Developing A New Term Weighting Schema Through Text-Document Analysis and Natural Language Processing NLP

Manal Sheikh Oghli
Web Science program, Syrian Virtual University
Damascus, Syria

Muhammad Mazen Almustafa
Web Science program, Syrian Virtual University
Damascus, Syria

***Abstract*—Term weighting is one of the branches concerned with information retrieval IR. It studies the importance of “word” or “phrase” in a certain text, as the issue of determining the importance of keywords is essential and effective in the modern retrieval systems.**

Studies showed that the appropriate weight of the term importance affects the retrieval results. Thus, the distribution, location, indication, and synchronization of the term with other terms in the document are factors that should be taken into consideration upon measuring the similarity between documents or between query and document.

The paper sheds light on some weak points in the traditional term weighting (TF-IDF) of the vector space model. It also reviews some algorithms to improve TF-IDF performance, then develop a new mechanism for term weighting depending on text analysis and natural language processing through features of the terms deduced from the information derived from text analysis and processing, and conducting the appropriate mathematical tests to reach a new way for term weighting. This new way enhances the ability of the suggested system to retrieve the most appropriate information requested by the user; which is the most essential goal all retrieval systems are seeking to achieve.

***Keywords*— Information Retrieval IR, Vector Space Model VSM, Indexing, Term Weighting, TF-IDF, Natural Language Processing NLP)**

I. INTRODUCTION

The effective research systems do not work directly with documents or queries as different techniques and strategies are used to represent the essential meaning in the form of parts of a document or inquiries; a process that is called indexing [1].

Terms in the vector space model are represented as a vector getting out of a set of concepts, where the vector represents the keywords and terms extracted from documents. The biggest challenge facing this model; however, is determining the suitable value of the vector constituents or what is known as Term Weighting, in addition to terms independence [2] [3].

According to the VSM model, the long document that could contain the same terms appearing in the query - only in the title and abstract - can be of a great relevance to query, but in this model, it will have less significance in comparison with a short document having the same terms in the footing. A flaw in the VSM document representation appears in that the term arrangement is missing. Documents that do not have close query term cannot be preferred to documents having separate terms in various parts of the document [3].

Processes in the vector space model are three stages: The first stage is “Indexing”, where terms are deducted from the text. The second stage is “Term Weighting”, and the third is “Classification” as regards query and similarity [4].

This paper suggests a new mechanism for term weighting with the aim of overcoming the shortcomings of the vector space model through identifying the term features of document terms where features give quantitative or qualitative indicators that determine the value of information, significance to document, and the relation between terms and their occurrence within the document and their grammatical position that increases the effectiveness of this model.

II. INDEXING IN INFORMATION RETRIEVAL SYSTEMS

This process implies determining the keywords that represent a document on the basis of its contents. It is a significant stage in the retrieval system. Indexing is defined as “a process that determines keywords or descriptive terms, what is called “Index Terms” that represent the document on basis of its contents to reach an effective access of documents.” [5].

Text processing and analysis is the first step of indexing in retrieval systems to get a more effective retrieval. To achieve this, there should be an appropriate structure for indexing. The most used data structure is the Inverted index which is a term-oriented mechanism which is the most competent and flexible index structures [6].

The structure of the Inverted Index has two components; vocabulary and document list. Vocabulary is a set of various terms concluded from documents. Each document is represented by a list of some referential words stored alphabetically [6] [7]. It should be noted that some statistical information could be stored about each term in each document, like term frequency, term position, and other useful features for the retrieval process.

To establish the “Index”, text should undergo analysis and processing within information retrieval systems; such as:

A. *Linguistic Analysis (Tokens Extraction):*

The most important question that should be raised at this stage is the following: What are the right tokens that should the system process and store? [8].

At this stage, a text is analyzed, distinctive terms chosen, and unnecessary symbols and punctuation marks removed. This process is usually referred to as “Tokenization”, where the document is divided into units called “Tokens”. This results in a set of words of semantic significance [9].

Here, you have to differentiate between “Token” and “Term”. “Token” means a distinctive symbol. It is a representation of a series of letters in a certain document combined to form a good-to-process semantic unit, whereas "Term" is a token processed to be inserted in the IR Index [6] [8].

B. Stemming & Lemmatization:

This process is also called in general "Abstraction". Usually, the terms of abstraction, stemming and lemmatization refer to the change of the word structure and reduction of term form to a common form.

"Stemming" refers to extracting part of the end of the word and removing any suffixes; i.e., removing any additions to the word, whereas "Lemmatization" depends on the morphological analysis of words to remove the inflectional suffixes only and restore the basic form of the word, as stated in the "Linguistic Dictionary", which is known as "root", or "Lemma" [8] [9].

The aim of abstraction or stemming is reducing the different forms of the word generated due to inflection, and sometimes the derivative forms that are related to the word to a common form [6] [8]. Hence, when a user specifies a term to search for, it is necessary to retrieve all documents that have grammatical variables of the term, which prevents any exact match between the query term and the document containing this term.

Through this process, all grammatical forms of terms are represented in a basic common form. This also helps reduce the size of documents and makes search faster through searching for the abstract term instead of searching for the whole term.

The first stemmer of the English language was developed in 1968 by Julie Beth Lovins who introduced the concept of "abstraction" based on the "Dictionary of Common Suffixes". This logarithm was based on the principle of removing suffixing with view to the longest match. This logarithm led to reasonable results in the field of information retrieval [10].

Then came Martin Porter who published a paper in 1980 in the Programme Journal to describe a very simple logarithm, in concept. This logarithm is controlled by certain rules that determine whether the suffix could be omitted or not depending on the minimum left after omission. The logarithm; however, repeatedly proved to be empirically effective [8]. Porter did not depend on an abstraction dictionary; rather, he used lists of suffixes, then linked each suffix with a special criterion to delete the suffix from the word to get a true abstract work when applying the criterion. The logarithm consists of several stages; each of which contains a number of rules to remove suffixes and it is available in different languages except Arabic [6].

The Paice/Husk stemmer was published for the first time in 1990. It was developed by Chris Paice with the help of Gareth Husk, who used a table of indexed rules which determine whether suffixes would be omitted or replaced [10].

C. Removing Stop Words

Stop words is a list of linguistically common words and have a limited effect on the categorization and selection of documents that goes appropriately with user needs.

They are functional words that have no meaning, and are part of how nouns in a text are described and expressed [6]. Like pronouns, connectors and prepositions, that appear in all text documents [9].

Rarely do these words refer to anything related to the subject of the document; hence, such functional words will not be of help in the search processes [6].

The general strategy used to define this list of words depends on sorting terms by aggregation frequency (total number of the times each term appears in a certain set of documents). Then, the most common terms are taken, and filtered, often manually, due to their poor semantic content in relation to the documents being indexed [6].

D. Term Weighting:

Term Weighting is defined as a digital computing aiming to express the importance of a word within a group. It is usually used as a weighing factor in the search processes within information retrieval systems [11].

It is a calculation process and determining a digital value for each term to consider its contribution in distinguishing a certain document. Terms are descriptors of content in the documents used in indexing, and through which the relatedness of documents to queries is evaluated. These terms are classified as objective and non-objective, where the weighting process is applied to the non-objective terms which reflect the contents of a document. Then, these terms are weighted and their significance in relation to the information included in the document is demonstrated [12].

III. TF-IDF ALGORITHM

TF-IDF is an abbreviation of Term Frequency–Inverse Document Frequency, which is a technique commonly used in text mining and information retrieval. The Term Frequency, which was one of the most important developments in the field of information retrieval, was defined, for the first time, by Scientist Gerard Salton in the nineteen seventies [13].

This was supplemented by the work accomplished by Spark Jones who presented her paper about the Inverse Document Frequency (IDF) [14], and resulted in the quick adoption of the two-method combination; i.e., (TF) and (IDF), as two new methods for term weighting [3] [15].

The term frequency (TF) is also called "Local Term Weigh" and is defined as the number of recurrences of the term being searched within a document.

$$TF(t) = \frac{\text{Number of times term t appears in a document}}{\text{Total number of terms in the document}}$$

The IDF, on the other hand, is called "Global Term Weight" and reflects the term frequency within a number of documents.

$$IDF(t) = \log_e \frac{\text{Total number of documents}}{\text{Number of documents with term t in it}}$$

The term weighting could be calculated by use of TF and IDF through the following equation:

$$\boldsymbol{TF - IDF_{t,d} = TF_{t,d} * IDF_t}$$

IV. LIMITATIONS OF TF-IDF:

Despite the numerous features of the traditional way TF-IDF, there are a lot of shortcomings which cannot be ignored. Those shortcomings are like the following:

1) The traditional method assumes that calculating the term frequency gives an independent proof of similarity, which is not always correct [16].
2) This method calculates term weighting on basis of the term frequency, and is not concerned with the term position in the text [17] [18].
3) The traditional TF-IDF is a technique used to select an unsupervised feature as it is only limited only to the document [16]. Nonetheless, it does not discuss the

common connotation or occurrence with other terms in the document. It is not, either, concerned with the relation between words and the importance of the term in relation to the text itself [17] [18].

Therefore, this domain was and is still a motive for a lot of researchers to study the possibility of producing different forms of weighting plans and improve TF-IDF algorithm with the aim of developing information retrieval systems.

Here are three examples:

- In 2017, a number of scholars; namely, Thabit Sabbah, Ali Selamat, Md Hafiz Selamat, Fawaz S.AlAnzi, Enrique Herrera Viedma, Ondrej Krejcar, Hamido Fujita, proposed four schemes for term weighting depending on the TF-IDF scheme. They were: mTF, mTFIDF, TFmIDF, and mTFmIDF. The schemes take into consideration the missing term counting with calculating the weights of current terms to improve the performance of Text Classification (TC).

  The first proposal examines the number of missing terms in a text in comparison to the total number of terms in a certain group to present a new weighting called "mTF". The second proposal; however, examines the percentage of the number of missing term documents to the number of texts in a group, which researchers called "mIDF".

  Based on the previous two proposals, scholars presented different standard weighting schemes of the TF-IDF, on the basis of the proposed mIDF and mTF schemes [19].

  In this study, scholars depended on the idea that some terms should disappear when other terms already exist in the text, and vice versa. Therefore, the weights of terms would, definitely, be affected.
- In 2018, Rajendra Kumar Roul, Jajati Keshari Sahoo, and Kushagr Arora from Zuarinagar University in India, conducted a study that demonstrated the shortcomings of the TF-IDF, then proposed four different techniques for term weighting with view to overcoming those shortcomings by modifying the traditional TF-IDF. The study showed that the text representation based on language, like the BSM Vector Space Model, greatly affect, especially in the fields where language is processed naturally (NLP), the information retrieval (IR), and that the transfer of texts to vector renders the possibility of conducting any mathematical operation on vector-represented texts. Therefore, term weighting plays a big role in representing documents more accurately.

  The four techniques proposed by the researchers in this paper entail a modification of the known TF-IDF algorithm by the addition of some mathematical operations to documents such as Inter-class dispersion, where the term is distributed in a unified way among the different classes, which means term dispersion will be low., therefore, the weight the term will contribute in will also be low. If; however, the term has a big disparity, this means it is good and the weight it is contributing in will be high. The second proposal was about modification of the traditional IDF through giving the value 0 or 1 depending on the frequency of the term in all documents or non-occurrence in any document. The third proposal took into consideration the number of documents containing that term and belong to a definite category, and the total number of documents in this category. The last proposal examined the importance of the document length in the term weighting [16].
- In 2019, Shuzhi Sam Ge and Ting ZHANG from the University of Electronic Sciences and Technology in China suggested a new term weighting way. They called it "TF-IDF-ρ"

  Their study showed that the traditional algorithm TF-IDF is one of the term weighting algorithms and the most common text representation way [18]. Nevertheless, it does not function properly, and has a lot of shortcomings.

  Throughout the study, they worked on improving this algorithm by suggesting a new idea that depends on the "Class Discriminative Strength", which the term affects. They suggesting benefiting from this to improve the traditional TF-IDF algorithm.

  The study also showed that the selection and weight calculation of the distinctive terms of texts defines to a great extent whether the text has been properly classified or not.

  Researchers suggested that the class discriminative strength ρ represents the discriminatory power of a feature item, equal to a total number of the category in corpus divided by the number of classes of feature item occurrence in.

  This way suggested by researchers represents in assigning a greater weight for distinctive terms which appear in greater proportions as a strength that distinguishes classification or category, with the aim of shedding light on the ability of this term to distinguish different texts [18].

V. NATURAL LANGUAGE PROCESSING (NLP):

The need to analyze texts before retrieval is one of the biggest obstacles facing Information Retrieval Systems (IRS) as the latter mainly depend on understanding the content of texts to be retrieved, and analyzing the words used to build queries, then making a link between keywords and the text database, and conducting the appropriate weighting process to reach the proper text. This led to the arising need of a textual analysis process.

Natural Language Processing (NLP) is a way to analyze texts in a computer. It includes collecting knowledge about how humans understand and use language, to develop the appropriate tools and techniques that make computer systems capable of understanding and processing natural languages to perform the various required tasks [20].

Dr. Michael J.Garbade defines Natural Language Processing (NLP) as a branch of Artificial Intelligence (AI) that handles the interaction between computers and humans through the use of natural language. It shows that the ultimate goal of Natural Language Processing is reading, understanding and realizing the human languages in a valuable way and inferring the required meaning from them [21].

VI. STANFORD CORENLP

The Stanford CoreNLP is one of the widely used tools as most of the common essential natural language processing are available, such as tokenization and even coreference resolution through combining several constituents of natural language analysis [22].

One of the initial release goals that was developed in 2006 was obtaining Annotators Pipeline swiftly, and providing a light framework through the use of Java objects and applying them on any text instead of applying them on a single sentence only. In 2009, the system was developed to be used more easily and by a wider range of users, as the system provided the interface of command line and the ability to write outside annotations with different formats, including XML.

Control of annotations could be through the Object Properties, and the Stanford CoreNLP set could be packaged in a way that makes them easy to access by various languages, such as: Python, Ruby, Perl, Scala, JavaScript, Net and even C# [22].

The current release entails a set of processing tools designed to take initial textual inputs, giving whole textual analysis outputs, and linguistic annotations appropriate for the effective textual analysis [23].

The most important tasks for natural language processing which the CoreNLP set of tools carries out are: Tokenization, Lemma, Named Entity Recognition, where names are recognized as one of the shapes (person, location, organization..), numbers (set، duration ،time ،date ،number ،money), and classification of symbols according to the Part of Speech they belong to and have been symbolized through a set of Tags [23].

In spite of some notes on the analytical tasks of these tools, it could be said that it is a set of easy-to-understand tools that could be used as a constituent within a bigger and scalable system [22].

Here is a review of the system structure through which Stanford CoreNLP analyzes and process texts:

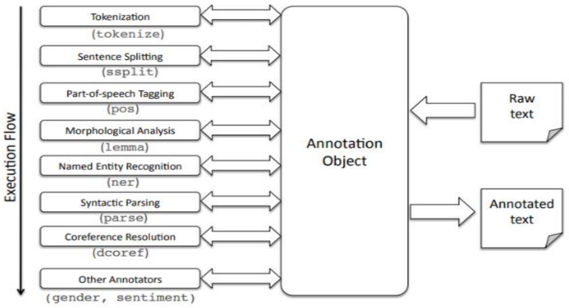


Figure 1: System structure of Stanford CoreNLP [22]

VII. A PROPOSED METHODOLOGY FOR TEXT PROCESSING AND INDEXING:

The CoreNLP set of tools was benefited from in the text analysis and extracting features of each document and the most significant terms through a specific text analysis methodology that used the following:

1) Analyzing the text through the Stanford CoreNLP set of tools and dividing it into a group of tokens.
2) Classifying tokens through POS analysis into groups correlated to the grammatical position of token, then sorting the tokens into four main groups: (Functional tokens, verbs, nouns, and adjectives)
3) Omission of functional tokens as they perform a functional task in the text and do not play a role in defining the subject of the document.
4) Applying Lemma on (verbs, adjectives and nouns).
5) Automatic filtering of some non-functional tokens. Some tokens, considered Stop Words, have been defined in a proposed system. Those are some verbs, nouns, and adjectives commonly used but do not participate in defining the subject of the document. It should be noted that the number of tokens in this list are about 200, such as: (do, like, good, great ...).
6) Indexing the tokens resulting from previous processes like terms within a proposed database system, and some features like (POS, NER, and Order).

VIII. PROPOSED METHODOLOGY IN TERM WEIGHTING:

The common method used in the fields of natural language processing (NLP) is:

1) Looking for features in the document.
2) Defining the significance of these features.
3) Sending the weighted features for the sake of taking the right decision [24].

As long as the term frequency is not considered the only dimension that information retrieval systems (IRS) rely on in determining the relatedness of queries and documents. Therefore, we suggest the following two parameters:

A. *Addition of the POS Parameter:*

This coefficient defines the lexical matching between the term in the query and the term in the document. For example: The term (book) has two different meanings in the following statements:

- Book a study seat in the Syrian Virtual University to develop your scientific level.

- The Syrian Virtual University website has many important digital books.

The term (book) will take a unified shape after the abstraction process in both texts. Therefore, it is necessary to distinguish between both terms, when it appears as a verb and as a noun, and determine how much it conforms to the user desire.

Therefore, the study suggests classify the POS results into a classes (nouns, verbs, and adjectives) which are identical to their classification within the indexing process.

Then, values are given to terms common to query and text according to the following table:

POS=POS
NER =NER
	POS=POS
	One NER
		POS=POS
		Not NER
			POS ≠ POS
			Same Class
				POS ≠ POS
				Not Same Class

1	0.8	0.5	0.3	0.1

Thus, the proposed equation for the POS calculation of the text will be like this:

$$POS = \frac{\sum_{t=0}^{n} \textbf{POS Value}}{\textbf{Total Count Of Term } (N)}$$

B. Addition of the Correlation Parameter:

This parameter examines how much words are correlated through studying the position of words and the calculating distances between them in the text. The correlation parameter could be calculated through the following equation:

$$\boldsymbol{Corr} = \frac{\boldsymbol{Common\ Terms\ (n)^2}}{[\sum_{k=0}^{n-1} \boldsymbol{dis(term)+1}] * \boldsymbol{Count\ Term\ Of\ Query}}$$

Where (n) Common Terms is the number of common terms between query and text, and (dis) is the distance between term (i) and term (i+1); i.e.:

$$Dis = Order\ Term_{i+1} - Order\ Term_i$$

as long as terms will be distributed within the text with a possibility of frequency, then calculating the less value of distance between terms, and calculating correlation depending on these values.

Thus, we notice that the position of terms gives an added value through which we could make a differentiation process between documents based on their value.

C. Algorithm of the Proposed Term Weighting NLP-TF-IDF:

The proposed way is a supportive way of the traditional term weighting algorithm, through some features indexed by the use of CoreNLP tools, therefore, the weighting will be calculated through the following equation:

$$\boldsymbol{NLP\ Similarity} = \boldsymbol{Cosine\ Sim_{TF-IDF}} * \frac{[\,\boldsymbol{1+(10*POS)+Corr}]}{\boldsymbol{2}}$$

IX. Evaluation In the Information Retrieval Systems:

The effectiveness in information retrieval systems is a measure of retrieved documents by the system to meet the needs of users. The process of identifying the effectiveness of retrieval of a certain inquiry is referred to as “Effectiveness Evaluation” [12].

The measures of effectiveness of retrieval systems are precision (the percentage of relevant documents as for the retrieved group), and recall (the percentage of relevant documents in the retrieved group as for all documents) [12] [19].

Some evaluation systems used the F1 measure as a measure combining precision and recall [3] [19].

Here is how the F1, precision and recall are calculated:

$$Recall = \frac{\mathrm{TP}\ (Retrieved\ \&\ Relevant)}{\mathrm{TP}\ +\mathrm{FN}\ (Not\ Retrieved\ \&\ Relevant)}$$

$$Precision = \frac{\mathrm{TP}\ (Retrieved\ \&\ Relevant)}{\mathrm{TP}+\mathrm{FP}\ (Retrieved\ \&\ Non\_Relevant)}$$

$$F1\ = 2\ \times \frac{Precision \times Recall}{Precision + Recall} = \frac{2*\mathrm{TP}}{2*\mathrm{TP}+\mathrm{FN}+\mathrm{FP}}$$

X. Experimental Results:

The CISI was chosen to be a normative data set. It is a number of scientific articles; 1460 articles published between 1969 and 1977. They included the author's name, title of article and abstract. The group was provided with a query group and expert results for each query [25].

A partial set of this data was used. This set had 300 texts to be tested. The first 30 queries were chosen to be tested. As long as the selected texts to be a pilot group were 300 texts, we found out that 3 queries had zero results within the selected documents. Therefore, the results were not shown within the test results. Our results were compared to the traditional TF-IDF algorithm and the Porter Index.

A. First Experimental Stage:

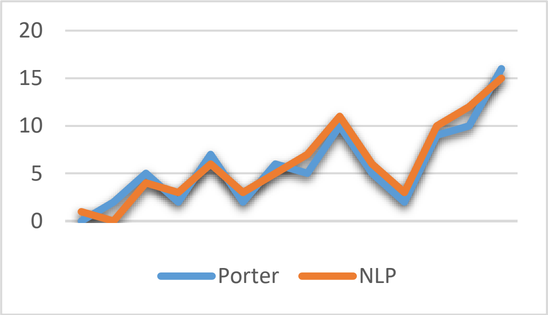
At this stage, the order of retrieved documents is tested by the use of Cosine similarity of the traditional Term Weighting algorithm according to the proposed system methodology by use of the Porter Stemmer, where a definite number of documents is retrieved; i.e., fixing the value of TP + FP = 25. The traditional algorithm of Term Weighting TF-IDF is not modified at this stage, but it is used and the similarity is calculated by the use of the Cosine Similarity.

Here is a comparison of the evaluation factors of the three information retrieval systems (Recall, Precision and F1):

Query	**Porter Index**			**NLP Index**			**F1**
	Recall	**Prec**	**F1**	**Recall**	**Prec**	**F1**	**$\mathbf{F1_{NLP}}$ - $\mathbf{F1_{Porter}}$**
1	0.5882	0.4	0.4761	0.6470	0.44	0.5238	0.0476
2	0.4	0.08	0.1333	0	0	0	-0.1333
3	0.7	0.28	0.4	0.6	0.24	0.3428	-0.0571
5	0.2	0.08	0.1142	0.3	0.12	0.1714	0.0571
8	0	0	0	0	0	0	0
9	0.6	0.12	0.2	0.6	0.12	0.2	0
10	0.8333	0.2	0.3225	0.6666	0.16	0.2580	-0.0645
11	0.1935	0.24	0.2142	0.1935	0.24	0.2142	0
12	0	0	0	0.3333	0.04	0.0714	0.0714
13	0.5	0.64	0.5614	0.4687	0.6	0.5263	-0.0351
14	1	0.04	0.0769	1	0.04	0.0769	0
15	0.4090	0.36	0.3829	0.4090	0.36	0.3829	0
16	0.5	0.08	0.1379	0.5	0.08	0.1379	0
17	0.25	0.04	0.0689	0.25	0.04	0.0689	0
18	0.6666	0.08	0.1428	0.6666	0.08	0.1428	0
19	0.5	0.24	0.3243	0.5	0.24	0.3243	0
20	0.3125	0.2	0.2439	0.4375	0.28	0.3414	0.0975
21	0.2857	0.08	0.125	0.2857	0.08	0.125	0
22	0.0909	0.08	0.0851	0.1363	0.12	0.1276	0.0425
23	0.2608	0.24	0.25	0.2608	0.24	0.25	0
24	0.4615	0.24	0.3157	0.3846	0.2	0.2631	-0.0526
25	0.2857	0.08	0.125	0.4285	0.12	0.1875	0.0625
26	0.5833	0.28	0.3783	0.5833	0.28	0.3783	0
27	0.3225	0.4	0.3571	0.3871	0.48	0.4285	0.0714
28	0.5	0.16	0.2424	0.5	0.16	0.2424	0
29	0.2777	0.2	0.2325	0.3333	0.24	0.2790	0.0465
30	0.3913	0.36	0.375	0.4347	0.4	0.4166	0.0416

Table-1: Comparison of Evaluation Values between NLP and Porter Index

Here is a chart demonstrating the difference of the TP values between NLP, Porter Index:



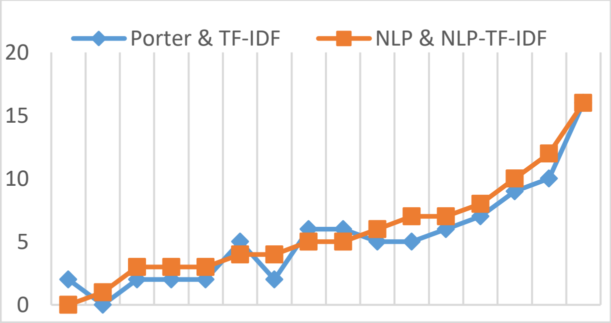
The results of this stage showed the following:

1) The values of the three evaluation parameters (Recall ، Precision ،F1) with the values of (0.72, 0.74, 0.75) increased by the use of the traditional weighting measure of the indexed documents in the proposed indexing system by use of Porter Stemmer.
2) The new documents by use of NLP Index were 25 different from the documents that appeared in the Porter Index, which indicates the indexing significance in the information retrieval systems and their huge impact on results.
3) Although the Porter Indexing outperformed 5 queries, the total results of the three-evaluation parameter indicated the high results achieved by the proposed indexing system. Thus, the indexing with the use of the NLP tool set outperformed 9 queries and the results were similar in 13 other queries.

B. Second Experimental Stage:

At this stage, the mechanism of calculating similarity between query and documents was modified, then the results of the traditional TF-IDF and the modified NLP-TF-IDF was compared. The following results were concluded:

Here is a chart of the differences of the TP values between the traditional method (Porter Index & TF-IDF) and the proposed mechanism (NLP index & NLP-TF-IDF):



Here are the values of the three-evaluation parameters:

Query	Porter Index & TF-IDF			NLP Index & NLP-TF-IDF		
	Recall	Prec	F1	Recall	Prec	F1
1	0.5882	0.4	0.4761	1	0.04	0.0769
2	0.4	0.08	0.1333	0	0	0
3	0.7	0.28	0.4	0.3333	0.04	0.0714
5	0.2	0.08	0.1142	0.6666	0.08	0.1428
8	0	0	0	0.25	0.04	0.0689
9	0.6	0.12	0.2	0.75	0.12	0.2068
10	0.8333	0.2	0.3225	0	0	0
11	0.1935	0.24	0.2142	0.6	0.12	0.2
12	0	0	0	0.6666	0.16	0.2580
13	0.5	0.64	0.5614	0.2857	0.08	0.125
14	1	0.04	0.0769	0.4285	0.12	0.1875
15	0.4090	0.36	0.3829	0.5	0.16	0.2424
16	0.5	0.08	0.1379	0.3	0.12	0.1714
17	0.25	0.04	0.0689	0.8	0.32	0.4571
18	0.6666	0.08	0.1428	0.5	0.24	0.3243
19	0.5	0.24	0.3243	0.5833	0.28	0.3783
20	0.3125	0.2	0.2439	0.3846	0.2	0.2631
21	0.2857	0.08	0.125	0.4375	0.28	0.3414
22	0.0909	0.08	0.0851	0.5882	0.4	0.4761
23	0.2608	0.24	0.25	0.3333	0.24	0.2790
24	0.4615	0.24	0.3157	0.1818	0.16	0.1702
25	0.2857	0.08	0.125	0.4090	0.36	0.3829
26	0.5833	0.28	0.3783	0.2173	0.2	0.2083
27	0.3225	0.4	0.3571	0.4347	0.4	0.4166
28	0.5	0.16	0.2424	0.2258	0.28	0.25
29	0.2777	0.2	0.2325	0.3870	0.48	0.4285
30	0.3913	0.36	0.375	0.5	0.64	0.5614

Table-2: Comparison of Evaluation Values Between Suggested System and Porter indexing & TF-IDF

The results of this stage experiments showed the following:

1) The number of queries in which the proposed system outperformed the Porter increased to be 11 queries, and the results were similar in 12 queries. However,
2) The number of queries in which the Porter Indexing excelled by applying the proposed weighting mechanism decreased to be 4 queries.
3) As a result, the number of true documents that used the proposed weighting method rose to be /139/ documents, whereas it was only /130/ documents with the use of Porter Indexing.
4) The new /27/ documents that appeared were different from those that appeared using the Porter Indexing with the traditional Term Weighting algorithm.
5) The test results of the second stage showed that the proposed mechanism of term weighting raised the medium value of the F1 parameter to be (0.7%) higher than the use of the traditional term weighting algorithm TF-IDF of indexed data using the NLP tool set. The value of F1 as a medium value raised at (1.4%) compared to similarity values of the TF-IDF of the indexed data using the Porter Stemmer. The maximum value of the increase of the F1 parameter value reached (9.7%).

XI. CONCLUSION:

Text analysis and natural language processing is a way to understand the user desire, therefore, the right text analysis leads to more accurate retrieval results that meet user needs.

The NLP tool set could achieve great results in this field; hence, those tools could be benefited from in raising the efficacy of the information retrieval systems through developing new mechanisms for term weighting that depend, in its content, on the distinctive features that could be elicited from the analysis of such tools to texts.

The test results also affirmed the importance of indexing and its effect on the retrieval results, on the one hand, and the possibility of benefiting from the features of the NLP tools in developing the traditional term weighting mechanism TF-IDF through adding parameters that contribute in defining the relevant documents to the user's desire, on the other hand.

It should be noted that despite the shortcomings of these tools in processing the suffixes of nouns and adjectives, the results demonstrated the superiority of the proposed system method as aesthetical values to the traditional method in all parameters of information retrieval parameters evaluation. Therefore, a restructuring of indexing depending on NLP tool set and studying the suffix processing issue will lead to more effective and more accurate results.

REFERENCES

[1] J. Savoy and E. Gaussier, "Information Retrieval," in *Handbook of Natural Language Processing*, Chapman and Hall/CRC, 2010, pp. 455-484.

[2] A. Göker and J. Davies, "Information Retrieval Models," *Wiley*, pp. 1-19, 2009.

[3] M. Sheikh Oghli and M. M. Almustafa, "Comparison of basic Information Retrieval Models," *International Journal Of Enginering Research & technology (IJERT)*, vol. 10, no. 09, pp. 299-303, 2020.

[4] S. JABRI, A. DAHBI, T. GADI and A. BASSIR, "Ranking of Text Documents using TF-IDF Weighting," in *2018 4th International Conference on Optimization and Applications (ICOA)*, Mohammedia, Morocco, 2018.

[5] E. Chauhan and D. Asthana, "Review of Indexing Techniques in Information Retrieval," *International Journal of Engineering Science and Computing IJESC* , vol. 7, no. 7, pp. 13940-13942, 2017.

[6] W. B. Crof, D. Metzler and T. Strohman, Search Engines, Pearson Education, 2015.

[7] B. Saini, V. Singh and S. Kumar, "Information Retrieval Models and Searching Methodologies: Survey," *International Journal of Advance Foundation and Research in Science & Engineering (IJAFRSE)*, vol. 1, no. 2, pp. 57-62, 2014.

[8] C. D.Manning, P. Raghavan and H. Schütze, An Introduction to Information Retrieval, New York: Cambridge University Press, 2008.

[9] V. Murthy, D. B. V. Vardhan, K. Sarangam and P. V. p. Reddy, "A Comparative Study on Term Weighting Methods For Automated Telugu Text Categorization with Effective Classifiers," *International Journal of Data Mining & Knowledge Management Process (IJDKP)*, vol. 3, no. 6, pp. 95-105, 2013.

[10] V. Gurusamy, S. Kannan and K. Nandhini, "Performance Analysis: Stemming Algorithm for the English Language," *IJSRD - International Journal for Scientific Research & Development*|, vol. 5, no. 05, pp. 1933-1938, 2017.

[11] T. Xia and Y. Chai, "An Improvement to TF-IDF: Term Distribution based Term Weight Algorithm," *JOURNAL OF SOFTWARE*, vol. 6, 2011.

[12] V. Gudivada, D. L.Rao and A. R.Gudivada, "Information Retrieval: Concepts, Models, and Systems," in *Handbook of Statistics*, United States, 2018, pp. 331-401.

[13] M. Sanderson and B. Croft, "The History of Information Retrieval Research," *IEEE*, vol. 100, no. Special Centennial Issue, 2012.

[14] S. Jones, "A statistical interpretation of term specificity and its application in retrieval," *Journal of documentation*, vol. 28, no. 1, pp. 11-21, 1972.

[15] G. Salton and C. Yang, "on the Soecification of Term Value in Autoatic Indexing," *Cornell University*, pp. 73-173, 1973.

[16] R. K. Roul, J. K. Sahoo and K. Arora, "Modified TF-IDF Term Weighting Strategies for Text Categorization," in *14th IEEE India Council International Conference (INDICON)*, 2018.

[17] J. Ramos, "Using TF-IDF to Determine Word Relevance in Document Queries," January 2003.

[18] T. ZHANG and S. G. Sam , "An Improved TF-IDF Algorithm Based on Class Discriminative Strength for Text Categorization on Desensitized Data," in *ICIAI 2019: Proceedings of the 2019 3rd International Conference on Innovation in Artificial Intelligence*, Suzhou, China, 2019.

[19] T. Sabbah, A. Selamat, M. H. Selamat, F. S. Al-Anzi, E. H. Viedma and O. Krejcar, "Modified Frequency-Based Term Weighting Schemes for Text Classification," *Applied Soft Computin*, vol. 58, pp. 193-206, 2017.

[20] S. Joseph, H. Hlomani, K. Letsholo, F. Kaniwa and K. Sedimo, "Natural Language Processing: A Review," *International Journal of Research in Engineering and Applied Sciences*, vol. 6, no. 3, 2016.

[21] D. M. J.Garbade, "A Simple Introduction to Natural Language Processing," *Becoming Human: Artificial Intelligence Magazine*, 2018.

[22] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard and D. McClosky, "The Stanford CoreNLP Natural Language Processing Toolkit," in *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics*, Maryland USA, 2014.

[23] E. Nazaruka, J. Osis and V. Griberman, "Using Stanford CoreNLP Capabilities for Semantic Information Extraction from Textual Descriptions," in *Evaluation of Novel Approaches to Software Engineering*, Riga, Latvia, Riga Technical University, 2020.

[24] C. Buckley, "The importance of proper weighting methods," *Proceedings of the workshop on Human Language Technology*, pp. 349-352, 1993.

[25] H. R. Turtle, Inference networks for document retrieval, University of Massachusetts Amherst, 1991.