60 percent precision.

The authors in [32] summarized in their paper several work done in Arabic Sentiment Analysis (SA). Nowadays, researchers have gained a lot of interest in SA and is currently a prominent topic of study and research for Natural Language Processing (NLP). It is defined as the study and analysis of people’s comments, assessments and point of view regarding a certain topic. The number of Arabic SA research witnessed an increase in the past years; it presented 39 papers in 2016 with a total of 133 papers since 2003 [32]. In [20], the methodology adopted combined three different classifiers: Lexicon-based opinion classifier, maximum entropy method and at last KNN. The lexicon-based opinion classifier’s main objective is to be able to differentiate the distinct categories to be able to classify as much documents as possible. MEM also known as maximum entropy method is applied on the training set where the same is reapplied for KNN. This combined methodology improved accuracy from 50 percent to almost 80 percent. The authors pre-processing phase included removing stop words, stemming, assigning
weights to words and tokenization (taking a text or set of text and breaking it up into individual words).

The authors in [7] built a corpus from tweets having positive, negative and neutral polarity. They combined Ruby on Rails and an API provided by twitter to collect and classify the data. They based their training set on 4700 tweets which is a relatively a small dataset but were able to reach 80 percent accuracy of content analysis and published the corpus online for other researchers to benefit from it. Their work was mainly focused on Saudi dialect corpus that applied SA to twitter content to identify tweets polarity.

The authors in [5] classified Arabic sentiments using Mubasher product, an analysis tool, through varied techniques such as NB and SVM. The latter presented the best accuracy: 89 percent without N-gram feature. However, the dataset was also of small size. The pre-processing phase included: normalization, tokenization, removing stop words, stemming and filtering token by length 3. In the same context, and in an innovative method, authors of [19] were the first to treat Arabizi Sentimental Analysis. Arabizi is the term attributed to Arabic informal chat alphabet. It is the combination of Latin script and Arabic numerals which became widely used among Arabs on social media in the 21st century. The authors first tokenized tweets into words, mapped every emoticon into its corresponding word and at last converted the Arabizi words into Arabic word through a rule-based converter. NB and SVM were chosen to classify the tweets. SVM presented higher accuracy. It was also noted that upon removing neutral tweets at an early stage, an improved precision was shown for both classifiers. An 86 percent accuracy was reached in the 3206 tweets dataset used in this study.
Chapter 3

Stem Word Extraction and Weight Calculation

Since our work is focused on enhancing the work done by the authors of [26], [36] and [27] this chapter presents how the second and third layer in the mentioned work is implemented. This section is important as it will clarify how the adjustment presented in our work serves the initially proposed method. Several studies have shown the efficacy of stemming the words prior to indexing them, specially for the Arabic language. In addition, weight calculation is the methodology that will help us identify the importance of each word.

After reading the document and excluding the unnecessary words by comparing them with our new enhanced stop list, the algorithm will examine the remaining words. It will differentiate between verbs and nouns in order to apply the convenient stemming technique as each type is stemmed based on different rules. After stemming, a weight needs to be given to each word based on several criteria that will be discussed in the next chapter.

3.1 Stemming Definition

To elaborate, stemming a word is the mechanism adopted to restore the word to its initial form, also known as root. It is an extremely important procedure as it
will provide better accuracy to our counting method. Since the Arabic language is very rich, a single word may have many grammatical variations such as:

1. Singular or Plural
2. Masculine or Feminine
3. Definite or Indefinite Noun
4. Adjective or Subject
5. Attached or Detached Prefix/Suffix

Since those variations do not alter the general meaning of the document and lead to the same understanding whether put in its singular form or plural form for example, they should be treated as one entity instead of being segregated. In the next section, we will briefly present on what basis stemming is performed.

3.2 Identification Process

3.2.1 Rhyming

When reading a word, we need to first identify its type whether it is a verb or a noun. Since each type has a different rhyming, this will help us identify the stemming technique to follow. Each word will be matched against a set of predefined rhythms in the Arabic grammar. For verbs for example, we will be able to identify whether the word is in its singular or plural form along with its tense and whether attached pronouns are found. Once identified, the stemming technique will be applied based on the category this word falls under and will be returned to its correspondent root.

3.2.2 Identifying Nouns and Verbs

Similar to any other language, verbs are the words defining the action being performed in a sentence. It outlines what the subject is doing. Both nouns and
verbs form the main part of a sentence.

As mentioned, rhyming helps identify the type of the word, either a verb or a noun. For example, words that rhyme with "Yaf3al" (does), "If3al" (do) are definitely verbs while words that rhyme with "Fa3el" (doer) or "Maf3oul" (done) are nouns. In addition, another method to identify the word type is by inspecting the terms that precede it. For example, the pronouns "Lam" or "La", that also belong to the stop-list terms, are pronouns that always come before a verb. Hence, any word following those pronouns will be treated as a verb and stemmed accordingly. The same logic with different set of rules apply to nouns.

At last, if the above mentioned could not determine the word type, we inspect the attached pronouns which are either found at the beginning or the end of the word. Where some of them only attach to verbs while others only to nouns.

The below algorithm extracted from [26] and presented in figure 1, summarizes the possibilities considered in the model and on what grounds characterization is being based on:

```
WordType DecideVerbOrNoun (PrecededWord) {
    If PrecededWord belongs to 'أدوات النصب' or 'أدوات الجزم' Return Verb;
    Else If PrecededWord is 'إسم موصول' Then Return Verb;
    Else If PrecededWord rhymes with 'فعل' Then Return Verb;
    Else If PrecededWord is Verb Then Return Noun; /* 2 verbs can not precede each other */
    Else If PrecededWord is 'حرف جر' Then Return Noun;
    Else If attached to it the following prefixes: 'بال', 'ال', 'فالف', 'فال', 'كال', 'مفعل', 'فاعل' then
        Return Noun;
    Else If Word Rhymes with 'مفعل', 'فاعل' Return Noun;
    Else Return 'Unknown';
}
```

Figure 1: Verb and Noun decision Pseudo-code
3.2.3 Removing stop-lists terms

Words that do not add any significance to the text’s content will be excluded from our study as our counting technique will be highly affected by those redundant terms. For example, the letter "wa" that joins between two different words is highly used in Arabic and does not have any added value from a significance perspective. The same applies to pronouns such as "Lam", "Lan" and "La". However, the latter help us identify the words that follow as previously explained, nevertheless should be excluded. For that, a predefined list is prepared with all those exceptions where we are checking whether each word belongs to that list and is removed if it is the case. The same logic is applied to stop-list phrases where the first word will be matched with the phrase in the predefined list while looping on the remaining words and shall be removed accordingly if we find a positive match. As an example "To whom it may concern" will be excluded in this case. We have expanded the previous used stop list to contain additional 210 words that if excluded, presented more accurate results such as numbers, months and special characters.

3.3 Extracting Stem Words

After identifying the type each word belongs to, we need to apply the correspondent stemming algorithm which is identified in our "Stemming Algorithm". The output will be the word in its root form. We will tackle verb stemming at first and then noun stemming as suggested in [22].

3.3.1 Verb Stemming

3.3.1.1 Inspecting Attached Prefix and Suffix Pronouns

Verbs in Arabic may have 2 forms of pronouns which help in adding more details to the word: attached and discrete. The latter pronouns can be easily identified and are placed in the stop-list terms to be discarded as previously mentioned.
On the other hand, the attached pronouns whether at the beginning, at the end or possibly both ends of the word, will need to be identified to be removed so that proper assessment to which stemming technique to adopt is presented. A finite list containing the prefixes and suffixes is defined. Words will be checked against this list and the aforementioned pronouns will be removed based on an algorithm responsible for pattern matching. The list can be found in Appendix A.

3.3.1.2 Checking Verbs against the “Five Verbs”

According to the authors in [16], [26] and [34] the “Five Verbs” are five standard and known verbs having special properties in the Arabic language. They can be only put in the present tense and mostly end with the letter “N”. Non-essential letters attached to the five verbs are not classified as pronouns. This restriction and rule resulted in a gap in the first phase where attached pronouns are removed since those letters were undetected. For example “Yaktouboun” (Verb for They Write), has the first and last 2 letters as non-essential, those letter were not removed with the previous steps performed. For that, rhyming is used to identify whether a verb is a member of this mentioned set or not. Then, proper stemming is applied in this case.

3.3.1.3 Checking Verbs against the “Ten Verb Addition”

Similarly to previous set of verbs, the “Ten verbs” have also special properties with a different derivation formats that are built from three letter root. The derivations of those verb exist in ten different forms. Three of them are obtained by adding one letter to the original stem verb, five are obtained by adding two letters, and the other two derivations are obtained by adding three letters [27]. Similarly, rhyming will be also adopted to detect the verbs derivation and the non-essential attached pronouns. Once the verb is identified, the algorithm will remove the letters and proceed with the stemming accordingly. The ‘ten derivations’ list
extracted from [26] along with an example is presented below in figure 2:

<table>
<thead>
<tr>
<th>أصل الفعل</th>
<th>مثال</th>
<th>الزيادات</th>
<th>أصل الفعل</th>
<th>مثال</th>
<th>الزيادات</th>
</tr>
</thead>
<tbody>
<tr>
<td>هزم</td>
<td>إنهم الأداء</td>
<td>ضرم</td>
<td>أضرم النيران</td>
<td>فعل</td>
<td></td>
</tr>
<tr>
<td>قرف</td>
<td>قرف خطأ قد أداها</td>
<td>سرع</td>
<td>سرع البحث</td>
<td>فعل</td>
<td></td>
</tr>
<tr>
<td>زهر</td>
<td>زهر الورد</td>
<td>كتل</td>
<td>فاعل الاداء</td>
<td>فعل</td>
<td></td>
</tr>
<tr>
<td>غرق</td>
<td>غرق عيناها</td>
<td>سب</td>
<td>تسب في وفاته</td>
<td>فعل</td>
<td></td>
</tr>
<tr>
<td>خرج</td>
<td>استخرج النفط</td>
<td>عطف</td>
<td>تعاطف مع صديقه</td>
<td>فاعل</td>
<td></td>
</tr>
</tbody>
</table>

Figure 2: List of Ten Derivations

3.3.2 Noun Stemming

Several factors renders the process of noun stemming more complex than verb stemming, even if several rules are similar. The different forms in which a noun comes in plays a major role for this complexity such as:

1. **Number**: Singular, double or plural form, whereas each format may have several exceptions

2. **Gender**: Male or female,

3. **Derivations**: where noun may have no specific format

The following steps are adopted to extract the root from the noun:

1. If the noun is in its plural form, it will be restored to its singular form

2. Detect in case of any attached pronouns (Prefix or Suffix)

3. Validate and compare the noun against the five nouns

4. Validate and compare the noun against the common derivations: M-derivations, T- derivations, and miscellaneous derivations
Additional guidelines can be found in [26]. After all the words in the document are properly stemmed, a certain weight should be given to each term. In [26] the weight relied on three factors, while in our enhanced method, a new factor is introduced and the formula now consists of four factors. We will describe below how the weight is calculated.

3.4 Weight Calculation

Weight calculation and assignment is a crucial phase as it is the main mechanism for determining which words will be chosen as indexes in the text.

3.4.1 Factors affecting Weight

As mentioned in [26], the word’s weight calculation is determined and affected by three factors:

1. The word count,

2. The stem count, and

3. The spread of that word throughout the document. This factor was newly introduced and implemented by the authors in [26]. They based their assumption on the fact that sometimes a word may be frequently appearing in a certain paragraph of a text only but it does not necessarily conclude the subject of the whole text. However, if a term is found throughout the whole text (i.e., spread) then it is more probable to be representative and should be considered as an index word. The more spread the word is, the larger the factor. Which eventually leads to a higher weight.

In our enhanced work, we added the “Synonym Count” abbreviated as “sym” in our formula as a factor making the total factors affecting the equation four components. In chapter four we will explain in details the proposed approach and
how this added factor has a major affect on the accuracy of weight calculation and indexes retrieval.

### 3.4.2 Formulas and terminologies

Performing count calculation of: the word, its stemmed varieties and the associated synonyms with it, is straight forward. However, the spread factor calculation is more challenging and requires more effort due to the need to keep the history and location of the word to identify the difference in position. This can be achieved with the use of the following formula:

\[
    w = m \times sm \times sym \times f
\]

Where:

1. \( w \) is the weight
2. \( m \) is the count of a specific term
3. \( sm \) is the count of the stem words of a certain term
4. \( sym \) is the count of the synonyms of a certain term
5. \( f \) is the spread of the word in the text.

Further details on the calculation procedure, formula verification and concrete examples can be found in [26]. In the next chapter, we will discuss our proposed approach and the effect of introducing the new factor “Synonym Count” in the weight calculation formula.
Chapter 4

Proposed Solution

This chapter details our enhancements and proposed solutions which were built on top of the solution found in [26]. As previously mentioned, auto-indexing of Arabic documents relies on automatically retrieving relevant words that if chosen, provide a proper representation of the text’s subject. Those terms are usually referred to as indexes. The objective of this thesis is to ameliorate the previous approaches adopted, and to improve the extraction result percentage of the relevant index words.

Having said that, and following the successful improvement presented in [27] where the authors generated item-sets of recurrent and frequent patterns using Apriori, we decided to benefit from this method to build our own enhanced approach. Apriori is an algorithm initially proposed by the authors in [3] which is designed for extracting item set that frequently appear together. It is achieved by mining and association rule learning over relational databases. In order to take further advantage of the link between words in a text, we decided to integrate a thesaurus containing wider options. This approach is adopted following the fact that Arabic language is a rich one where several words and terminologies can be used to describe the same term. The thesaurus that we built would contain words along with their synonyms where each is grouped under its own category. We associate a “Key Term” to each mentioned category that represents the set of words contained under it. This will allow us to have a comprehensive view of
all synonyms that are associated to the same topic in the studied texts. Many words that were not previously extracted are now successfully considered when adopting this approach as the weight calculation was adjusted to consider the synonyms. With the introduction of the mechanism built on benefiting from the relations of words, more indexes will be extracted out of the text, leading to a more accurate auto-indexer system.

The following example illustrates the solution proposed. Suppose we are indexing a text that belongs to the subject of “Nift” or “Petrol/Fuel” in English. Ideally, the word “Petrol” would be picked as an index based on the previous counting methods due to its frequent appearance in the text. However, the words “Dollar” or “OPEC” might not surpass the threshold to be chosen in the indices list as they would not be frequently found since their word count would be minimal. Nevertheless, those terms are extremely representative of the text and are in fact indices to be considered. In order for the algorithm to choose those terms, the weight of those words must be higher. Since the initial three factors (word count, stem count and word spread) were already properly handled, adding a fourth factor (synonym count) to the formula would increase the weight of a word that is not frequently found in the text but is a synonym to another word that is frequently found. Below example presented in figure 3 shows the effect and impact of our proposed changes:

![Figure 3: Example of the Synonym Count Effect on Weight](image-url)

<table>
<thead>
<tr>
<th>After adding the new criterion, formula was re-calculated as per the following:</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Weight = Word Count * Stemmed Count * Word Spread * Synonym Count</td>
</tr>
<tr>
<td>- Old Weight $\gamma_{wp} = 83.25$ (Not retrieved as an Index)</td>
</tr>
<tr>
<td>- New Weight $\gamma_{wp} = 582.75$ (Successfully retrieved as an Index)</td>
</tr>
</tbody>
</table>

In the next section, we will detail the steps adopted which led to the enhanced solution.
4.1 Phases of our solution

Our solution consists of additional three phases where each will be discussed in depth in the following sections:

1. Stop-List Adjustment
2. Thesaurus building, integration and management
3. New weight formula calculation

4.1.1 Stop-List adjustment

After examining the results shown in the previous work done by the authors of [26] and [27], the RI (Retrieved Irrelevant) index was considered relatively high. In order to improve it, we analyzed the reason behind having this amount of retrieved irrelevant words and found out that many of the indexes retrieved were actually stop words that were not taken into consideration in the list used. Following this ascertainment, we managed to add around 210 words that if excluded, presented more accurate results. Some of those are numbers, months and special characters. This adjustment led to an improvement in the Precision percentage which will be shown in the following chapter.

4.1.2 Thesaurus building, integration and management

To build our thesaurus, we had to go through several steps and methodologies since building one from scratch is a tedious job. We could not find a thesaurus online that would be easily integrated with the program already built by the authors of [26] which we worked on enhancing. Hence, we needed to design and implement a thesaurus of our own. We will present the technologies used in the back-end and then the logic adopted to have an optimized and coherent thesaurus.
4.1.2.1 Technologies and frameworks used

JSON

Since the core of our work is based on data understanding and manipulation, we need to employ a strong and powerful tool to achieve the desired results. In order to properly represent the data in hand, we first decided to use the JavaScript Object Notation (JSON) to represent the data in a tree like structure. JSON is an open-standard file format that is language independent. However, for simplicity and fast adaptation adopts conventions that are similar to the most used languages such as the ones used in C-family, Java and several others. These characteristics help JSON to become a requested technology for data-interchange language [15]. JSON dictionary structure is focused on:

1. A compound of name/value pairs. Also known as object, record, or associative array.

2. An ordered list of elements. In other languages, this is conventionally known as an array, vector or list.

Below in figure 4 is an example of how the items (words and synonyms) were saved in the thesaurus. We can see that the “key term” for this selection is found at the top of the list and all related synonyms are displayed under it.
Python

Python is known to be a powerful high-level programming language. It is designed with features to facilitate data analysis, visualization, artificial intelligence, and scientific computing. For that, we decided to use Python as our languages due to the several benefits that serve our purpose and as it is most suitable for machine learning and AI-based projects including but not limited to simplicity and access to abundant libraries and frameworks.

NLTK Library

NLTK is a promising platform for creating and implementing programs to help manipulate and exploit human language data, mainly targeted for Python language. It provides a user friendly interface for over fifty corpora and lexical resources. For Arabic language, it presents WordNet Library. It also includes libraries for processing, classification, tokenization, stemming, parsing, and semantic reasoning [9]. In our work, we heavily depend on WordNet’s corpora and resources to achieve our goal. Figure 5 presents the imported library used in our work. Detailed steps will follow in the following section.
4.1.2.2 Steps and methodologies

Wordnet contains a function “Synsets()” that lists synonyms of a certain word. In order to build our thesaurus, we went over the set of words found in the studied texts and retrieved the synonym of each word. This process can be performed on any given input. Moving forward, we saved each entry in the JSON file and made sure to eliminate duplicate data. In addition, we made sure to concatenate all the synonyms of a certain word if found under different key terms, together.

Let’s take the example of two words that are synonyms, if we list the synonyms of Word1 first we will find that Word2 is in the list. If we apply the same for Word2 we will also find that Word1 in the list since the function used lists all possible synonyms of the word. Hence, we made sure not to create two different entries in that case and group them all together.

Following the same strategy, we were able to build a robust thesaurus that also included words that are logically related and not just literal synonyms. Below found in figure 6 is a sample retrieved from our thesaurus for “Mawared/Resources” as a key term. As shown, several terms that are related to the key term “Mawared” were displayed under it such us “Gaz, Kaz, Nift and Petrol”. All those words are logically related to each other and contribute to the same meaning even if they are not actually a definition of the key term.
Moreover, below in figure 7 is the algorithm and logic employed to ensure optimization:

```python
synonyms_list = omw.synsets(word)
if len(synonyms_list) == 0:
    continue
arabic_word_synonyms_list = synonyms_list[0]
synonyms_list = arabic_word_synonyms_list.lemma_names(lang='arb')
if len(synonyms_list) == 0:
    continue
synonyms_list = stem_list(synonyms_list)
key = ""

found = False
for synonym in synonyms_list:
    if found:
        break
    for ar in arabic_text:
        if unicondata.normalize('NFKD', ar).casefold() == unicondata.normalize('NFKD', synonym).casefold():
            key = ar
            arabic_dictionary[key] = synonyms_list
            found = True
            break
if not found:
    arabic_dictionary[synonyms_list[0]] = synonyms_list

for key in arabic_dictionary:
    syns = arabic_dictionary[key]
    for synonym in syns:
        for word in arabic_text:
            if unicondata.normalize('NFKD', synonym).casefold() == unicondata.normalize('NFKD', word).casefold():
                if synonym in arabic_dictionary_counter:
                    arabic_dictionary_counter[synonym] += 1
                else:
                    arabic_dictionary_counter[synonym] = 1
                if key != synonym:
                    if key in arabic_dictionary_counter:
                        arabic_dictionary_counter[key] += 1
                    else:
                        arabic_dictionary_counter[key] = 1
```

Figure 7: Thesaurus Building Algorithm
After building the thesaurus, integrating it, and managing it, we introduced a new function that searches within the file upon request for the items and synonyms found and accordingly updated the counting method previously employed to cater for the newly introduced method. A thorough explanation will be presented in the following section.

4.1.3 New weight formula calculation

After successfully building a thesaurus containing several synonyms and terms that are related together under a unified key term, the next step is to modify the counting method and formula previously adopted to now take into consideration the introduced criterion. We will describe below the adjustment performed on the counting process to finally conclude the enhanced formula to be used in the weight calculation of each word.

Adjusted formula

In the work done by the authors of [26], the following formula was adopted to calculate the weight of each word:

\[
    w = m \times sm \times f
\]

Where “m” is the count of a certain word, “sm” is the count of stem words of a certain word, and “f” is the spread of the word within the document.

In our work, we added to this formula the “sm” factor, which is the synonyms count of the word being weighed. In this way, a better representation to the synonyms is possible since the words that belong to the same meaning are being treated as a pool and not individually. The formula would now become:

\[
    w = m \times sm \times sym \times f
\]

Adjusted counting method

In order to obtain the synonym count referred to as “sym” factor in the formula,
the counting method in the text was adjusted so that the words belonging to the same key term are grouped and counted together. The function will refer to the thesaurus file to know the current word belongs to which group of words and be counted and displayed correctly.

Below in figure 8 is an example extracted from one of the texts we tested. As shown, each word under “Neft” key term has its own count. Those related words were then summed to produce the total of the words belonging to this key term. This sum is the “sym” factor that will be introduced to each calculated word.

Figure 8: New Counting Method

As an example the word “Dollar” was not initially retrieved as an index using the old method as its weight did not exceed the threshold set:

\[
OldWeight(Dollar) = 83.25
\]

After adding the “sym” factor and re-calculating the weight it became:

\[
NewWeight(Dollar) = 582.75
\]

Where the latter was successfully retrieved as a correct index after surpassing the threshold which led to an increased number of RR (Retrieved Relevant) indices and eventually to a better Recall percentage.

In the same list, another example is found where the word “Wokoud” was not initially retrieved as an index using the old method as it only appeared once in
the text and its weight was negligible.

\[ \text{OldWeight}(Wokoud) = 1 \]

After adding the “sym” factor and re-calculating the weight it became:

\[ \text{NewWeight}(Wokoud) = 450 \]

Where the latter was successfully retrieved as a correct index after surpassing the
threshold which led to an increased number of RR (Retrieved Relevant) indices
and eventually to a better Recall percentage. The same strategy was applied
throughout the whole texts and a remarkable improvement was shown, results
will be displayed in the following chapter.

4.2 Outcome of the proposed solution

Our solution expands the output of the previously extracted as we were able to
present new words that are considered relevant and should be chosen in indexing
the text. It promotes words that were not associated with the convenient weight
as they were not frequently found in the text but were synonyms to other words
that are found abundantly.

4.3 Index selection

The selection proposed in this work is based on retrieving as much relative terms
as possible but only choosing five indexes. This was achieved after all words were
given the appropriate weights, and then the highest five were picked. The calcu-
lation of the weight was described in Chapter three. The proposed mechanism
which presented a new way to find relation between those words gave a more ac-
curate weight to the words. This led to a better index selection where the related
results outcome will be exhibited and analyzed in the following chapters.
Chapter 5

Experimental Results

To elaborate further and show the promising improvements our method presented, we will present detailed results and compare them to the previous work done on the same texts chosen by the authors of [26] and [27].

The main goal of our work and automatic indexing in general is to automate this work so that manual and human interventions are minimized. The outcome of our work is an enhanced software of the previously produced by the authors of [26] in the field of automatic indexing where the integration of the thesaurus presented more accurate results.

In this chapter we will exhibit briefly the workflow of the software produced by the authors of [26] and [27] as our work was built on top of it. We will then proceed with describing the process we adopted during the implementation and how we integrated our proposition with the latter. Following that, the experimental results will be presented and compared with the previous work done. The improvement shown will induce the importance and effectiveness of our solution where our main objective is to additionally retrieve significant words that could not have been chosen by the previous automatic indexers due to some limitation presented, where when bypassed led to an increase in accuracy.
5.1 Previous Implementations

5.1.1 Workflow and Technical Aspect

Since the work is mainly related to words and data manipulation, the most suitable data structure is the use of arrays and classes. The software followed an object-oriented design. For example, the class “Word” has two attributes:

1. Word - string of characters for each word in the text
2. Distance - integer that represents the distance of the term

The second class “Document” class contains the full document, it has three attributes:

1. Words - array having objects of the first class “Word” containing all words of the current text
2. CountWords - integer holding the occurrence of the word
3. AverageIdealDistance - integer used to measure the interval value of the words

After running the calculation function, the index words are now placed in a list of strings which will be later used as an input for the second phase proposed by [27] in order to generate the set of words that are frequently found together in texts of similar topic or category. Following that, the auto-indexer will perform another iteration taking into consideration the set generated. More details can be found in the original work in [27].

At a higher level, the above was implemented in order to determine the weight for each word. After calculation is performed for each word, the weight is saved and then the words having the highest five weights are chosen as indexes. We will now present the results of this implementation.
5.1.2 Results

According to the research and papers read throughout the process of this thesis, we conclude that the work done by the authors of [26] and [27] is considered to be a remarkable and advanced approach in the area Arabic automatic text indexing. It did not only focus on the occurrence of the word but also took into consideration the count of the stems, the word’s spread throughout the text in addition to determining new sets of terms that are found to frequently jointly appear when given documents belonging to the same topic. Below in figure 9 is the detailed result obtained by the mentioned implementations:

![Figure 9: Full results of [26] and [27] respectively](image-url)

Below you will find the explanation of the terminologies and annotation found in figure 9 which were also used in our implementation.

1. N: Total count of words in a text
2. I: The index words extracted through manual indexing

3. RR: Retrieved Relevant index words retrieved using the program implemented

4. RI: Retrieved Irrelevant index words retrieved using the program implemented

5. NRR: Not Retrieved Relevant words; NRR = I – RR

6. Precision = RR / (RR + RI); In Information Retrieval, precision is the proportion of relevant instances among all extracted specimen

7. Recall = RR / (RR + NRR) = RR / I; In IR, recall is the proportion of the total amount of relevant instances that were actually fetched

8. F-Measure = (2 * Precision * Recall) / (Precision + Recall); It provides a way to combine both precision and recall into a single measure that captures both properties and gives a way to express both concerns with a single score.

Words that were chosen by the auto-indexer and also found in the manually indexed list are categorized as retrieved relevant (RR). While words that were only chosen by the auto-indexer and do not have a match in the manually indexed list were flagged as retrieved irrelevant (RI). At last, words that were only found in the manually indexed list and the auto-indexer failed to retrieve were counted and placed in the not retrieved relevant field (NRR). While the F-Measure is an indicator combining both Precision and Recall in a single number.

5.1.3 Results Analysis

To facilitate the analysis process, a summary of the obtained results is presented in below figure 10 showing the average outcome of each work:
The experiments main focus was founded on the calculation of 2 indicators responsible of validating the effectiveness of the approaches. These indicators are: the Precision and the Recall. As mentioned earlier, the precision measures the percentage of relevant index words retrieved out of all the words retrieved. While the recall measures the percentage of relevant words retrieved in comparison to the manually chosen index words. The latter measures the retrieval accuracy of the relevant index words. The results in the figure 10 show that the approach adopted by the authors of [27] (Nasrallah’s work) which was based on the use of the Apriori algorithm was able to ameliorate both factors in comparison to the work done by [26] (Daher’s work). Hence, leading to a decrease in the NRR as desired. As a result, the integration proposed by the authors of [27] benefiting from the relation and correlation found between the words did enhance the recall measure by six percent and the precision by two percent. And as a global result, the F-measure was increased by 0.03 to show improvement in the correlation between precision and recall. In the following section we will present our results and compare them with the above mentioned work.

5.2 Our Implementation

As a recapitulation, our method was an enhancement of the previous implementations where we extended the limitation of benefiting from the relationship of texts that only belong to the same topic to generalize the concept and building a thesaurus catering for all possible topics.
5.2.1 Workflow and Technical Aspect

In our work, we kept the same data structure, but enforced an additional step which is to refer to the total sum of the grouped words instead of individual count. This was possible as we are now reading from the thesaurus file the words belonging to the same key term, and taking that into consideration before displaying the final count. Since as we mentioned, words are now grouped in a pair of key term having under it the synonyms. We will exhibit in the next section the newly produced results.

5.2.2 Results

We used the same texts tested in [26] and [27] to be able to properly assess the enhancement presented and eliminate any unwanted variable that may play a role in altering the test results. Our tested Arabic texts are 25 in total and are related to the Oil and Gaz in the Arab world.

<table>
<thead>
<tr>
<th>Arabic Text</th>
<th>N</th>
<th>I</th>
<th>RR</th>
<th>EI</th>
<th>NER</th>
<th>Recall (RR)</th>
<th>Precision (RR-RI)</th>
<th>F-Measure</th>
<th>Improvement Recall</th>
<th>Improvement Precision</th>
<th>Improvement F-Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test 1</td>
<td>413</td>
<td>71</td>
<td>64</td>
<td>12</td>
<td>7</td>
<td>90%</td>
<td>84%</td>
<td>0.87</td>
<td>10%</td>
<td>7%</td>
<td>11%</td>
</tr>
<tr>
<td>Test 2</td>
<td>443</td>
<td>90</td>
<td>71</td>
<td>12</td>
<td>19</td>
<td>79%</td>
<td>86%</td>
<td>0.82</td>
<td>8%</td>
<td>8%</td>
<td>11%</td>
</tr>
<tr>
<td>Test 3</td>
<td>459</td>
<td>84</td>
<td>68</td>
<td>17</td>
<td>16</td>
<td>81%</td>
<td>80%</td>
<td>0.80</td>
<td>7%</td>
<td>3%</td>
<td>7%</td>
</tr>
<tr>
<td>Test 4</td>
<td>466</td>
<td>92</td>
<td>63</td>
<td>12</td>
<td>29</td>
<td>68%</td>
<td>84%</td>
<td>0.75</td>
<td>5%</td>
<td>11%</td>
<td>11%</td>
</tr>
<tr>
<td>Test 5</td>
<td>475</td>
<td>95</td>
<td>81</td>
<td>12</td>
<td>14</td>
<td>85%</td>
<td>87%</td>
<td>0.86</td>
<td>8%</td>
<td>7%</td>
<td>10%</td>
</tr>
<tr>
<td>Test 6</td>
<td>477</td>
<td>89</td>
<td>75</td>
<td>12</td>
<td>14</td>
<td>84%</td>
<td>86%</td>
<td>0.85</td>
<td>7%</td>
<td>12%</td>
<td>12%</td>
</tr>
<tr>
<td>Test 7</td>
<td>484</td>
<td>103</td>
<td>87</td>
<td>15</td>
<td>16</td>
<td>84%</td>
<td>85%</td>
<td>0.85</td>
<td>5%</td>
<td>9%</td>
<td>5%</td>
</tr>
<tr>
<td>Test 8</td>
<td>488</td>
<td>71</td>
<td>60</td>
<td>13</td>
<td>11</td>
<td>85%</td>
<td>82%</td>
<td>0.83</td>
<td>4%</td>
<td>8%</td>
<td>8%</td>
</tr>
<tr>
<td>Test 9</td>
<td>502</td>
<td>82</td>
<td>68</td>
<td>28</td>
<td>14</td>
<td>83%</td>
<td>71%</td>
<td>0.76</td>
<td>2%</td>
<td>4%</td>
<td>5%</td>
</tr>
<tr>
<td>Test 10</td>
<td>508</td>
<td>87</td>
<td>73</td>
<td>15</td>
<td>14</td>
<td>84%</td>
<td>83%</td>
<td>0.83</td>
<td>5%</td>
<td>6%</td>
<td>7%</td>
</tr>
<tr>
<td>Test 11</td>
<td>509</td>
<td>75</td>
<td>62</td>
<td>17</td>
<td>13</td>
<td>83%</td>
<td>78%</td>
<td>0.81</td>
<td>12%</td>
<td>12%</td>
<td>18%</td>
</tr>
<tr>
<td>Test 12</td>
<td>520</td>
<td>88</td>
<td>71</td>
<td>15</td>
<td>17</td>
<td>85%</td>
<td>83%</td>
<td>0.82</td>
<td>8%</td>
<td>13%</td>
<td>14%</td>
</tr>
<tr>
<td>Test 13</td>
<td>531</td>
<td>92</td>
<td>83</td>
<td>22</td>
<td>9</td>
<td>90%</td>
<td>79%</td>
<td>0.84</td>
<td>7%</td>
<td>8%</td>
<td>10%</td>
</tr>
<tr>
<td>Test 14</td>
<td>564</td>
<td>80</td>
<td>73</td>
<td>18</td>
<td>7</td>
<td>91%</td>
<td>80%</td>
<td>0.85</td>
<td>8%</td>
<td>14%</td>
<td>15%</td>
</tr>
<tr>
<td>Test 15</td>
<td>592</td>
<td>91</td>
<td>70</td>
<td>22</td>
<td>21</td>
<td>77%</td>
<td>76%</td>
<td>0.77</td>
<td>10%</td>
<td>13%</td>
<td>18%</td>
</tr>
<tr>
<td>Test 16</td>
<td>613</td>
<td>86</td>
<td>81</td>
<td>19</td>
<td>5</td>
<td>94%</td>
<td>81%</td>
<td>0.87</td>
<td>11%</td>
<td>15%</td>
<td>18%</td>
</tr>
<tr>
<td>Test 17</td>
<td>643</td>
<td>97</td>
<td>73</td>
<td>23</td>
<td>24</td>
<td>75%</td>
<td>76%</td>
<td>0.76</td>
<td>10%</td>
<td>17%</td>
<td>22%</td>
</tr>
<tr>
<td>Test 18</td>
<td>680</td>
<td>116</td>
<td>81</td>
<td>8</td>
<td>24</td>
<td>70%</td>
<td>91%</td>
<td>0.79</td>
<td>14%</td>
<td>14%</td>
<td>22%</td>
</tr>
<tr>
<td>Test 19</td>
<td>799</td>
<td>95</td>
<td>87</td>
<td>20</td>
<td>8</td>
<td>92%</td>
<td>81%</td>
<td>0.86</td>
<td>8%</td>
<td>12%</td>
<td>14%</td>
</tr>
<tr>
<td>Test 20</td>
<td>856</td>
<td>118</td>
<td>104</td>
<td>37</td>
<td>14</td>
<td>88%</td>
<td>74%</td>
<td>0.80</td>
<td>12%</td>
<td>14%</td>
<td>20%</td>
</tr>
<tr>
<td>Test 21</td>
<td>888</td>
<td>130</td>
<td>113</td>
<td>31</td>
<td>17</td>
<td>87%</td>
<td>78%</td>
<td>0.82</td>
<td>12%</td>
<td>16%</td>
<td>22%</td>
</tr>
<tr>
<td>Test 22</td>
<td>900</td>
<td>120</td>
<td>101</td>
<td>31</td>
<td>19</td>
<td>84%</td>
<td>77%</td>
<td>0.80</td>
<td>8%</td>
<td>18%</td>
<td>21%</td>
</tr>
<tr>
<td>Test 23</td>
<td>983</td>
<td>155</td>
<td>125</td>
<td>34</td>
<td>30</td>
<td>81%</td>
<td>79%</td>
<td>0.80</td>
<td>8%</td>
<td>15%</td>
<td>21%</td>
</tr>
<tr>
<td>Test 24</td>
<td>1157</td>
<td>202</td>
<td>149</td>
<td>35</td>
<td>53</td>
<td>74%</td>
<td>81%</td>
<td>0.77</td>
<td>12%</td>
<td>15%</td>
<td>25%</td>
</tr>
<tr>
<td>Test 25</td>
<td>1549</td>
<td>183</td>
<td>163</td>
<td>43</td>
<td>20</td>
<td>85%</td>
<td>79%</td>
<td>0.84</td>
<td>4%</td>
<td>23%</td>
<td>24%</td>
</tr>
</tbody>
</table>

Average | 13.4 | 0.83 | 0.80 | 0.82 | 0.80 | 8% | 12% | 19% |

Figure 11: Summary of our results using the Thesaurus Integration Method

43
In the above figure 11 we present our results in the same format found in the previous work. The first column has the text number reference, the second column has the total number of words in the document, the third column has the number of manual indexes associated with the text. The fourth, fifth, and sixth columns contain the count of indexes associated with Retrieved Relevant, Retrieved Irrelevant and Not Retrieved Relevant respectively. The calculation of Recall, Precision and F-measure is performed in the seventh, eighth and ninth column. Finally the improvement percentage in comparison to [27] of both factors is calculated in the tenth and eleventh columns. While the F-measure improvement is found in the last column. At the bottom of the figure, the average results of the columns of our interest is calculated.

5.2.3 Results Analysis

The results in figure 11 demonstrate that our presented approach extracts on average 81 percent of the relevant index words from the full list of terms retrieved, this is the Precision indicator. To clarify further, our method’s false negative is less than 20 percent. As for the Recall, we are able to retrieve 83 percent of the terms that are manually chosen (humanly selected) as index words. We analyzed where our method performed best and found out that in texts that are rich in synonyms, our method was more than 90 percent accurate.

In addition, we conducted a comparison between our results and the results obtained by the authors of [27] who only implemented the Apriori algorithm without the thesaurus integration method. The following observation was noticed: The Recall was around 76 percent in the previous work, while the Precision was equivalent to 69 percent. As for our numbers, the Thesaurus integration proposed approach improved the recall by 8 percent and the precision by 12 percent on the previous solution adopted by [27] with a total of 10 and 14 percent respectively on the initial work presented by [26]. For a global comparison, the F-Measure captures both properties (Precision and Recall) and gives a way to express both
concerns with a single score. As an average, we reached a 0.82 score in the F-Measure, with a fifteen percent improvement. In addition, the Not Retrieved Relevant rate was reduced by around 8 percent, where our method only missed retrieving 17 indexes on average, which is considered a remarkable improvement.

Our approach successfully enhanced the percentage of all factors without a noticeable drawback, which can be considered a prodigious improvement in the study of automatic indexing of Arabic texts. Many of the terms and words that did not appear frequently but are significant to the text’s subject are now chosen with the remaining indexes. Which led to an increased number of the relevant candidate terms. Hence, a wider variety of correlated words is extracted allowing the user to properly and efficiently determine the text’s topic and index words using the methodology presented and implemented in this thesis.

<table>
<thead>
<tr>
<th></th>
<th>Daher’s Work</th>
<th>Nasrallah’s Work</th>
<th>Our proposed Solution</th>
<th>Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision</td>
<td>67%</td>
<td>69%</td>
<td>81%</td>
<td>12%</td>
</tr>
<tr>
<td>Recall</td>
<td>70%</td>
<td>76%</td>
<td>83%</td>
<td>7%</td>
</tr>
<tr>
<td>Not Retrieved Relevant (# Words)</td>
<td>31</td>
<td>26</td>
<td>17</td>
<td>-9</td>
</tr>
<tr>
<td>F-Measure</td>
<td>0.68</td>
<td>0.71</td>
<td>0.82</td>
<td>15%</td>
</tr>
</tbody>
</table>

Figure 12: Comparison of our results using the Thesaurus Integration Method with the previous methods

We have summarized the results outcome in the above figure 12 to visualize the improvement our method provided.
Chapter 6

Conclusion

The automatic indexing topic has acquired a lot of focus in the past couple of decade due to the technological advancement and the need to process a huge amount of data in a fast and efficient way. Automatic detection of topics, issues and subjects is now widely used on social media, especially by companies like Facebook and Twitter that are thriving to get accurate results of statuses, tweets and comments being posted to determine any potential threat or violations to their policies. Due to globalization and ease of Internet access now to most people living on the planet, Arabic, a language used by many has also emerged on social media. And due to the complexity of this language, a very robust method of auto-indexing is needed to tackle several topics where some of them were mentioned in this work.

In our work, we presented a solution for the “Automatic Indexing of Arabic Texts” problem. We introduced a thesaurus for a better content understanding leading to a wider selection of words and synonyms to be indexed. Words that frequently appear together and contribute to the same meaning can now be identified as relevant indexes even if they were not abundantly found in the text. This solution highly depends on a well built thesaurus which should be dynamically and easily expanded in order to consider more terms in the future. The consolidation and development of the several methods presented including stemming, distance calculation, weight calculation, association of words and referring to a
thesaurus for indexing is believed to be an innovative methodology adopted to ameliorate the techniques of auto-indexing of Arabic texts and can be considered an important innovation and contribution to the field.

6.1 Further Work

Even if the work presented in this thesis is considered as a major improvement but there is always a window for enhancement due to the fast pace of technology and advancement in our days. Some propositions will be presented in the following sections:

6.1.1 Include additional stemming rules

The Arabic language being a very complex language has many rules and irregularities. Not all grammatical rules and restrictions were implemented in this work as it would require more experienced resources in the Arabic language to achieve this point. For example, more words can be added to the M-derivations, T-Derivations and irregular plural tables presented earlier as it would lead to better stemming results. In addition, further improvement can be considered on the stop-list by introducing additional words that should not be stemmed such as including all countries, cities, people names, etc...

6.1.2 Diacritization Analysis

In the Arabic language there are certain symbols called “7arakat” that are inserted to the word in order to dictate its pronunciation. Diacritization presents many benefits since the same word having the same number of characters can have several different meaning in case the “7arakat” are altered. This leads to being able to properly determine if the word being assessed is a noun, a verb or should be considered in the stop-list term after understanding the real meaning of the word. This will also lead to a better stemming outcome. Hence, adjusting
the software to be able to identify the different forms of a diacritized word will improve accuracy and content analysis.

6.1.3 Enhance the built thesaurus

According to the author of [27], using a thesaurus contributes immensely to the outcome of an auto-indexer due to its ability to link the words together. In this work a robust thesaurus was built and since it can be dynamically maintained, more words should be added to cater for more synonyms leading to a wider choice of indexes to be considered. Another proposition would be to try and integrate this thesaurus with google search or merriam webster dictionary for instant checking instead of having to save and maintain the words in the DB where the output will be relatively enormous.
Bibliography


[18] K. T. Durant and M. D. Smith. Predicting the political sentiment of web log posts using supervised machine learning techniques coupled with feature


[38] P. Melville, W. Gryc, and R. D. Lawrence. Sentiment analysis of blogs by combining lexical knowledge with text classification. In *Proceedings of the*


[49] C. Schneider. The biggest data challenges that you might not even know you have. *IBM Watson*, 2016.


Appendix A

List of Prefix and Suffix

As mentioned earlier, the list of attached pronouns is a finite one. The following tables contain all possible attached pronouns. The first one contains a list of four prefix pronouns. As for the second and third tables contain the list of all suffix pronouns along with their possible combinations.

<table>
<thead>
<tr>
<th>prefix</th>
<th>active/passive</th>
<th>active/passive</th>
<th>active/passive</th>
</tr>
</thead>
<tbody>
<tr>
<td>يفعل/ستفعل</td>
<td>ألف-للناطق المتكلم</td>
<td>ألف-للناطق المتكلم</td>
<td>ألف-للناطق المتكلم</td>
</tr>
<tr>
<td>تفعل/ستفعل</td>
<td>الثائر للناطق المتكلم</td>
<td>التأي للفتاة والنجوم</td>
<td>النون للناطق المتكلم</td>
</tr>
</tbody>
</table>

Figure 13: List of prefix pronouns

<table>
<thead>
<tr>
<th>suffix</th>
<th>suffix</th>
<th>suffix</th>
<th>suffix</th>
</tr>
</thead>
<tbody>
<tr>
<td>فعلوا</td>
<td>ألفا</td>
<td>الألف-للناطق المتكلم</td>
<td>فعالا</td>
</tr>
<tr>
<td>فعلت</td>
<td>للفتاة والنجوم</td>
<td>الفعال للناطق المتكلم</td>
<td>الفعال للناطق المتكلم</td>
</tr>
<tr>
<td>فعلت</td>
<td>للفتاة والنجوم</td>
<td>الفعال للناطق المتكلم</td>
<td>الفعال للناطق المتكلم</td>
</tr>
<tr>
<td>فعلت</td>
<td>للفتاة والنجوم</td>
<td>الفعال للناطق المتكلم</td>
<td>الفعال للناطق المتكلم</td>
</tr>
<tr>
<td>فعلت</td>
<td>للفتاة والنجوم</td>
<td>الفعال للناطق المتكلم</td>
<td>الفعال للناطق المتكلم</td>
</tr>
<tr>
<td>فعلت</td>
<td>للفتاة والنجوم</td>
<td>الفعال للناطق المتكلم</td>
<td>الفعال للناطق المتكلم</td>
</tr>
<tr>
<td>فعلت</td>
<td>للفتاة والنجوم</td>
<td>الفعال للناطق المتكلم</td>
<td>الفعال للناطق المتكلم</td>
</tr>
<tr>
<td>فعلت</td>
<td>للفتاة والنجوم</td>
<td>الفعال للناطق المتكلم</td>
<td>الفعال للناطق المتكلم</td>
</tr>
<tr>
<td>فعلت</td>
<td>للفتاة والنجوم</td>
<td>الفعال للناطق المتكلم</td>
<td>الفعال للناطق المتكلم</td>
</tr>
<tr>
<td>فعلت</td>
<td>للفتاة والنجوم</td>
<td>الفعال للناطق المتكلم</td>
<td>الفعال للناطق المتكلم</td>
</tr>
<tr>
<td>فعلت</td>
<td>للفتاة والنجوم</td>
<td>الفعال للناطق المتكلم</td>
<td>الفعال للناطق المتكلم</td>
</tr>
<tr>
<td>فعلت</td>
<td>للفتاة والنجوم</td>
<td>الفعال للناطق المتكلم</td>
<td>الفعال للناطق المتكلم</td>
</tr>
<tr>
<td>فعلت</td>
<td>للفتاة والنجوم</td>
<td>الفعال للناطق المتكلم</td>
<td>الفعال للناطق المتكلم</td>
</tr>
<tr>
<td>فعلت</td>
<td>للفتاة والنجوم</td>
<td>الفعال للناطق المتكلم</td>
<td>الفعال للناطق المتكلم</td>
</tr>
<tr>
<td>فعلت</td>
<td>للفتاة والنجوم</td>
<td>الفعال للناطق المتكلم</td>
<td>الفعال للناطق المتكلم</td>
</tr>
<tr>
<td>فعلت</td>
<td>للفتاة والنجوم</td>
<td>الفعال للناطق المتكلم</td>
<td>الفعال للناطق المتكلم</td>
</tr>
<tr>
<td>فعلت</td>
<td>للفتاة والنجوم</td>
<td>الفعال للناطق المتكلم</td>
<td>الفعال للناطق المتكلم</td>
</tr>
<tr>
<td>فعلت</td>
<td>للفتاة والنجوم</td>
<td>الفعال للناطق المتكلم</td>
<td>الفعال للناطق المتكلم</td>
</tr>
<tr>
<td>فعلت</td>
<td>للفتاة والنجوم</td>
<td>الفعال للناطق المتكلم</td>
<td>الفعال للناطق المتكلم</td>
</tr>
</tbody>
</table>

Figure 14: List of suffix pronouns
<table>
<thead>
<tr>
<th>جعلناه - رأيتك</th>
<th>ـه/كما</th>
<th>فعلنا</th>
</tr>
</thead>
<tbody>
<tr>
<td>فعلاهما - علموكما</td>
<td>ـهم/كم</td>
<td>فعلنا</td>
</tr>
<tr>
<td>أطيعتهم - داعبكم</td>
<td>ـه/كم</td>
<td>فعلنا</td>
</tr>
<tr>
<td>جعلناه - أحببتكن</td>
<td>ـهن/كن</td>
<td>فعلنا</td>
</tr>
</tbody>
</table>

Figure 15: List of the combination of prefix and suffix pronouns
Appendix B

Case Study

مليون برميل إنتاج البحرين من النفط
السماحة - الهيئة الوطنية للنفط والغاز

بلغ إجمالي إنتاج النفط في مملكة البحرين والذي يتمثل على (حلل البحرين وحلل أبوعسفة) للقراءة مابين
يناير/كانون الثاني و سبتمبر/أيلول الماضي 49,445 مليون برميل، ولاحظ أن إنتاج النفط خلال شهر
سبتمبر من حلل البحرين وحلل أبوعسفة هو الأفضل مقارنة مع معدل الإنتاج في الأشهر الأخرى من العام
2010.

وبلغ معدل البحرين للمعادل اليومي للفترة من هناك حتى شهر سبتمبر من العام الماضي (2010) 32 ألف برميل وهو
المعادل اليومي نفسه لإنتاج لشهر سبتمبر من عام 2009. ما وضعه في تقرير الإحصاءات الإجمالية الذي أصدرته الهيئة الوطنية للنفط والغاز، الخاصة بإنتاج النفط والغاز، والواردات النفطية وكذلك المشتقات النفطية والمنتجات المحلية للحالة مابين يناير/كانون الثاني و سبتمبر/أيلول 2010.

ويلتزم أن معدل الإنتاج اليومي لشهر سبتمبر من العام 2010، أفضل من المعدل اليومي لشهر ينار من العام
الجاري الذي بلغ 30 ألف برميل يومياً.

ومنها ينضخ النفط الخام المستورد من المملكة العربية السعودية فقد بلغ 64,253 مليون برميل في العام 2010،
بالمقارنة مع 60,171 مليون برميل للشهر نفسه من العام 2009، أي زيادة قدرها 4,082 مليون برميل وهي
تعادل 6.8 في المئة. وقد بلغ المعدل اليومي للنفط الخام المستورد 236 ألف برميل يومياً، مقارنة مع 220 ألف
برميل يومياً خلال الفترة نفسها من العام 2009. وتعود أسباب زيادة في الكميات المستوردة من النفط إلى
الطلب على المشتقات النفطية المكررة بمصنع التكرير بالمملكة. أما النفط المحلي الذي تم ضخه إلى مصفاة
التكرير خلال النسخة من العام قد وصل إلى 7,270 مليون برميل، بالإضافة إلى 69,751 مليون برميل برميل خلال العام 2009، أي زيادة نسبتها 4.3 في المئة.

وبالتالي النفط الخام الذي تم ضخه إلى المصفاة في النفط المستورد من المملكة العربية السعودية والذي يشكل 89
في المئة وكذلك المنتج من حلل البحرين والذي يشكل 11 في المئة. وقد بلغ المعدل اليومي للنفط الخام الذي تم
ضخه إلى مصنع التكرير 266,410 برميل، مقارنة مع 255,501 برميل خلال الفترة نفسها من العام 2009.

ونتيجة أسباب الزيادة إلى زيادة إنتاج مصفاة البحرين لتلقي الغاز على المشتقات النفطية.

أما إنتاج مصفاة البحرين خلال النسخة من العام 2010، فقد بلغ 74,313 مليون برميل مقارنة مع
70,722 مليون برميل خلال العام 2009، وبعكس يومي قدره 272 ألف برميل، مقارنة مع 259 ألف برميل
المعدل البحري للشهر نفسه من العام 2009؛ إذ تبلغ نسبة الزيادة 5.1 في المئة.
In this text, the author of [28] had a 66 percent precision and 81 percent recall. While the author of [27] who used the Apriori algorithm and generated the frequent sets improved to 68 percent precision and 84 percent recall. While our method presented an 81 percent precision and 91 percent recall with a 12 percent and 7 percent improvement respectively. We were able to additionally retrieve 8 relevant words which are:

The number of words in this text Document denoted by N is 799 words.
The previous implementations retrieved the word “Bahrain” but failed to retrieve “Manama” which is its capital as it was not redundant in the text. But when we introduced a thesaurus having sentimental understanding this additional words was successfully retrieved and indexed. The same applies to the remaining words which are synonyms of words available in the text but were not previously retrieved. The word “Gasoline” for example was only found once in the text, but it falls under the resources category “Mawared” and is highly relevant to the text. The previous methods failed to retrieve it while we successfully did. The below shows how the new counting method was performed, from it we retrieved the new “synonym count” factor and recalculated the weight:
Regarding our precision improvement it was due to the elimination of the numbers and additional stop words added to the previous list. For example the previous implementations retrieved “255.501”, “2010”, “2009” and several other numbers found in the text. Those retrievals are not important and do not present any benefits to the indices list.