http://dx.doi.org/10.21622/ACE.2023.03.1.001

An Innovative Automatic Indexing Method For Arabic Text

Ramzi A. Haraty[1], Sanaa Kaddoura[2], Sultan Al Jahdali[3] And Nour K. Masri[4]

[1] Department of Computer Science and Mathematics, Lebanese American University, Beirut, Lebanon
[2] Zayed University, Abu Dhabi, United Arab Emirates
[3] Department of Computer Science, College of Computers and Information Technology, Taif University
[4] Department of Computer Science and Mathematics, Lebanese American University, Beirut, Lebanon

Emails: {rharaty@lau.edu.lb, sanaa.kaddoura@zu.ac.ae, aljahdali@tu.edu.sa, nour.masri@lau.edu}

Received on, 22 December 2022 - Accepted on, 17 February 2023 - Published on, 20 May 2023

ABSTRACT

The study of automatic indexing and text retrieval methods for language has a long history. Automatic indexing involves extracting words from a document to categorize it based on subject matter and to improve the information retrieval process. Despite extensive research in other languages, there remains limited investigation into automated Arabic text categorization. In this research, the researchers introduce an innovative method to enhance the accuracy of automatic indexing of Arabic texts by incorporating a thesaurus. Their approach extracts new relevant words by referencing thesaurus, which contains words, synonyms, and correlations identified through its construction using a natural language toolkit and a WordNet library.

Synonyms with similar meanings that frequently appear together are grouped using a JavaScript Object Notation dictionary. The research results demonstrate a significant improvement in accuracy and efficiency compared to prior studies.

Index Terms: *Arabic Text, Automatic Indexing, Building Thesaurus, Frequent Sets, Synonyms.*

INTRODUCTION

Indexing text documents involves analyzing their content to identify the subject matter. It is crucial for information retrieval and has become more complex as the length of the text increases. This process is utilized in a variety of domains, including tabloid articles, online editorials, and other types of text. The widespread use of search engines such as Bing illustrates the importance of indexing texts for information retrieval. According to a study by Statista Research Department, 5.3 billion people were using the Internet in 2022 [1]. Despite the growth of technology, the Arabic language and literature continue to hold significance, especially in Arab countries.

From an academic viewpoint, managing, categorizing, and identifying different types of online content, such as articles, documents, conferences, and social media interactions, have become crucial due to the huge number of Internet users. Bergman's article highlights that the deep web, which is part of the World Wide Web but contains un-indexed content, is estimated to be 500 times larger than the indexed Web [2]. As the fastest growing category of information on the Internet, it is not accessible through conventional search engines. In the era of AI and machine learning, indexing texts is essential for retrieving relevant information quickly. Smart knowledge depends on keywords and relevant terms.

http://dx.doi.org/10.21622/ACE.2023.03.1.001

Text document indexing involves analyzing its content to determine its topic. Creating indexes creates a reference for data, making it easier and faster to locate relevant information. Historically, this process was done manually by experts in vocabulary, grammar, and analysis. However, manual indexing is both time-consuming and expensive, and the results can be subjective due to personal bias in sensitive topics like politics and history. With the growing volume of documents, manual indexing has become a challenge. Additionally, indexing is now used in various applications such as archiving, spam detection, automatic message routing, page content filtering, and more, leading to a demand for more efficient automatic indexing techniques.

A. Methodology and Contribution

The proposed approach - Thesaurus Integration - is an enhanced and innovative method for extracting more accurate indexes. The main idea behind the proposed method is to take into consideration synonyms available within the text and key terms that are not necessarily 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) Adding new terms to the Stop List
3) Identifying terms that do not require stemming

In fact, introducing a thesaurus presented a remarkable improvement. Moreover, to enhance further, the researchers have worked on adding more words and elements to the "Stop Words list" which were not previously considered and were leading to false positives. In addition, they have also identified words that should be excluded from being stemmed and added them as an exception while processing the related algorithm such as common and proper nouns.

B. Motivation

The Arabic language, being the mother tongue of Arabs living in the Middle East and North Africa, in addition to being a very complex and rich language, has made the researchers eager to tackle this topic. Feelings aside, and from a technical perspective, the below were also crucial arguments behind investing in this research:

1) To index Arabic documents available in both the global network and the un-indexed Deep Web is of extensive need.
2) To keep abreast with the emerging of Artificial Intelligence (AI) and Internet of Things (IoT) nowadays and benefiting from lower error rates.
3) To benefit from the rich Arabic research and documents.
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 [3]. 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 twenty synonyms which

http://dx.doi.org/10.21622/ACE.2023.03.1.001

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 twenty 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). However, 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 umbrella and extract more relevant indexes. This leads to a better comprehension of the text being analyzed.

In this work, the researchers highlight the importance of the work previously done on automatic indexing of Arabic text and present their proposed solution.

The remainder of the paper is organized as follows: Section 2 provides the background of the work. Section 3 tackles the related work done on automatic indexing of documents. Section 4 highlights the importance of stemming and weight assignment, which is the pre-processing phase to be able to proceed with thesaurus integration. Section 5 presents the researchers' proposed solution and the procedure implemented to choose the relevant words. Section 6 presents the implementation of their work as well as the previous work. In section 7, the researchers offer the experimental results; and in section 8 they offer a conclusion and a proposition for future work.

BACKGROUND

The researchers have identified two different types of indexing each based on a distinct concept: full-text indexing and thesaurus indexing[4].

A. Full-text Indexing

This type focuses on picking terms that are only found in the presented document. It disallows referencing words that are more commonly used or researched, if they are not present in the text. For example, given a sample text about Albert Einstein, usually the following terms are associated with Einstein: physics, relativity theory, quantum, etc. If the latter words are not mentioned in the article, even if highly representative of the topic, they 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. Thus, this presents limitations and weakens text enrichment.

B. 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 Einstein and his research by searching for "physics" or "relativity theory". The difficulty of a thesaurus base indexing is demonstrated in its implementation as it depends on lexicon understanding. From a technical perspective, it requires having a file which contains words and their corresponding 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

http://dx.doi.org/10.21622/ACE.2023.03.1.001

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 explored further in this work.

Regardless of the indexing method adopted, the outcome is the same: A 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 indicators in the subject heading:

- Name: The proper name of the individual, organization, or corporation the text is tackling, i.e., "Albert Einstein".
- Position: The social or political rank, i.e., "Researcher" or "Scientist".
- Location: Any Geographic Location: Country, City, Place, etc., i.e., "Germany" or "Austria".
- Activity: Reason behind the document, i.e., "Research Article".

The above information will be concatenated to produce the following sample subject heading: Albert Einstein > Researcher > Germany > Research Article. The produced information 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.

C. Importance of The Arabic Language

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 be-longed [5]. Even prior to the rise of Islam, Arabic literacy was of extreme importance as it was the medium of oral poetry. In fact, the poet was socially considered from the top ranked statuses one could reach. 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 [6].

Following the rise of Islam, Arabs and Muslims 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 [7].

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 [5]. The western population, who 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 [8].

Arabic belongs to a group of languages known as the Semitic language. The United Nations recognizes it as one of its six official languages [9]. It consists of twenty-eight letters and is written from right to left. The letters change their shapes and form depending on the position in the word 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 [10]. This rich language can construct complex and varied words from basic roots. It is enough to have three letters such as

http://dx.doi.org/10.21622/ACE.2023.03.1.001

'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 [11]. Since the Arabic language is not solely bounded to Arabic literacy but its role also expands to being:

- A contrivance of artistic and correct expression.
- A pillar of religion.
- An anchor of culture.
- A centerpiece of contemporary nationalism.

The researchers realize that the 276 million Arabic speaking population's culture, manuscripts and activities have their weight and is of great importance. This volume and the striking complexity require indeed a meticulous and diligent technique 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 build an automatic indexer is introduced to facilitate and strengthen this loaded area.

RELATED WORK

The aim of the work at hand is to build on and enhance the Automatic indexing system for Arabic texts previously built by the authors of [12] and then updated by [13]. The software previously developed extracts representative indexes 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 word association technique was introduced. However, this approach presented some constraints where the pre-requisite texts to be processed must be from the same category to extract the association ratio. For that, and to enhance the work, the researchers have decided to integrate a thesaurus that can incorporate and index a text regardless of its category. Hence, eliminating the previous constraint. They will discuss 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 thesaurus- b a s e d approaches.

A. Text Processing

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 comprehend how people interpret and utilize languages to try and build corresponding tools and invent techniques to solicit the same on computers and machines to carry out the appropriate tasks [14]. Currently, this is being applied in various fields of studies, including but not limited to machine translation, and text processing and summarization. In specific, one crucial area of NLP application is multilingual text processing that seeks to take advantage of the WWW and online libraries [14][15]. Many analysts have proposed to employ WordNet libraries to enhance the statistical analysis results of natural language texts [16][17]. 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

http://dx.doi.org/10.21622/ACE.2023.03.1.001

Italian, Dutch, Spanish, German and French during the late twentieth century. Arabic was introduced to WordNet in 2006 and later expanded in 2015 [18][19].

The author of [20] lists several NLP packages that are widely used:

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

b. InQuery that parses sentences, stems words and recognizes proper nouns and concepts based on term co-occurrence.

c. LinguistX parser from XEROX PARC that extracts syntactic information and is used in InfoSeek.

d. NetOwl from SRA, a text mining system.

In late 2018, a team of scientists from the Google AI Language lab led by J. Devlin introduced a new linguistic model called BERT [21]. This model is designed for deep pre-training of bidirectional text representations for use in machine learning models. BERT stands out for its ease of use, requiring only the addition of one output layer to existing neural architecture to achieve text models that surpass existing ones in various natural language processing problems.

To evaluate BERT's performance, the model was tested on various standard datasets after undergoing task-specific additional training. On the GLUE test (General Language Understanding Evaluation [22]), a suite of tasks and datasets that test natural language comprehension, the BERT-based model demonstrated an average improvement of 4.5% and 7% (for standard and large neural networks, respectively) compared to the best-known models." [23]

2) *Arabic Text Processing:* The authors in [24] presented the issue of recognizing Arabic handwritten characters which still pose challenges to the scientific society. 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.

To additionally improve stemming techniques, the authors in [26] 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 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.

http://dx.doi.org/10.21622/ACE.2023.03.1.001

In a more recent paper, the authors in [27] 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 can 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.

B. Automatic Indexing Characterized by Frequency and Words Spread

The authors of [13] presented a model that 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). These layers are:

1) *Read Whole Document and Exclude Stop Words and 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. The 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 confirms 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 [29].

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 Key terms:* Finally, when all weights are presented, the algorithm will identify the highest number to select the best ranked terms as representative indexes to the text.

C. 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 nowadays as it provides insights on data. Thus, automating this process became an interest for many researchers. Some of the methods adopted are Naïve Bayes (based on Bayes's Theorem) proposed by [30], Decision Trees by [31], Neural Networks by [32], and

http://dx.doi.org/10.21622/ACE.2023.03.1.001

Support Vector Machine (SVM) by [33].

In specific for Arabic language, the author of [34] 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.

The author of [35] implemented a tagger (tool that produces tags) that uses rule-based techniques along with statistical methods to automate the process. Like English, Arabic word types are categorized into several divisions: verbs, nouns, and particles. For better accuracy and grouping facilitation affixes were also removed.

In the work done by [36], 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 to classify and extract from it the corresponding 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 [13].

D. Thesaurus Based Approach

Within the framework of information retrieval (IR), a thesaurus is a means 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 [37]. From a linguistic perspective, 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 IR Systems (IRS), as it boosts precision and control of idioms [38]. It should ideally consist of a list of essential words related to a subject or a key term. The use of thesaurus has improved IRSs by 10 to 20 percent [39]. Using concepts and synonyms rather than just the available words in the text for automated indexing adds specificity to the document representation [40]. 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, the researchers will present the different thesaurus building methods done for both languages.

1) *Building Thesaurus for Arabic Language*: Arabic is a quite complex language. Among the complexities 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 [41], developing automatic Text Categorization (TC) for Arabic is a demanding, complex and requires a considerable amount of time to perform. Also, TC techniques for Arabic documents are not as efficient as it is for English due to its linguistic structure. Such reasons may justify the lack of research in

http://dx.doi.org/10.21622/ACE.2023.03.1.001

this domain.

The authors in [39] 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 stems of the words [7]. The authors in [42] proposed a new classifier for Arabic text categorization called FRAM. They divided the work into two stages: in the first stage, they pre-processed the texts that were already categorized, then they extracted the relevant keywords. In the second stage, they built a database from the feature terms. During testing, uncategorized documents had classified using FRAM and compared accuracy with other three Bayesian learning classifiers. FRAM outperformed these 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 selections and eliminating the lowest frequency of the presented. The authors in [43] 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 still considered small.

The authors in [44] implemented a semantic indexing and query method for IRS. The authors relied on Arabic WordNet as their 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 documents used was higher in all measures. On average, they obtained a 60 percent precision. The authors in [45] summarized in their paper several works done in Arabic Sentiment Analysis (SA).

Nowadays, scholars have gained a lot of interest in SA and it has currently become 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 points of view regarding a certain topic. In [46], the methodology adopted combined three different classifiers: Lexicon-based opinion classifier, maximum entropy method and k-nearest neighbors (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. The authors in [47] 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 small dataset, but they were able to reach 80 percent accuracy of content analysis and published the corpus online for other researchers to benefit from. Their work was mainly focused on Saudi dialect corpus that applied SA to twitter content to identify tweets polarity. The authors in [48] 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 the n-gram feature. However, the dataset was also of small size. The pre-processing phase included: normalization, tokenization, removing stop words, stemming, and filtering. In the same context, and in an

http://dx.doi.org/10.21622/ACE.2023.03.1.001

innovative method, the authors of [49] 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 then converted the Arabizi words into Arabic words 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.

STEM WORD EXTRACTION AND WEIGHT CALCULATION

Since this work is focused on enhancing the work done by the authors of [13], this section presents how the second and third layers in the mentioned work are 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, especially for the Arabic language. In addition, weight calculation is the methodology that will help the researchers 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 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 section.

A. 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 the researchers counting method. Since the Arabic language is very rich, a single word may have many grammatical variations such as:

- Singular or Plural
- Masculine or Feminine
- Definite or Indefinite Noun
- Adjective or Subject
- 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 their singular or plural forms, they should be treated as one entity instead of being segregated.

B. Identification Process

1) *Rhyming:* When reading a word, the researches need to first identify its type whether it is a verb or a noun. Since each type has a different rhyming, this will help them 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, the researchers will be able to identify whether the word is in its singular or

http://dx.doi.org/10.21622/ACE.2023.03.1.001

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 it will be returned to its correspondent root.

2) *Identifying Nouns and Verbs:* Like 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 previously mentioned, rhyming helps identify the type of the word, either a verb or a noun. For example, words that rhyme with "يفعل" (does), "افعل" (do) are verbs while words that rhyme with "فاعل" (doer) or "مفعول" (done) are nouns. In addition, another method to identify the word type is by inspecting the terms that precede it. For example, the pronouns "ال" 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 reasoning applies to nouns with a different set of guidelines. At last, if the above mentioned methods could not determine the word type, the researchers inspect the attached pronouns which are either found at the beginning or the end of the word; some of them only attach to verbs while others only to nouns. Algorithm 1 presented in Figure 1, summarizes the possibilities considered in the model and on what grounds characterization is being based.

```
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: 'ال', 'بال
                                ', 'كال', 'فال' then
          Return Noun;
 Else If Word Rhymes with 'فاعل' or 'مفعول'
          Return Noun;
 Else
          Return 'Unknown';
   }
```

Fig 1. Verb and noun identification

3) *Removing Stop-Lists Terms:* Words that do not add any significance to the text's content will be excluded from this study as our counting technique will be highly affected by those redundant terms. For example, the letter "و" 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 "لا". However, the latter helps one identify the words that follow as previously explained, nevertheless, they 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 this 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 the researchers 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 words that if excluded, presented more accurate results such as numbers, months, and special characters.

http://dx.doi.org/10.21622/ACE.2023.03.1.001

C. Extracting Stem Words

After identifying the type that each word belongs to, the researchers need to apply the correspondent 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 [50].

1) *Verb Stemming - Inspecting Attached Prefix and Suffix:* Pronouns Verbs in Arabic may have two 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 defined. Words will be checked against this list and the pronouns will be removed based on an algorithm responsible for pattern matching.

2) *Checking Verbs against the "Five Verbs":* According to the authors in [50][51][52], the "Five Verbs" are five standards 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". Nonessential 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, "يكتبون" (Verb for They Write) has the first and last two letters as nonessential, those letters 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) *Checking Verbs against the "Ten Verb Addition":* Similarly, to the previous set of verbs, the "Ten verbs" have also special properties with a different derivation formats that are built from a three-letter root. The derivations of those verbs 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. 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 along with an example is presented below in Figure 2.

الزيادات	مثال	أصل الفعل	الزيادات	مثال	أصل الفعل
أفعل	أضرم النيران	ضرم	إنفعل	إنهزم الأعداء	هزم
فعّل	سرّع البحث	سرع	إفتعل	إقترف خطأ فادحا	قرف
فاعل	قاتل الأعداء	قتل	إفعلّ	إزهرّ الورد	زهر
تفعّل	تسبّب في وفاته	سبب	إفعوعل	إغرورقت عيناه	غرق
تفاعل	تعاطف مع صديقه	عطف	إستفعل	إستخرج النفط	خرج

Fig. 2. List of ten derivations

4) *Noun Stemming:* Several factors render 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:

- Number: Singular, double, or plural form, whereas each format may have several exceptions,
- Gender: Male or female, and
- Derivations: where noun may have no specific format.

http://dx.doi.org/10.21622/ACE.2023.03.1.001

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 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.

After all the words in the document are properly stemmed, a certain weight should be given to each term. In [12] the weight relied on three factors, while in the researchers' proposd enhanced method, a new factor is introduced. The researchers will describe in the next section how weight is calculated.

D. 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. As mentioned in [12], 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.

The authors 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 the enhanced work at hand, the researchers added the "Synonym Count" abbreviated as "sym" in our formula as a factor making the total factors affecting the equation four components.

THE PROPOSED SOLUTION

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 work 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 [13] where the authors generated item-sets of recurrent and frequent patterns using Apriori, the researchers decided to benefit from this method to build their own enhanced approach. Apriori is an algorithm initially proposed by the authors in [53] which is designed for extracting item set that frequently appear together. It is achieved by mining and association rule learning over relational databases. To take further advantage of the link between words in a text, they decided to integrate a thesaurus containing wider options.

This approach is adopted following the fact that Arabic language is rich where several words and terminologies can be used to describe the same term. The

http://dx.doi.org/10.21622/ACE.2023.03.1.001

thesaurus that the researchers built will contain words along with their synonyms where each is grouped under its own category. They associate a "Key Term" to each mentioned category that represents the set of words contained under it. This will allow them 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 "نفط" or "Petrol" 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 "وقود" or "OPEC" might not surpass the threshold to be chosen as indexes. This is because they are not frequently found since their word count would be minimal. Nevertheless, those terms are extremely representative of the text and are in fact indexes to be considered. 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. In the next section, the researchers will detail the steps adopted which led to the enhanced solution.

E. Phases of the Proposed Solution

The proposed 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

1) *Stop-List Adjustment:* After examining the results shown in the previous work done by the authors of [13], the RI (Retrieved Irrelevant) index was considered relatively high. To improve it, the researchers analyzed the reason behind having this amount of retrieved irrelevant words and found out that many of the indexes retrieved were stop words that were not taken into consideration in the list used. Following this ascertainment, they have managed to identify an additional 210 words that if excluded, presented more accurate results. This adjustment has led to an improvement in the precision percentage.

2) *Thesaurus Building, Integration, and Management*: WordNet contains a function "Synsets()" that lists synonyms of a certain word. To build the researchers' thesaurus, they 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, they saved each entry in the JSON file and made sure to eliminate duplicate data. In addition, they included all the synonyms of a certain word if found under different key terms, together. Let us take the example of two words that are synonyms, if the synonyms of Word1 are listed first, the researchers will find that Word2 is in the list. If the same is applied for Word2, the researchers will also find that Word1 in the list since the function used lists all possible synonyms of the word. Hence, they made sure not to create two different entries in that case and group them all together.

http://dx.doi.org/10.21622/ACE.2023.03.1.001

Following the same strategy, the researchers were able to build a robust thesaurus that also included words that are logically related and not just literal synonyms. Figure 3 displays a sample retrieved from the researchers' thesaurus for "موارد" or Resources as a key term.

Fig. 3. Sample Thesaurus element

As shown, several terms that are related to the key term " موارد" were displayed under it such us "نفط" "كاز" "غاز" and " وقود". 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.

Figure 4 presents the thesaurus building algorithm and logic employed to ensure optimization.

```
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 unicodedata.normalize('NFKD', ar).casefold() == unicodedata.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 unicodedata.normalize('NFKD', synonym).casefold() == unicodedata.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
```

Fig. 4. Thesaurus building algorithm

http://dx.doi.org/10.21622/ACE.2023.03.1.001

After building the thesaurus, integrating it, and managing it, the researchers 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.

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. The researchers 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.

a) Adjusted formula: In the work done by the authors of [12], the following formula was adopted to calculate the weight of each word:

$$w = m \times sm \times f \quad (1)$$

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 the work at hand, the researchers added to this formula the "sym" 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:

$$w = m \times sm \times sym \times f \quad (2)$$

b) Adjusted counting method: To obtain the synonym count referred to as "sym" factor in the formula, the text's counting method was modified to group and count together words belonging to the same key term. The function will refer to the thesaurus file to know where the current word belongs and to which group of words it will be counted to be displayed correctly.

Figure 5 is an example extracted from one of the texts tested.

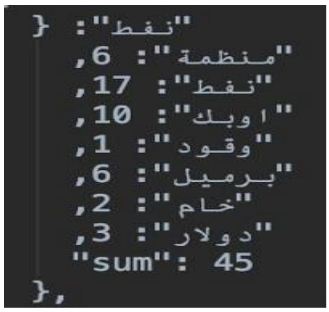

Fig. 5. New counting method

As shown, each word under "نفط" 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. 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:

$$OldWeight(Dollar) = 83.25 \quad (3)$$

http://dx.doi.org/10.21622/ACE.2023.03.1.001

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

NewWeight(Dollar) = 582.75 (4)

When adopting the proposed method, the word "Dollar" was successfully retrieved as a correct index after surpassing the threshold. The new retrievals led to an increased number of RR (Retrieved Relevant) indexes and eventually to a better recall percentage.

The proposed solution expands the output of the previously extracted as the researchers 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.

The selection proposed in this work is based on retrieving as much relative terms as possible but only choosing the highest indexes. This was achieved after all words were given the appropriate weights, and then the highest ones were picked. The proposed mechanism which presented a new way to find relation between those words gave a more accurate weight to the words. This led to a better index selection.

IMPLEMENTATIONS

The main goal of the work at hand and automatic indexing in general is to automate this work so that manual and human interventions are minimized. In this section, the researchers will briefly exhibit the workflow of the software produced by the authors of [13] as their work was based on it. They will then proceed with describing the process they adopted during the implementation and how they integrated their 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 the proposed solution where the main objective is to additionally retrieve significant words that could not have been chosen by the previous automatic indexers because of some limitation presented, where when bypassed led to an increase in accuracy.

A. Previous Implementaions

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.

http://dx.doi.org/10.21622/ACE.2023.03.1.001

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 to generate the set of words that are frequently found together in texts of a similar topic or category. Following that, the auto-indexer will perform another iteration taking into consideration the set generated.

At a higher level, the above was implemented 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 weights are chosen as indexes. We will now present the results of this implementation.

B. The Researchers' Implementation

1) *Terminologies and Annotations:* The following are the explanation of the terminologies and annotations used in the implementation.

- N: Total count of words in a text.
- I: The index words extracted through manual indexing.
- RR: Retrieved Relevant index words retrieved using the programimplemented.
- RI: Retrieved Irrelevant index words retrieved using the programimplemented.
- NRR: Not Retrieved Relevant words.

$$NRR = I - RR \quad (5)$$

- Precision: In information retrieval, precision is the pro- portion of relevant instances among all extracted specimens.

$$Precision = RR/(RR + RI) \quad (6)$$

- Recall: In information retrieval, recall is the proportion of the total amount of relevant instances that were fetched.

$$Recall = RR/(RR + NRR) = RR/I \quad (7)$$

- F-measure: 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. The F-measure is calculated as per the following formula:

$$(2 \times Precision \times Recall)/(Precision + Recall) \quad (8)$$

Words that were chosen by the auto-indexer and found in the manually indexed list are categorized as 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 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 field NRR.

2) *Workflow and Technical Aspect:* In the work at hand, the researchers 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 they 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. As mentioned earlier, words are now grouped in a pair of key terms, having under it the synonyms. The researchers exhibit in the next section the newly produced results.

http://dx.doi.org/10.21622/ACE.2023.03.1.001

RESULTS AND DISCUSSION

A. Results

To elaborate further and show the promising improvements the proposed method present, the researchers will give detailed results and compare them to the previous works. They used the same texts tested in [12] and [13] to be able to properly assess the enhancement presented and eliminate any unwanted variable that may play a role in altering the test results. The tested Arabic texts are 25 in total and are related to the Oil and Gaz in the Arab world.

In figure 7, the researchers present their 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 [13] 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 interest is calculated.

Arabic Text Input			Our Results								
Text#	N	I	RR	RI	NRR	Recall (RR/I)	Precision (RR/RR+RI)	F-Measure	Improvement Recall	Improvement Precision	Improvement F-Measure
1	413	71	64	12	7	90%	84%	0.87	10%	7%	11%
2	443	90	71	12	19	79%	86%	0.82	8%	8%	11%
3	459	84	68	17	16	81%	80%	0.80	7%	3%	7%
4	466	92	63	12	29	68%	84%	0.75	5%	11%	11%
5	475	95	81	12	14	85%	87%	0.86	8%	7%	10%
6	477	89	75	12	14	84%	86%	0.85	7%	12%	12%
7	484	103	87	15	16	84%	85%	0.85	4%	5%	5%
8	488	71	60	13	11	85%	82%	0.83	4%	8%	8%
9	502	82	68	28	14	83%	71%	0.76	2%	4%	5%
10	508	87	73	15	14	84%	83%	0.83	6%	7%	8%
11	509	75	62	17	13	83%	78%	0.81	12%	12%	18%
12	519	88	71	15	17	81%	83%	0.82	8%	13%	14%
13	531	92	83	22	9	90%	79%	0.84	7%	8%	10%
14	584	80	73	18	7	91%	80%	0.85	8%	14%	15%
15	592	91	70	22	21	77%	76%	0.77	10%	13%	18%
16	613	86	81	19	5	94%	81%	0.87	11%	15%	18%
17	643	97	73	23	24	75%	76%	0.76	10%	17%	22%
18	680	116	81	8	24	70%	91%	0.79	14%	14%	22%
19	799	95	87	20	8	92%	81%	0.86	8%	12%	14%
20	856	118	104	37	14	88%	74%	0.80	12%	14%	20%
21	888	130	113	31	17	87%	78%	0.82	12%	16%	22%
22	890	120	101	31	19	84%	77%	0.80	8%	18%	21%
23	981	155	125	34	30	81%	79%	0.80	8%	19%	21%
24	1157	202	149	35	53	74%	81%	0.77	12%	19%	25%
25	1349	183	163	43	20	89%	79%	0.84	4%	23%	24%
Average					17.4	0.83	0.81	0.82	8%	12%	15%

Fig. 6. Summary of our results using the Thesaurus Integration Method

B. Discussion

The results in figure 6 demonstrate that the 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, the proposed method's false

http://dx.doi.org/10.21622/ACE.2023.03.1.001

negative is less than 20 percent. As for the Recall, the researchers can retrieve 83 percent of the terms that are manually chosen as index words. They analyzed where their method performed best and found out that in texts that are rich in synonyms that this method was more than 90 percent accurate.

In addition, they conducted a comparison between their results and the results obtained by the authors of [13] 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 the numbers presented, the thesaurus integration proposed approach improved the recall by 8 percent and the precision by 12 percent. For a global comparison, the F-Measure captures both properties (Precision and Recall) and paves the way to express both concerns with a single score. As an average, the researchers reached a 0.82 score in the F-Measure, with a 15% improvement. In addition, the Not Retrieved Relevant rate was reduced by around 8 percent, where their method only missed retrieving 17 indexes on average, which is considered – as far as the researchers' knowledge - 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. This led to an increased number of the relevant candidate terms. Hence, a wider variety of correlated words is extracted allowing the user to determine the text's topic properly and efficiently. To visualize and properly present the improvement the proposed method provided, the researchers displayed the results outcome in figure 7 by calculating the average results of all the studied texts.

	Daher's Work [12]	Nasrallah's Work [13]	Our proposed Solution	Improvement
Precision	67%	69%	81%	12%
Recall	70%	76%	83%	7%
Not Retrieved Relevant (# Words)	31	26	17	-9
F-Measure	0.68	0.71	0.82	15%

Fig 7. Comparison of our results using the Thesaurus Integration Method with the previous methods

CONCLUSION AND FUTURE WORK

The automatic indexing topic has acquired a lot of focus in the past couple of decades 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. Due to the complexity of this language, a very robust method of auto-indexing is needed to tackle several topics. In the presented work, the researchers presented a solution for the "Automatic Indexing of Arabic Texts" problem. They 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.

As for future work, the researchers plan on augmenting their solution with diacritization analysis. In the Arabic language there are certain symbols called "7arakat" that are inserted to the word to dictate its pronunciation. Diacritization presents many benefits since the same word having the same number of characters can have several different meanings 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 our solution to be able to identify the different forms of a diactritized word will improve accuracy and content analysis.

ACKNOWLEDGEMENTS

This work was supported by the Lebanese American University – Beirut, Lebanon.

REFERENCES

[1] "Number of internet and social media users worldwide as of January 2023," statistica.com, https://www.statista.com/statistics/617136/digital-population-worldwide (accessed March 12, 2023).

[2] M. K. Bergman, "White paper: the deep web: surfacing hidden value," *Journal of electronic publishing*, vol. 7, no. 1, 2001.

[3] C. Schneider, "The biggest data challenges that you might not even know you have," *IBM Blog AI for the Enterprise*, 2016.

[4] N. Mansour, R. A. Haraty, W. Daher, and M. Houri, "An auto-indexing method for Arabic text," *Inf Process Manag*, vol. 44, no. 4, 2008, doi: 10.1016/j.ipm.2007.12.007.

[5] M. H. Ibrahim and A. G. Chejne, "The Arabic Language: Its Role in History," *Language (Baltim)*, vol. 48, no. 3, 1972, doi: 10.2307/412051.

[6] R. A. Nicholson, *A literary history of the Arabs*. 2013. doi: 10.4324/9780203038956.

[7] *The Qu'ran*, Surat 12, Verse 2, New York, USA: Oxford University Press, 2015.

[8] A. S. Khatib, "Terminological specifications and applications in the Arabic language," *In Proc. Cultural Fifteenth Season of the Arabic Language - Academy of Jordan*, Amman, Jordan, pp. 177–213, 1997.

[9] "United Nations – Official Languages," UN.org, https://www.un.org/en/our-work/official-languages (accessed March 12, 2023).

[10] A. Issa and A. Siddeik, "Arabic language and computational linguistics," *International Journal on Studies in English Language and Literature*, vol. 6, no. 11, pp. 4-13, November 2018.

[11] R. A. Haraty, N. Mansour, and W. Daher, "An Arabic auto-indexing system for information retrieval," in *IASTED International Multi-Conference on Applied Informatics*, 2003, vol. 21.

[12] R. A. Haraty and R. Nasrallah, "Indexing Arabic texts using association rule data mining," *Library Hi Tech*, vol. 37, no. 1, 2019, doi: 10.1108/LHT-07-2017-0147.

[13] C. L. Borgman, "Multi-media, multi-cultural, and multi-lingual digital libraries: Or how do we exchange data in 400 languages?," *D-Lib Magazine*, vol. 3, no. 6. 1997.

http://dx.doi.org/10.21622/ACE.2023.03.1.001

[14] A. J. Warner, "Natural language processing," *Annual review of information science and technology*, vol. 22, pp. 79–108, 1987.

[15] C. Fellbaum, "WordNet: An electronic lexical database. 1998," *Br J Hosp Med (Lond)*, vol. 71, no. 3, 1998.

[16] R. Mihalcea and D. I. Moldovan, "AutoASC – A system for automatic acquisition of sense tagged corpora," *International Journal of Pattern Recognition and Artificial Intelligence, vol. 14, no. 1, pp. 3-17, 2000*.

[17] W. Black *et al.*, "Introducing the Arabic WordNet project," in *GWC 2006: 3rd International Global WordNet Conference, Proceedings*, 2005.

[18] S. Feldman, "NLP meets the jabberwocky natural language processing in information retrieval," *Online (Wilton, Connecticut)*, vol. 23, no. 3, 1999.

[19] J. Devlin, M.-W. Chang, K. Lee, K. T. Google, and A. I. Language, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *Naacl-Hlt 2019*, no. Mlm, 2018.

[20] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding," in *EMNLP 2018 - 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, Proceedings of the 1st Workshop*, 2018. doi: 10.18653/v1/w18-5446.

[21] R. A. Haraty, M. M. Allaham, and A. El-Homaissi, "Towards diactritizing Arabic text," in *26th International Conference on Computer Applications in Industry and Engineering, CAINE 2013*, 2013.

[22] R. Alnefaie and A. M. Azmi, "Automatic minimal diacritization of Arabic texts," in *Procedia Computer Science*, 2017, vol. 117. doi: 10.1016/j.procs.2017.10.106.

[23] M. V. Koroteev, "BERT: a review of applications in natural language processing and understanding," 2021, arXiv:2103.11943.

[24] R. A. Haraty and C. Ghaddar, "Arabic Text Recognition," *International Arab Journal of Information Technology*, vol, 1, no. 2, pp. 156-163, July 2004.

[25] R. A. Haraty and S. A. Khatib, "T-Stem - A Superior Stemmer and Temporal Extractor for Arabic Texts," *Journal of Digital Information Management*, vol. 3, no. 3, pp. 173-180, September 2005.

[26] R. A. Haraty and R. Varjabedian, "ADD: Arabic duplicate detector - a duplicate detection data cleansing tool," 2004. doi: 10.1109/aiccsa.2003.1227569.

[27] J. Xu, A. Fraser, and R. Weischedel, "Empirical studies in strategies for Arabic retrieval," in *SIGIR Forum (ACM Special Interest Group on Information Retrieval)*, 2002. doi: 10.1145/564376.564424.

[28] A. McCallum and K. Nigam, "A Comparison of Event Models for Naive Bayes Text Classification," *AAAI/ICML-98 Workshop on Learning for Text Categorization*, 1998, doi: 10.1.1.46.1529.

[29] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz, "A Bayesian approach to filtering junk e-mail," *Learning for Text Categorization: Papers from the AAAI Workshop*, vol. WS-98-05, no. Cohen, 1998.

[30] T. Joachims, "Text categorization with Support Vector Machines: Learning with many relevant features," 1998. doi: 10.1007/bfb0026683.

[31] S. Al-Harbi, A. Almuhareb, A. Al-Thubaity, M. S. Khorsheed, and A. Al-Rajeh, "Automatic Arabic Text Classification," *Text*, no. August, 2008.

[32] S. Khoja, "APT : Arabic Part-Of-speech Tagger," *Proceedings of the Student Workshop at NAACL*, 2001.

http://dx.doi.org/10.21622/ACE.2023.03.1.001

[33] National Information Standards Organization, "ANSI/NISO Z39.19-2005: Guidelines for the Construction , Format , and Management of Monolingual Controlled Vocabularies," 2005.

[34] W. R. Hersh, D. H. Hickam, and T. J. Leone, "Words, concepts, or both: optimal indexing units for automated information retrieval.," *Proceedings / the ... Annual Symposium on Computer Application [sic] in Medical Care. Symposium on Computer Applications in Medical Care*, 1992.

[35] O. Medelyan and I. H. Witten, "Thesaurus based automatic keyphrase indexing," in *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries*, 2006, vol. 2006. doi: 10.1145/1141753.1141819.

[36] C. M. Rahman, F. A. Sohel, P. Naushad, and S. Kamruzzaman, "Text classification using the concept of association rule of data mining," *in Proc. International Conference on Information Technology*, Kathmandu, Nepal, May 2003, pp. 234-241.

[37] B. Sharef, N. Omar, and Z. Sharef, "An automated Arabic Text Categorization based on the Frequency Ratio Accumulation," *International Arab Journal of Information Technology*, vol. 11, no. 2, 2014.

[38] M. Lassi, "Automatic thesaurus construction," University College of Boars, Tech. Rep. 2002.

[39] G. Kanaan and M. Wedyan, "Constructing an automatic thesaurus to enhance Arabic information retrieval system," *in Proc. of The 2nd Jordanian International Conference on Computer Science and Engineering, (JICCSE)*, 2006, pp. 89–97.

[40] M. Rushdi-Saleh, M. T. Martín-Valdivia, L. A. Ureña-López, and J. M. Perea-Ortega, "OCA: Opinion corpus for Arabic," *Journal of the American Society for Information Science and Technology*, vol. 62, no. 10. 2011. doi: 10.1002/asi.21598.

[41] M. A. Abderrahim, M. Dib, M. E. A. Abderrahim, and M. A. Chikh, "Semantic indexing of Arabic texts for information retrieval system," *Int J Speech Technol,* vol. 19, no. 2, 2016, doi: 10.1007/s10772-015-9307-3.

[42] G. S. Kaseb and M. F. Ahmed, "Arabic Sentiment Analysis approaches: An analytical survey," *Int J Sci Eng Res*, vol. 7, no. 10, 2016.

[43] A. El-halees, "Arabic opinion mining using combined classification approach," Proceeding The International Arab Conference On Information Technology, Azrqa, Jordan., 2011.

[44] "42Saudi twitter corpus for sentiment analysis.pdf."

[45] H. Al-Rubaiee, R. Qiu, and D. Li, "Identifying Mubasher software products through sentiment analysis of Arabic tweets," in 2016 International Conference on Industrial Informatics and Computer Systems, CIICS 2016, 2016. doi: 10.1109/ICCSII.2016.7462396.

[46] R. M. Duwairi, M. Alfaqeh, M. Wardat, and A. Alrabadi, "Sentiment analysis for Arabizi text," in 2016 7th International Conference on Information and Communication Systems, ICICS 2016, 2016. doi: 10.1109/IACS.2016.7476098.

[47] P. Gillman, Text Retrieval: The State of the Art, London, UK: Taylor Graham Publishing, 1990.

[48] E. Deeb, New Arabic Grammar Rules, Beirut, Lebanon: Lebanese Book Publishing, 1970.

[49] A. Kindery, F. Rajihy, and F. Shimry, Arabic Grammar Book, Kuwait City, Kuwait: Rissala Publishing, 1996.

[50] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules," in Proc. of 20th International Conference on Very Large Data Bases, {VLDB'94}, 1994.