ISSN 2278-3091
Volume 13, No.6, November - December 2024
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse041362024.pdf
https://doi.org/10.30534/ijatcse/2024/041362024

Analyzing the Impact of Preprocessing Techniques on Arabic Document Classification: Comparative Study

Mahmoud Moshref [1], Khalid Khalis Ibrahim[2], Bassam Hammo[3,4], Derar Eleyan[5]
[1]Palestine Technical University, Kadoorie, Tulkarm, Palestine, moshref2008@gmail.com
[2]Tikrit University, Tikrit, Iraq, khalid.kh.ibrahim@tu.edu.iq
[3]Princess Sumaya University for Technology, Amman, Jordan, b.hammo@psut.edu.jo
[4]The University of Jordan, Amman, Jordan, b.hammo@ju.edu.jo
[5]Nablus University for Vocational and Technical Education, Nablus, Palestine, d.eleyan@nu-vte.edu.ps

Received Date : October 12, 2024 Accepted Date: November 19, 2024 Published Date: December 06, 2024

ABSTRACT

Texts classification is an important field that can be used in data mining, information retrieval, machine learning. Documents classification now widely used in different domains, such as mail spam filtering, article indexing, Web searching, and Web page categorization. There are many researches in documents classification for English language, but a few research in Arabic language, while there are large community in the world that uses this language. This paper analyzes the effect of preprocessing, such as tokenization, normalization and removing stop words, stemming using Khoja stemmer, stemming using light stemmer, stemming using Khoja stemmer with tokenization, normalization and removing stop words, and stemming using light stemmer with tokenization, normalization and removing stop words on documents classification. This study uses three classification algorithms, Naïve Bayes (NB), K-Nearest Neighbor (KNN), and Support Vector Machine (SVM), it is applied on online Arabic corpus prepared by Diab Abuaiadh. Rapid Miner tool is used to apply the three classification models. Whereas the condition of the documents before preprocessing is compared with their condition after preprocessing to determine the extent of the effect of preprocessing on documents classification. the results demonstrate variation between document classification before preprocessing and after preprocessing, and difference between the three algorithms in terms of Accuracy, Precision, Recall, and F1-Score, whereas it will be discussed later.

Key words: Natural Language Processing, Machine Learning, Preprocessing, Document Classification, Naïve Bayes, KNN, SVM, Tokenization, Normalization, Stop Words, Stemming**.**

1. INTRODUCTION

Document Classification is one of the most popular Natural Language Processing (NLP) tasks, it become important for put the documents on their proper category according to their content. Specially, when we take Arabic documents as a case study, it would appear that efforts to investigate Arabic documents classification and categorization are rare, and much less attention was directed towards document classification in Arabic, this related to several reason as the lack of rich representative resources for training an Arabic Document classifier [1] [2]. Whereas there is rapid growth in using Arabic documents in internet, these documents must be classified according to their subject, or to their importance to get useful information from these documents.

Text or Document Classification has been used for different applications such as: documents organization, text filtering, spam filtering, mails routing, word sense disambiguation, news monitoring, automatic other uses as sentiment analysis, detecting trends in customer feedback, Web page categorization, spam detection and topic labeling [3]. Since big data that come from internet become most important, abundant information is now available on the Web, in huge collections of text documents stored in an unstructured text format, so there is difficult for users to find the information they need. These documents need to classified into subject categories to make them easier to use on text mining tasks, or information retrieval or machine learning [4].

In general, the document classification pass through three steps: the first one is preprocessing that allows removing stop-words, tokenization, normalization, etc., the second is feature extraction that codifies the Arabic text, and the third is machine learning algorithms (classifiers) that is run on the training set to generate as output. These algorithms as Naïve Bayes (NB), K-Nearest Neighbor (KNN), and Support Vector Machine (SVM), Decision Tree, Decision Table, etc. Document preprocessing is the task that changes the documents into an appropriate presentation for the classification system. Several preprocessing techniques can be applied such as normalization, tokenization, stop word elimination, and stemming technique. After preprocessing the document modeling must be done, which is known as the method that extracts features from the text and converts it into the algebraic vector. Then the classification method can be done to constructs the classification model and model evaluation [5] [6].

These research paper analyze the effect of preprocessing processes as tokenization, normalization and removing stop words, stemming using Khoja stemmer, stemming using light

stemmer, stemming using Khoja stemmer with tokenization, normalization and removing stop words, and stemming using light stemmer with tokenization, normalization and removing stop words on document classification. To achieve this goal a data mining and machine learning tool Rabid Miner are used, the implementation process is done using three classifiers naive Bayes (NB), k-nearest neighbor (KNN), and support vector machine (SVM). These algorithms are applied using Arabic document corpus prepared by Diab Abuaiadh. It is composed of 2700 documents equally spread across nine categories (Arts, Economics, Health, Law, Literature, Politics, Religion, Sports and Technology). Whereas it contains 300 documents for each subject.

The structure of this research is organized as follows: Section 2 provides a related works that addressing the effect of preprocessing in Arabic text classification. Section 3 outlines Arabic language overview, Section 4 represent the research methodology, where it illustrating the relation between machine learning and preprocessing. Section 5 details the experiments and results evaluation. Finally, Section 6 concludes the research.

2. RELATED WORKS

Ismail Hmeidi et. al., [4] concerned with text classification of Arabic articles. It contains a comparison of the five best known algorithms for text classification. It also studies the effects of utilizing different Arabic stemmers (light and root-based stemmers) on the effectiveness of these classifiers. Furthermore, a comparison between different data mining software tools (Weka and Rapid Miner) is presented. The results illustrate the good accuracy provided by the SVM classifier, especially when used with the light10 stemmer.

Yousif A. Alhaj et. al., [5] aimed to study the impact of stemming techniques, as Information Science Research Institute (ISRI), Tashaphyne, and ARLStem on Arabic document classification. They used three classification algorithms, as Naïve Bayesian (NB), support vector machine (SVM), and K-nearest neighbors (KNN). The chi-square feature selection is used to select the most relevant features. Experiments are conducted on CNN Arabic corpus to assess the performance of the classification system. To evaluate the classifiers, authors used the K-fold cross-validation method and Micro-F1. The results indicate that the ARLStem outperforms the ISRI and Tashaphyne stemmers.

Abdullah Ayedh et. al., [7] studied the effect of three classification techniques Naive Bayes (NB), k-Nearest Neighbor (KNN), and Support Vector Machine (SVM). Experimental analysis on Arabic datasets reveals that preprocessing techniques have a significant impact on the classification accuracy, especially with complicated morphological structure of the Arabic language. Findings of this study show that the SVM technique has outperformed the KNN and NB techniques.

Adel H. Mohammad et. al., [8] applied three well known classification algorithm. Algorithm applied are K-Nearest Neighbour (KNN), C4.5 and Rocchio algorithm. These well-known algorithms are applied on in-house collected Arabic data set. Data set used consists from 1400 documents belongs to 8 categories. Results show that precision and recall values using Rocchio classifier and KNN are better than C4.5.

Roiss Alhutaish, and Nazlia Omar [9] investigated the use of the K-Nearest Neighbour (KNN) classifier, with an I_{new}, cosine, jaccard and dice similarities, in order to enhance Arabic Automatic Text Categorization. they represent the dataset as un stemmed and stemmed data; with the use of TREC-2002, in order to remove prefixes and suffixes. However, for statistical text representation, Bag-Of-Words (BOW) and character-level 3 (3-Gram) were used. In order to, reduce the dimensionality of feature space; they used several feature selection methods. Experiments conducted with Arabic text showed that the K-NN classifier, with the new method similarity I_{new} 92.6% Macro-F1, had better performance than the K-NN classifier with cosine, jaccard and dice similarities.

Rehab M. Duwairi [10] compared the performance of three classifiers for Arabic text categorization, using Naïve Bayes, K-nearest-neighbors (KNN), and distance- based. They use preprocessed documents by removing punctuation marks and stop words, stemming was used to reduce dimensionality of feature vectors of documents. The accuracy of the classifiers is compared using recall, precision, error rate and fallout. The results show that Naïve Bayes outperformed other classifiers.

Rehab M. Duwairi, Islam Qarqaz [11] deal with sentiment analysis in Arabic reviews from a machine learning perspective. Three classifiers were applied on an in-house developed dataset of tweets/comments. In particular, the Naïve Bayes, SVM and K-Nearest Neighbor classifiers were run on this dataset. The results show that SVM gives the highest precision while KNN (K=10) gives the highest Recall.

Rouhia M. Sallam et. al., [12] proposed approach to achieve highest categorization accuracy. The proposed approach uses Frequency Ratio Accumulation Method (FRAM) as a classifier. The proposed approach is tested with known datasets. The experiments are done without both of normalization and stemming, with one of them, and with both of them. The obtained results of proposed approach are generally improved compared to existing techniques.

Carlos Adriano Gon¸calves et. al., [13] analyzed the impact of pre-processing techniques in text Classification of MEDLINE English documents. then assessed the effect of combining different pre-processing techniques together with several classification algorithms available in the WEKA tool. This research show that the application of pruning, stemming and WordNet reduces significantly the number of attributes and improves the accuracy of the results.

Abdullah Y. Muaad et. al., [14] aimed to identify the effectiveness of machine learning (ML) algorithms through

preprocessing and representation techniques. Authors in this study use various feature selection algorithms. They use classifiers such as multinomial Naive Bayes (MNB), Bernoulli Naive Bayes (BNB), Stochastic Gradient Descent (SGD), Support Vector Classifier (SVC), Logistic Regression (LR), and Linear SVC. All of these AI classifiers are evaluated using five balanced and unbalanced benchmark datasets: BBC Arabic corpus, CNN Arabic corpus, OpenSource Arabic corpus (OSAc), ArCovidVac, and AlKhaleej. The evaluation results show that the classification performance strongly depends on the preprocessing technique, representation methods and classification technique, and the nature of datasets used.

Mahmoud Masadeh et. al., [15] used preprocessing techniques and representation models to enhance the overall classification performance, and evaluate the effectiveness of Arabic text classification using Machine Learning (ML), depends on various factors, such as stemming in preprocessing, feature extraction and selection, and the nature of the dataset., preprocessing methodologies were used to transform each Arabic term into its root form and reduce the dimensionality of representation. In the representation of Arabic text, feature extraction and selection processes were imperative, as they significantly enhance the performance of Arabic text classification. This study implemented the chosen classifiers using various feature selection algorithms. The comprehensive assessment of classification outcomes is conducted by comparing various classifiers, including Multinomial Naive Bayes (MNB), Bernoulli Naive Bayes (BNB), Stochastic Gradient Descent (SGD), Support Vector Classifier (SVC), Logistic Regression (LR), and linear Support Vector Classifier (LSVC). These ML classifiers were assessed utilizing short and long Arabic text benchmark datasets called BBC Arabic corpus and the COVID-19 dataset.

Riyad Al-Shalabi, and Rasha Obeidat [16] presented the results of classifying Arabic language documents by applying the KNN classifier, one time by using N-Gram namely unigrams and bigrams in documents indexing, and another time by using traditional single terms indexing method (bag of words) which supposes that the terms in the text are mutually independent which is not the case. Results show that using N-Grams produces better accuracy than using Single Terms for indexing; the average accuracy of using N-grams is 74%, while with Single terms indexing the average accuracy is 67%.

Jaffar Atwan et. al., [17] presented the implementation of a Naïve Bayes classifier for Arabic text with and without stemmer. A set of four categories and 800 documents were used from the Text Retrieval Conference (TREC) 2001 dataset. The results showed that Naïve Bayes with light stemmer achieves better results than Naïve Bayes without stemmer.

Anoual El Kah1, and Imad Zeroual [18] investigated the impact of selected preprocessing techniques on the efficiency of different text classification algorithms. The effects of stop words removal, stemming, lemmatization, and all possible combinations are examined. The reported results (+10.75% to +28.73%) prove the effectiveness of using these techniques either individually or in combination.

Djelloul Bouchiha et. al., [19] presented a comparative empirical study to see which combination of feature extraction in ML algorithm acts well when dealing with Arabic documents. So, they implemented one hundred sixty classifiers by combining 5 feature extraction techniques and 32 machine learning algorithms. The experiments were carried out using a huge open dataset. The comparison study reveals that TFIDF-Perceptron is the best performing combination of a classifier.

3. ARABIC LANGUAGE OVERVIEW

Arabic language is one of the Semitic languages in antiquity and one of the six official languages of the United Nation [17]. It is an important language because there are more than 422 million people over the world spoke Arabic. It is a native language in the Arab world, it come in the second place after English. Arabic language has complicated morphology [7]. Whereas, Arabic is used in important applications same as English. These applications as Medicine, Engineering, Science, etc. These study concern in Arabic language because there is few research in Arabic language processing field.

Some of what distinguishes the Arabic language from other languages. The Arabic alphabet consists of 28 characters (أ، ب، ت، ث، ج، ح، خ، د، ذ، ر، ز، س، ش، ص، ض، ط، ظ، ع، غ، ف، ق، ك، ل، م، ن، هـ، و، ي), In additional to the Hamza (ء) which is considered as a letter in some Arabic linguistic references. Arabic written from right to left, Arabic letter style different depending in the position if letter (beginning, middle or end of a word) [6]. For example, the letter haa (ه), has the following style (هـ) when appears in the beginning. But it become in this (ـهـ) style when appear in the middle of word. And it will take this (ـه) style in the end of the word. Other issues that affect Arabic language is the diacritics which are symbols placed above or below the letter such as sada, dama, fathah, kasra, sukon, double dama, double fathah, double kasra. So, this diacritic makes parsing Arabic text a non-trivial task [7][10].

Arabic language is a very rich language and complex morphology in comparison with English. This richness comes from the size of vectors created. Arabic language has filtering mechanisms. Most words in Arabic language can be mapped into their roots using stemming. Root in Arabic language available in three, four, five and six letters. Also, over 80% of Arabic words can be mapped into three-letter root. In English, prefixes and suffixes are added to the beginning or end of the root to create new words. In Arabic, in addition to the prefixes and suffixes there are infixes that can be added inside the word to create new words that have the same meaning. In the Arabic language, the problem of synonyms and broken plural forms are

widespread. In the Arabic language, one word may have more than lexical category (noun, verb, adjective) [7][8].

There is amount of research studies that conducted in Arabic text classification most of these studies try to find an effective and accurate way to classify Arabic text. But these studies found different factors that we must take in consideration such as dataset, Preprocessing, data division, feature extraction, and feature selection [20]. From here, can say for any researcher who want to process Arabic text, must have a good preprocessing mechanism which include: tokenization, normalization, remove stop words, and stemming using any Arabic stemmers.

4. RESEARCH METHODOLOGY

4.1 Document Preprocessing

For applying machine learning in natural language processing a very important tasks or techniques must be done namely document pre-processing, it is required to transform the text into an understandable format [3]. Preprocessing techniques are mainly used to convert each Arabic word into its root and decrease the representation dimension, ambiguity among the datasets, and increase the accuracy and effectiveness of the classification system, to be a suitable form for the representation step. To perform preprocessing, many commonly used tasks namely tokenization, normalization, stop word removal, and stemming, which can be done using RapidMiner tool [7][14][17][21]. Figure 1 shows a general overview of architecture of Arabic documents classification using machine learning.

4.1.1 Tokenization

Tokenization is a method for dividing texts into tokens. Words are often separated from each other and delimited by blanks (white space, semicolons, commas, quotes, and periods) [7]. In tokenization, text is divided into units, and typically here, those units are words [6]. These tokens might be separate words, sentences or paragraphs, the output of tokens becomes the input for the next preprocessing step [5].

4.1.2 Normalization

Normalization is a process that converts a list of words to a more uniform sequence [6]. Normalization aims to normalize certain letters that have different forms in the same word to one form (canonical form). As an example Hamza ‘‘ ء ’’ in (أ, إ, آ) into ا, Taa Marboutah ‘‘ ة ’’ into ‘‘ ه ’’, Yaa Mamdoda ‘‘ ي ’’ into ‘‘ ى ’’, it aims also to remove the Diacritics (Tashkeel), elongation (Kashida), and duplicate letters, as an example in ‘‘ الْعَــرَبِيَّةُ ’’ into ‘‘ العربيه ’’ [7] [19].

4.1.3 Removing Stop Words and Special Character

Removing stop words means elimination of insignificant words, that don't have any effect on the text, and do not have any meaning or indications about the content. These words repeated many times in text, it has high weights. whereas removing it will improve and speed text processing. These words can be prepositions (such as in: في, on: على, from: من, to: الى, and about: حول), pronouns (such as he: هو, she: هي, and they: هم), demonstratives (such as this: هذا, these: هؤلاء, those: أولئك, and there: هنالك), Conjunctions (and: و, or: أو, until: حتى , and then: ثم), Special character and numbers removal, these special characters as (+, -, !, ?, ., ,, ;, :, {, }, ≠, =, &, #, %, $, [,], /, <, >, n, \, '',(,),). For Arabic language there is a list of 896 stop words was prepared to be eliminated from all the documents [7] [13].

4.1.4 Stemming

In information retrieval, machine learning, and natural language processing, stemming techniques are considered as an essential preprocessing stage before tackling any task of document classification [5]. Stemming means the process of removing all affixes (such as prefixes, infixes, and suffixes) from words, it returns the word to its root or stem [7]. Two efficient stemming algorithms: namely, Khoja stemmer, and light stemmer are applied in these research paper:

- ***Khoja Stemmer*:** is an Arabic stemmer that developed by Shreen Khoja, it is fast and highly accurate. The first thing the stemmer does is remove the longest suffix and the longest prefix, then matches the remaining word with the verbal and noun patterns, to extract the root. The stemmer has been developed in both C++ and Java [22]. Khoja considered as root stemming, which is also called heavy stemming, aims to transform a word to its root. In Arabic, most word roots consist of three letters, the results of root-stemmed words will be mostly words made from three letters [6]. For example, the words (teachers, “المعلمون”), (teacher, “المعلم”), (teacher (Feminine), “معلمه”), (learner, “متعلم”), (scientist, “عالم”) are derived from the same root (science, “علم”), or the same stem (teacher, “معلم”).

- ***Light stemmer*:** is the process of removing the very often prefixes and suffixes based on a predefined list of suffixes and prefixes. Unlike Khoja stemmer, the light stemming technique is not meant to retrieve the root of a selected Arabic word, this technique is not concern for dealing with infixes or recognize patterns. Many light stemmers have been recommended for the Arabic language, whereas there is no standard algorithm for Arabic light stemming. So that in some cases, when light stemmer truncates affixes from the word, it may yield a wrong stem for example (‘‘بستان’’ into ‘‘بستا’’) [5].

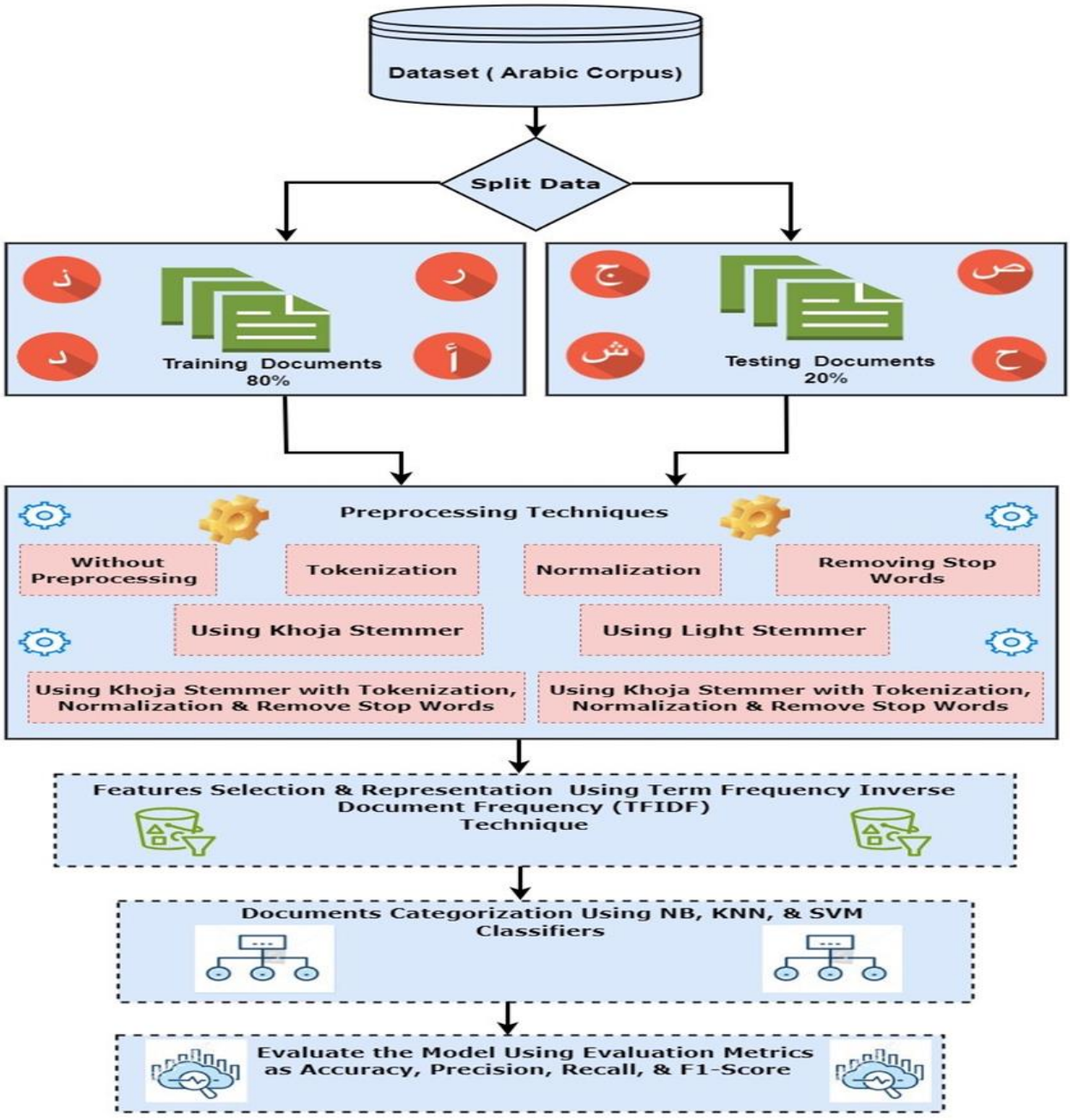

Figure 1: Architecture of Arabic Document Classification Using Machine Learning

4.2 Features Selection

In every dataset, two kinds of features are founded external features or irrelevant features which are not related to the content of the text such as Author name, publisher, year of publish, and number of pages, and internal features or relevant which are related to text content such as lexical items, single or compounds, grammatical categories and semantic relations [1].

Reducing the dimensionality of the dataset is the sconed most important task that are must done after preprocessing. It can be achieved by using features selection to improve the performance of document classification by removing the external features that are considered irrelevant for classification [9]. Whereas, Features selection considers as representation of text and the most important steps to make data ready for further processing so that machines can understand data [21]. RapidMiner tool provides features

selection task using Term Frequency Inverse Document Frequency (TFIDF) technique. In TFIDF computes the importance of a word within a text. It defined as the product of Term Frequency (TF) and Inverse Document Frequency (IDF) [19].

4.3 Supervised Learning Algorithms (Classifiers)

In machine learning there are two disciplines supervised learning algorithms which used in prediction, and classification, the other are unsupervised learning algorithms which used in clustering. the supervised algorithms or classifiers defined as Naïve Bayes, K-Nearest Neighbor, and Support Vector Machine (SVM) etc.

4.3.1 Naïve Bayes (NB)

These classifiers are among the simplest Bayesian network models, it is one of the most-used algorithms in text classification, NB is a classifier that depends on probabilistic condition. It uses Bayes Theorem "naive" assumption of conditional independence between every pair of features given the value of the class variable being classified is independent of each other, which assumes that the features such as tokens are conditionally independent [4].

It is highly used in text classification. In text and documents classification tasks, data contains high dimension (as each word represent one feature in the data). It is used in spam filtering, sentiment detection, documents classification etc. The advantage of using naïve Bayes is its speed. It is fast and making prediction is easy with high dimension of data.

4.3.2 K-Nearest Neighbor (KNN)

KNN is a classifier with low bias and high variance, the main advantages of KNN include simplicity, asymptotic performance, ease of implementation and when choosing features appropriately it results decent accuracy. Whereas the disadvantages of KNN such as poor accuracy if K is not selected properly, sensitivity to irrelevant parameters and the need for a proper similarity measure such as the Cosine measure [4].

The k-NN method uses the well-known principle of (birds of a feather flock together or literally equals with equals easily associate). suppose that a set of samples with known classification is available, the so-called training set. Intuitively, each sample should be classified similarly to its surrounding samples. Therefore, if the classification of a sample is unknown, then it could be predicted by considering the classification of its nearest neighbor samples [23].

4.3.3 Support Vector Machine classifier (SVM)

Support Vector Machine (SVM) can be used for classification or regression. For SVM as classifier, it builds an N dimensional hyper-plane that perfectly splits the data into two classes. It learns from a set of training data and predicts for each test or new tuple its probable class. In text documents, SVM is considered an excellent classifier, it gives high accuracy. It is used to solve high dimensionality problems in document classification. Disadvantages of SVM include complexity, poor interpretability and high memory requirements [4].

SVM is one of the classical machine learning techniques that can help the multidomain applications in a big data environment [24]. It distinctively affords balanced predictive performance, even in studies where sample sizes may be limited [25]. SVM approach based on the following [26]:

- **Class separation**: looking for the optimal separating hyperplane between the two classes by maximizing the margin between the classes' closest points, the points lying on the boundaries are called support vectors, and the middle of the margin is our optimal separating hyperplane.
- **Overlapping classes**: data points on the "wrong" side of the discriminant margin are weighted down to reduce their influence ("soft margin").
- ***Nonlinearity:*** when we cannot find a linear separator, data points are projected into a higher-dimensional space where the data points.
- **Problem solution:** the whole task can be formulated as a quadratic optimization problem.

5. EXPERIMENTS AND RESULTS

5.1 Experiments

To analyzing the effect of preprocessing in Arabic text classification. These documents were processed by several different ways as tokenization, normalization and removing stop words, stemming using Khoja stemmer, stemming using light stemmer, stemming using Khoja stemmer with tokenization, normalization and removing stop words, and stemming using light stemmer with tokenization, normalization and removing stop words on document classification. To achieve this goal a data mining and machine learning tool Rabid Miner were used, the implementation process was done using three classifiers naive Bayes (NB), k-nearest neighbor (KNN), and support vector machine (SVM). The experiments were conducted on a system equipped with an Intel Core i5-1135G7 processor, running at 2.4 GHz, and 8 GB of RAM. Using a dataset that are mentioned in 5.1.1, four standard metrics were used to evaluate the effect of preprocessing in document classification, namely Accuracy, Precision, Recall, and F1-Score, it will be discussed in 5.1.2, the results were came as illustrated in 5.2.

5.1.1 Dataset

There are many free datasets written as text documents on the web, it collected in corpus, it is used to take benefit from it, ant to conduct research and studies on it. In this research, an online Arabic corpus prepared by Diab Abuaiadh are used, this corpus contains 2700 documents, divided between nine Arabic subjects each subject has 300 documents, this subjects such as: art, economy, health. Law, literature, politics, religion, sport, and technology. The experiment done on four phases, in the

first phase chose 50 documents form each subject randomly, then chose 100, 200, 300 documents from each subject respectively.

5.1.2 Evaluation Metrics

In this paper four standard metrics were used to evaluate the effect of preprocessing techniques on Arabic documents classification. It represents the most useful and widely used methods for evaluating classifiers, this metrics based on confusion matrix that implemented in figure 2, namely Accuracy, Precision, Recall, and F1-Score which illustrated in equations (1), (2), (3), and (4), where TP represents all the documents which are indicated correctly to the specified category. TN represents all the documents which are correctly indicated not to belong to the category. FP represents all documents which are incorrectly indistinct to the category. FN represents all documents which are incorrectly not defined to the category.

	Actual Category	
Predicted Category	**Ture Positive (*TP*)**	**False Positive (*FP*)**
	False Negative (*FN*)	**Ture Negative (*TN*)**

Figure 2: Confusion Matrix

For each one of this metrics an important and crucial meaning which come to measure certain value as following:

- Accuracy (*ACC*) measures the classifier efficiency depends on its proportion of correct projections, represents the number of correctly categorized data instances over the whole number of data samples:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

- Precision (*P*) measures the number of positive categories, meaning positive predictive value, describes how many of the precisely foreseen cases were positive:

$$P = \frac{TP}{TP+FP} \quad (2)$$

- Recall (*R*) measures the performance of a model to predict all the positive categories, meaning sensitivity or true positive rate:

$$R = \frac{TP}{TP+FN} \quad (3)$$

- F1-Score (*F1*) measures the harmonic mean of precision and recall, it is a more reasonable measure than accuracy, represented in equation (4), and (5):

$$F1 = 2 \times \frac{P \times R}{P+R} \quad (4)$$

Or

$$F1 = \frac{2 \times TP}{2 \times TP + FP + FN} \quad (5)$$

5.2 Results

Figure 3 illustrates the superior performance of SVM algorithm in increasing overall accuracy, particularly evident in all scenarios for 50, 100, 200, and 300 documents, which involving documents without preprocessing, documents with tokenization, normalization and removing stop words, documents with stemming using Khoja stemmer, documents with stemming using light stemmer, documents with stemming using Khoja stemmer with tokenization, normalization and removing stop words, and documents with stemming using light stemmer with tokenization, normalization and removing stop words. The results show the superiority of SVM over KNN and NB, whether in cases of prepressing or without it.

Comparing the accuracy for documents without preprocessing, with preprocessed documents using different techniques, it becomes apparent that preprocessing techniques have an important effect on document classification accuracy. Whereas document without preprocessing consistently demonstrates the lowest accuracy when benchmarked against documents with tokenization, normalization and removing stop words, documents with stemming using Khoja stemmer, documents with stemming using light stemmer, documents with stemming using Khoja stemmer with tokenization, normalization and removing stop words, and documents with stemming using light stemmer with tokenization, normalization and removing stop words. Notably, documents without preprocessing achieves reduction in accuracy when using Naïve Bayes classifier by 20.3% compared to documents with tokenization, normalization and removing stop words, 2.3% compared to documents with stemming using Khoja stemmer, 15.9% compared to documents with stemming using light stemmer, 22.3% compared to documents with stemming using Khoja stemmer with tokenization, normalization and removing stop words, and 23.6% compared to documents with stemming using light stemmer with tokenization, normalization and removing stop words.

Documents without preprocessing achieves reduction in accuracy by 21.5%, 17.9%, 16.5%, 21.6%, and 22.4% compared to documents with tokenization, normalization and

removing stop words, documents with stemming using Khoja stemmer, documents with stemming using light stemmer, documents with stemming using Khoja stemmer with tokenization, normalization and removing stop words, and documents with stemming using light stemmer with tokenization, normalization and removing stop words respectively, when using KNN classifier. While it achieves reduction in accuracy by 10.5%, 9.7%, 10.5%, 10.3%, and 11.8% compared to documents with tokenization, normalization and removing stop words, documents with stemming using Khoja stemmer, documents with stemming using light stemmer, documents with stemming using Khoja stemmer with tokenization, normalization and removing stop words, and documents with stemming using light stemmer with tokenization, normalization and removing stop words respectively, when using SVM classifier.

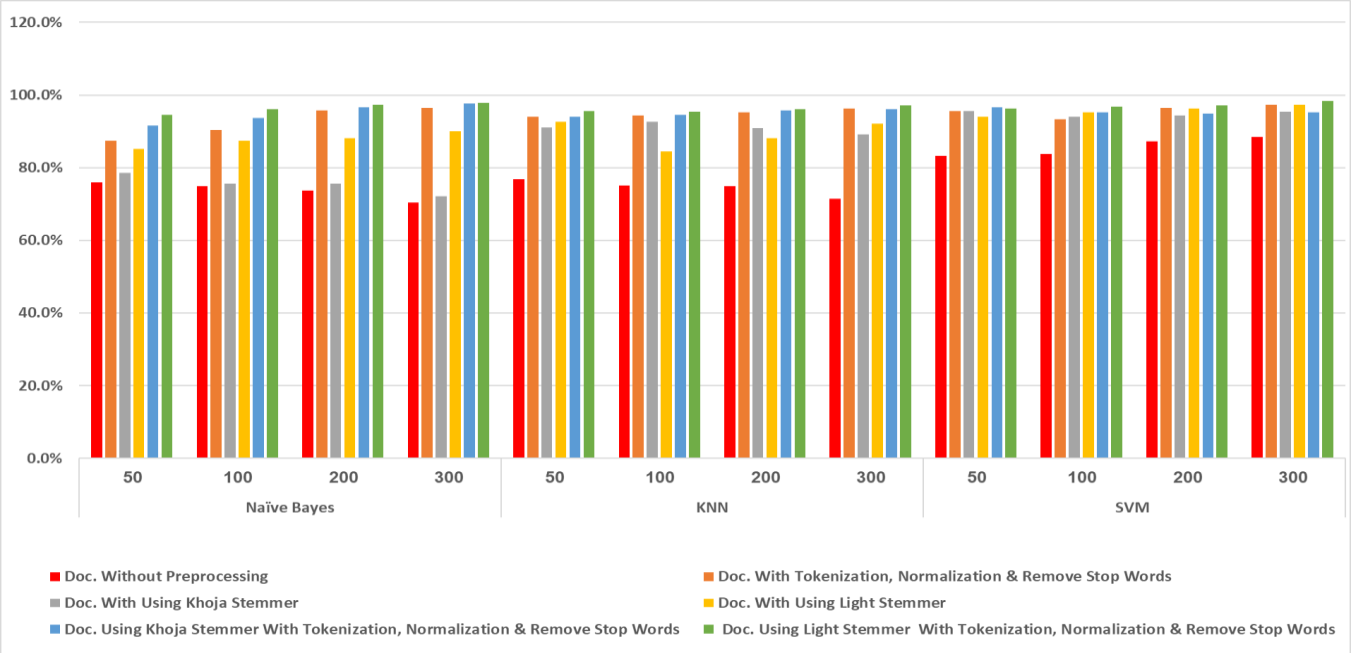

Figure 3: Accuracy for NB, KNN, and SVM Using Preprocessing, and without Using Preprocessing Techniques

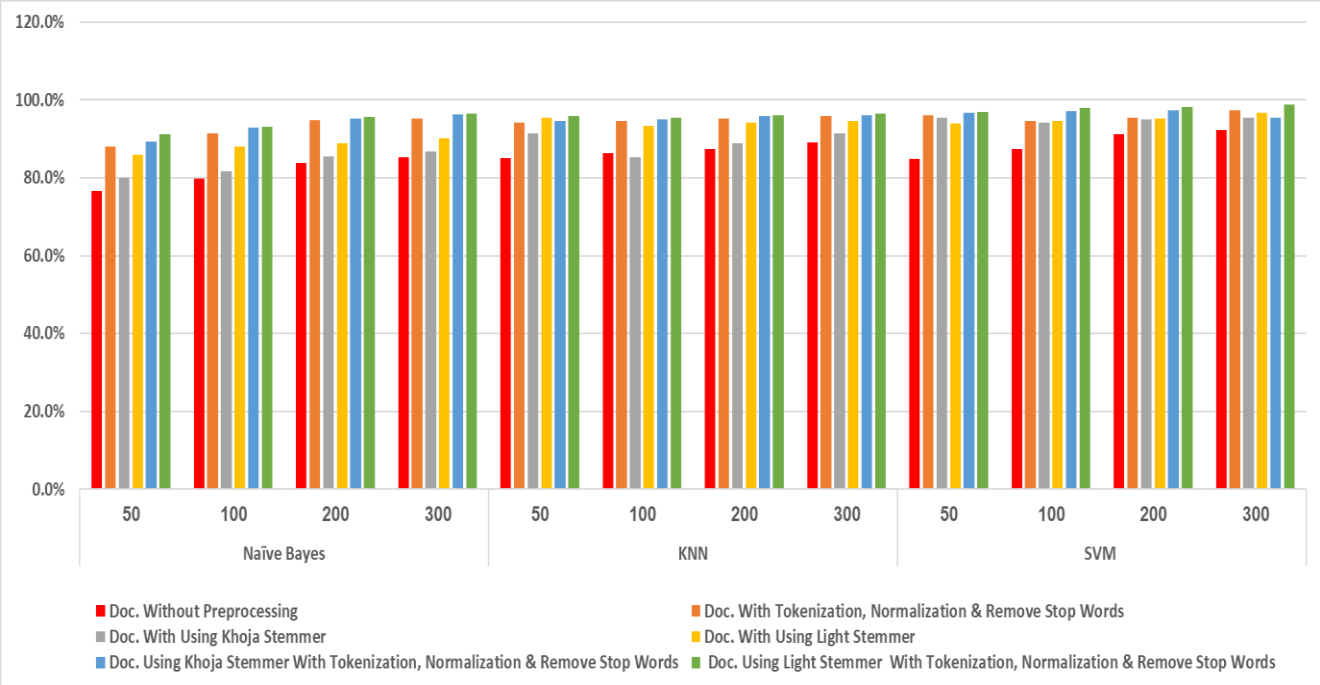

Figure 4: Precision for NB, KNN, and SVM Using Preprocessing, and without Using Preprocessing Techniques

SVM classifier, stands out as a highly efficient solution, consistently achieving the highest overall precision when compared to NB, and KNN, this superiority is vividly depicted in Figure 4. The graph underscores specifically that, documents without preprocessing demonstrates a substantial reduction in precision compared to other preprocessing techniques. The documents without preprocessing technique when using NB classifier, achieves remarkable precision reductions, including 11.9% compared to documents with tokenization, normalization and removing stop words, 2.5% compared to documents with stemming using Khoja stemmer, 7.8% compared to documents with stemming using light stemmer, 12.9% compared to documents with stemming using Khoja stemmer with tokenization, normalization and removing stop words, and 13.5% compared to documents with stemming using light stemmer with tokenization, normalization and removing stop words.

Documents without preprocessing achieves reduction in precision by 8.4%, 2.6%, 7.7%, 8.8%, and 9.3% compared to documents with tokenization, normalization and removing stop words, documents with stemming using Khoja stemmer, documents with stemming using light stemmer, documents with stemming using Khoja stemmer with tokenization, normalization and removing stop words, and documents with stemming using light stemmer with tokenization, normalization and removing stop words respectively, when using KNN classifier. While it achieves reduction in precision by 7.3%, 6.4%, 6.5%, 8.0%, and 9.2% compared to documents with tokenization, normalization and removing stop words, documents with stemming using Khoja stemmer, documents with stemming using light stemmer, documents with stemming using Khoja stemmer with tokenization, normalization and removing stop words, and documents with stemming using light stemmer with tokenization, normalization and removing stop words respectively, when using SVM classifier.

In the depicted Figure 5, The SVM classifier remains dominant in terms of overall recall, it overcome over NB, and KNN classifiers. And to conduct comparison regarding the recall achieved by documents using preprocessing techniques, and documents without preprocessing. The horizontal axis represents the number of documents used by the three classifiers, while the vertical axis indicates recall percentage. These results unequivocally establish, that documents without preprocessing achieves reduction in recall when using Naïve Bayes classifier by 11.5% compared to documents with tokenization, normalization and removing stop words, 5.6% compared to documents with stemming using Khoja stemmer, 8.2% compared to documents with stemming using light stemmer, 11.5% compared to documents with stemming using Khoja stemmer with tokenization, normalization and removing stop words, and 13.4% compared to documents with stemming using light stemmer with tokenization, normalization and removing stop words.

Documents without preprocessing achieves reduction in recall by 12.2%, 4.1%, 11.3%, 13.1%, and 14.2% compared to documents with tokenization, normalization and removing stop words, documents with stemming using Khoja stemmer, documents with stemming using light stemmer, documents with stemming using Khoja stemmer with tokenization, normalization and removing stop words, and documents with stemming using light stemmer with tokenization, normalization and removing stop words respectively, when using KNN classifier. While it achieves reduction in accuracy by 10.3%, 9.8%, 10.0%, 11.0%, and 12.0% compared to documents with tokenization, normalization and removing stop words, documents with stemming using Khoja stemmer, documents with stemming using light stemmer, documents with stemming using Khoja stemmer with tokenization, normalization and removing stop words, and documents with stemming using light stemmer with tokenization, normalization and removing stop words respectively, when using SVM classifier.

F1-Score, which measures the harmonic mean between precision and recall. It means that, F1 measuring the mean between positive categories, and performance of a model to predict all the positive categories. Figure 6 shows that, as mentioned earlier SVM classifier, outperform over other classifiers, consistently achieving the highest overall F1-Score when compared to NB, and KNN.

Comparing the F1-Score for documents without preprocessing, with preprocessed documents using different techniques. Whereas documents without preprocessing consistently demonstrates the lowest F1-Score when benchmarked against documents with preprocessing. This is clearly shown in percentages, where documents without preprocessing achieves lowest F1-score when using Naïve Bayes classifier by 11.7% compared to documents with tokenization, normalization and removing stop words, 4.0% compared to documents with stemming using Khoja stemmer, 8.0% compared to documents with stemming using light stemmer, 12.2% compared to documents with stemming using Khoja stemmer with tokenization, normalization and removing stop words, and 13.4% compared to documents with stemming using light stemmer with tokenization, normalization and removing stop words.

Documents without preprocessing achieves lowest F1-score also when using KNN, by 10.4%, 3.3%, 9.6%, 11.0%, and 11.8% compared to documents with tokenization, normalization and removing stop words, documents with stemming using Khoja stemmer, documents with stemming using light stemmer, documents with stemming using Khoja stemmer with tokenization, normalization and removing stop words, and documents with stemming using light stemmer with tokenization, normalization and removing stop words respectively, This is clearly shown in percentages. While it achieves reduction in F1-Score by 8.8%, 8.1%, 8.3%, 9.5%, and 10.7% compared to documents with tokenization, normalization and removing stop words, documents with

stemming using Khoja stemmer, documents with stemming using light stemmer, documents with stemming using Khoja stemmer with tokenization, normalization and removing stop words, and documents with stemming using light stemmer with tokenization, normalization and removing stop words respectively, when using SVM classifier.

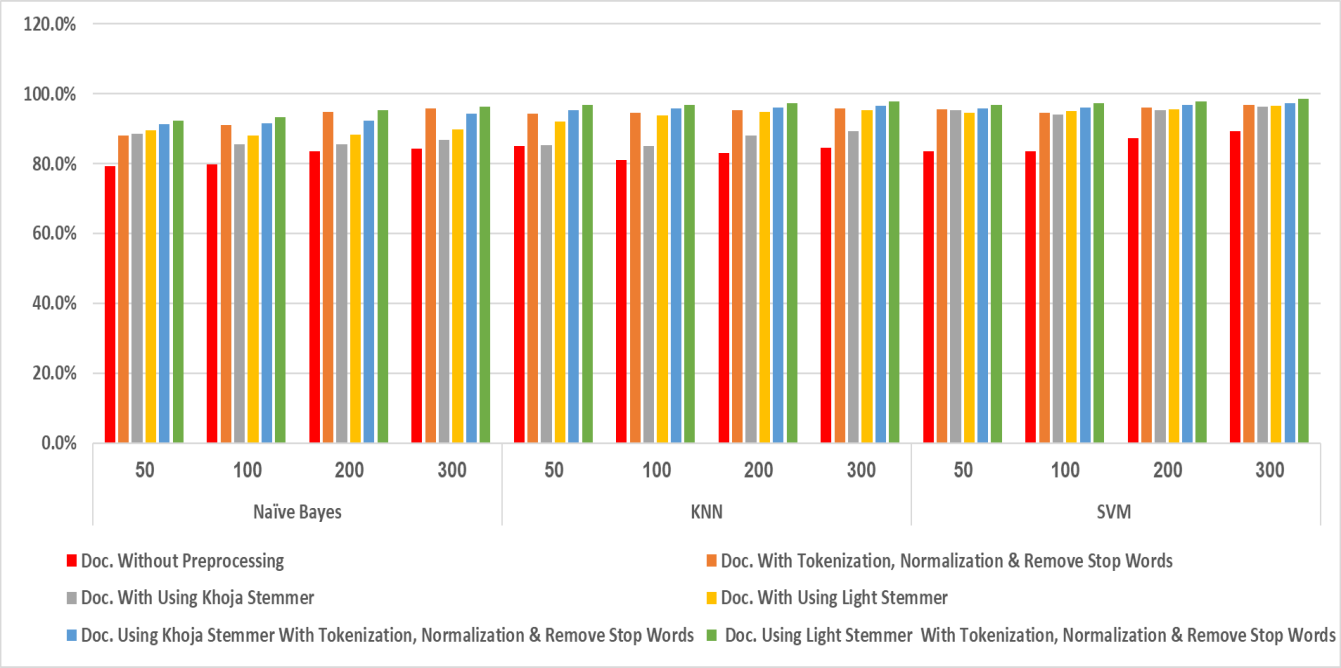

Figure 5: Recall for NB, KNN, and SVM Using Preprocessing, and without Using Preprocessing Techniques

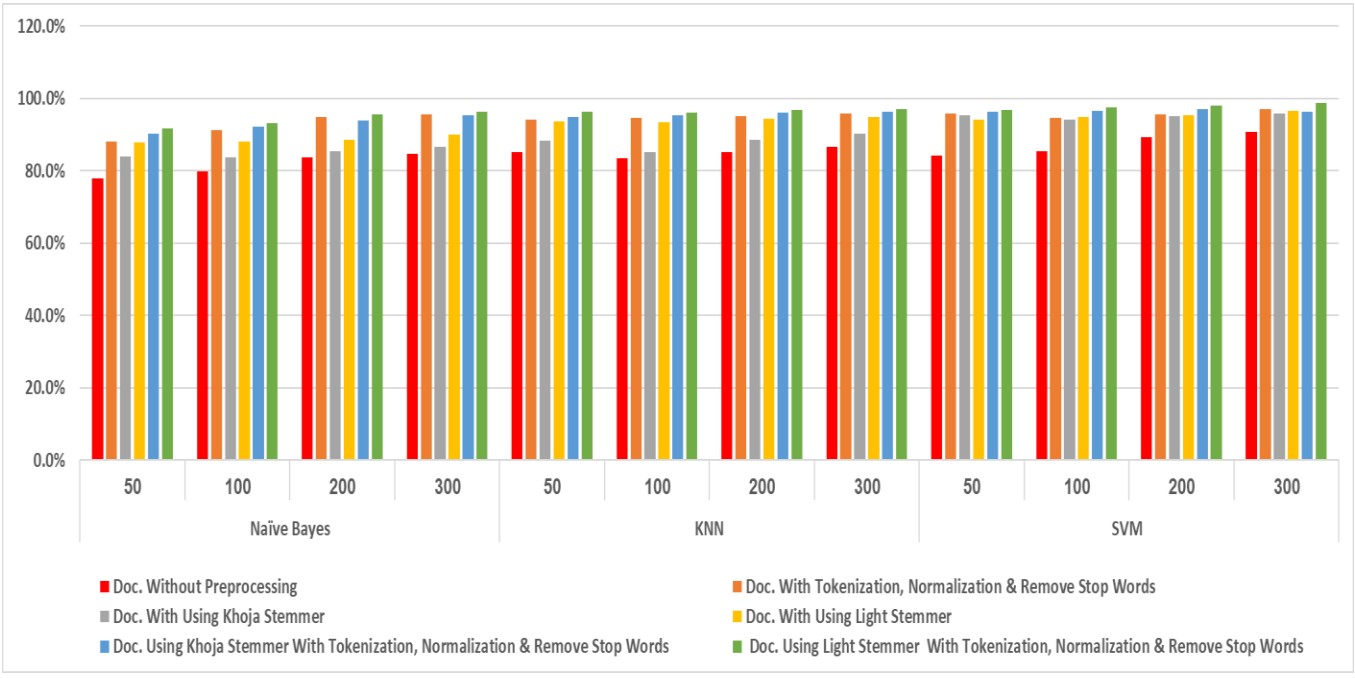

Figure 6: F1-Score for NB, KNN, and SVM Using Preprocessing, and without Using Preprocessing Techniques

In table 1, represents average accuracy, precision, recall, and F1-Score for the three classifiers NB, KNN, and SVM. It appears from the table that preprocessing has a meaningful effect on documents classification, as documents that were not preprocessed received a lower average accuracy, precision, recall, and F1-Score than documents which has undergone to preprocessing using different techniques. It also appears that the documents preprocessing using light stemmer with tokenization, normalization & remove stop words have the highest average accuracy, precision, recall, and F1-Score form other techniques when using the three classifiers NB, KNN, and SVM. Notice, in the table the following Abbreviations are used:

- Doc. W. P. (Documents Without Preprocessing)
- Doc. U. T. N. R. S. W. (Documents with Tokenization, Normalization & Remove Stop Words)
- Doc. U. K. S. (Documents with Using Khoja Stemmer)
- Doc. U. L. S. (Documents with Using Light Stemmer)
- Doc. U. K. S. T. N. R. S. W. (Documents Using Khoja Stemmer with Tokenization, Normalization & Remove Stop Words)
- Doc. U. L. S. T. N. R. S. W. (Documents Using Light Stemmer with Tokenization, Normalization & Remove Stop Words)

Table 1: Average Accuracy, Precision, Recall, and F1-Score for the Three Classifiers NB, KNN, and SVM

Doc. State	Naïve Bayes				KNN				SVM			
	ACC	P	R	F1	ACC	P	R	F1	ACC	P	R	F1
Doc. W. P.	73.7%	81.4%	81.7%	81.6%	74.6%	87.0%	83.4%	85.1%	85.7%	89.0%	85.9%	87.4%
Doc. U. T. N. R. S. W.	92.5%	92.4%	92.4%	92.4%	95.0%	95.0%	95.0%	95.0%	95.7%	95.9%	95.7%	95.8%
Doc. U. K. S.	75.5%	83.5%	86.5%	85.0%	90.9%	89.3%	86.9%	88.1%	94.8%	95.0%	95.3%	95.1%
Doc. U. L. S.	87.7%	88.3%	89.0%	88.6%	89.3%	94.3%	94.0%	94.1%	95.7%	95.2%	95.4%	95.3%
Doc. U. K. S. T. N. R. S. W.	94.9%	93.5%	92.4%	92.9%	95.1%	95.4%	95.9%	95.6%	95.5%	96.7%	96.5%	96.6%
Doc. U. L. S. T. N. R. S. W.	**96.5%**	**94.1%**	**94.3%**	**94.2%**	**96.1%**	**96.0%**	**97.2%**	**96.6%**	**97.2%**	**98.0%**	**97.7%**	**97.8%**

It is clear that preprocessing using light stemmer with tokenization, normalization & remove stop words technique enhances accuracy, precision, recall, and F1-Score when compared to all other technique Either with preprocessing or without preprocessing. The study establishes the effectiveness and robustness of this technique through these measurements. Where if compared preprocessing using light stemmer with tokenization, normalization & remove stop words technique, to document without preprocessing find that:

Accuracy Enhancement: that preprocessing using light stemmer with tokenization, normalization & remove stop words technique, demonstrates an enhanced accuracy compared to document without preprocessing, with increases of 23.6% when using NB, 22.4% when using KNN, and 11.8% when using SVM.

Precision Enhancement: that preprocessing using light stemmer with tokenization, normalization & remove stop words technique, demonstrates an enhanced precision compared to document without preprocessing, with increases of 13.5% when using NB, 9.3% when using KNN, and 9.2% when using SVM.

Recall Enhancement: that preprocessing using light stemmer with tokenization, normalization & remove stop words technique, demonstrates an enhanced recall compared to document without preprocessing, with increases of 13.4% when using NB, 14.2% when using KNN, and 12.0% when using SVM.

F1-Score Enhancement: that preprocessing using light stemmer with tokenization, normalization & remove stop words technique, demonstrates an enhanced F1-Score compared to document without preprocessing, with increases of 13.4% when using NB, 11.8% when using KNN, and 10.7% when using SVM.

6. CONCLUSION AND FUTURE WORK

Documents preprocessing is an indispensable service for documents classification, an important impact was achieved from preprocessing technique on Arabic documents. Numerous researchers are actively engaged in addressing this challenge. This paper introduces an analytical study about effect of preprocessing on Arabic document classification. Three classifiers have been introduced to underscore the efficacy of diverse preprocessing techniques in classifying Arabic documents, namely NB, KNN, and SVM. Whereas the experimentation indicates that representation, preprocessing, and feature selection, is essential in Arabic document classification. Simultaneously, diverse preprocessing techniques have a strong impact on Arabic documents classification, the findings clearly illustrate the benefits of documents with preprocessing techniques over documents without using preprocessing techniques.

Based on the analysis presented in this article, which is limited documents to one dataset (Diab Abuaiadh corpus). These documents with preprocessing techniques, or without using preprocessing techniques. And using the three classifiers BN, KNN, and SVM. the evaluation metrics indicts to superiority of SVM over KNN and NB, in increasing overall accuracy, precision, recall, and F1-Score. Also, all preprocessing techniques that are used outperformed documents without pre-processing by using all metrics. It also appears that preprocessing technique using light stemmer with tokenization, normalization & remove stop words outperform other used preprocessing techniques.

In the future work, it needs to further confirm the impact of preprocessing on the classification of Arabic documents, through three axes: The first axis, will be achieved by using several Arabic documents datasets, these datasets include different subjects. the second axis, will be achieved by using other classifiers such as Multinomial Naive Bayes (MNB), Bernoulli Naive Bayes (BNB), Stochastic Gradient Descent (SGD), Support Vector Classifier (SVC), Logistic Regression (LR), and linear Support Vector Classifier (LSVC). The third axis, will be based on other preprocessing techniques by using other types of stemmers.

REFERENCES

1. AbdulMohsen Al-Thubaity, Abdulrahman Almuhareb, Sami Al-Harbi, Abdullah Al-Rajeh, Mohammad Khorsheed. KACST Arabic Text Classification Project: Overview and Preliminary Results. Information Management in Modern Organizations: Trends & Challenge, 2008.
2. Omar Einea, Ashraf Elnagar, Ridhwan Al Debsi. SANAD: Single-label Arabic News Articles Dataset for automatic text categorization. Data in Brief, Volume 25, 2019,104076, ISSN 2352-3409, https://doi.org/10.1016/j.dib.2019.104076.
3. Tarik Sabri, Omar El Beggar, Mohamed Kissi. Comparative study of Arabic text classification using feature vectorization methods. Procedia Computer Science, Volume 198, 2022, Pages 269-275, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2021.12.239.
4. Ismail Hmeidi, Mahmoud Al-Ayyoub, Nawaf A. Abdulla, Abdalrahman A. Almodawar, Raddad Abooraig, Nizar A. Mahyoub. Automatic Arabic Text Categorization: A Comprehensive Comparative Study. Journal of Information Science, 2015, DOI: 10.1177/0165551514558172
5. Y. A. Alhaj, J. Xiang, D. Zhao, M. A. A. Al-Qaness, M. Abd Elaziz and A. Dahou. A Study of the Effects of Stemming Strategies on Arabic Document Classification. in *IEEE Access*, vol. 7, pp. 32664-32671, 2019, doi: 10.1109/ACCESS.2019.2903331.
6. Albalawi, Y., Buckley, J. & Nikolov, N.S. Investigating the impact of pre-processing techniques and pre-trained word embeddings in detecting Arabic health information on social media. J Big Data 8, 95, 2021, https://doi.org/10.1186/s40537-021-00488-w.
7. Ayedh A, TAN G, Alwesabi K, Rajeh H. The Effect of Preprocessing on Arabic Document Categorization. *Algorithms*. 2016, 9(2):27. https://doi.org/10.3390/a9020027.
8. Adel Hamdan Mohammad, Omar Al-Momani, and Tariq Alwada'n. Arabic Text Categorization using k-nearest neighbour, Decision Trees (C4.5) and Rocchio Classifier: A Comparative Study. International Journal of Current Engineering and Technology, 2016, E-ISSN 2277 – 4106, P-ISSN 2347 – 5161.
9. Roiss Alhutaish and Nazlia Omar. Arabic Text Classification using K-Nearest Neighbour Algorithm. The International Arab Journal of Information Technology, Vol. 12, No. 2, 2014.
10. Rehab Duwairi. Arabic Text Categorization. The International Arabic Journal of Information Technology, Vol. 4, No 2, April 2007.
11. R. M. Duwairi and I. Qarqaz. Arabic Sentiment Analysis Using Supervised Classification. *2014 International Conference on Future Internet of Things and Cloud*, Barcelona, Spain, 2014, pp. 579-583, doi: 10.1109/FiCloud.2014.100.
12. Sallam, Rouhia M., Hamdy M. Mousa, and Mahmoud Hussein. Improving Arabic text categorization using normalization and stemming techniques. *Int. J. Comput. Appl* 135.2, 2016, 38-43.
13. Gonçalves, Carlos & Gonçalves, Célia & Camacho, Rui & Oliveira, Eugénio. The Impact of Pre-processing on the Classification of MEDLINE Documents, 2010, 53-61.
14. Abdullah Y. Muaad, Hanumanthappa Jayappa Davanagere, D.S. Guru, J.V. Bibal Benifa, Channabasava Chola, Hussain AlSalman, Abdu H. Gumaei, Mugahed A. Al-antari. Arabic Document Classification: Performance Investigation of Preprocessing and Representation Techniques. *Mathematical Problems in Engineering*, vol. 2022, Article ID 3720358, 16 pages, 2022. https://doi.org/10.1155/2022/3720358.

15. Mahmoud Masadeh, Moustapha. A, Sharada B, Hanumanthappa J, Hemachandran K, Channabasava Chola and Abdullah Y. Muaad. Investigating the Impact of Preprocessing Techniques and Representation Models on Arabic Text Classification using Machine Learning. International Journal of Advanced Computer Science and Applications(IJACSA), 15(1), 2024. http://dx.doi.org/10.14569/IJACSA.2024.0150111 0.
16. Al-Shalabi, Riyad & Obeidat, Rasha. Improving KNN Arabic Text Classification with N-Grams Based Document Indexing. Proceedings of the Sixth International Conference on Informatics and Systems, 2008.
17. Jaffar Atwan, Mohammad Wedyan, Qusay Bsoul, Ahmad Hamadeen, Ryan Alturki and Mohammed Ikram. The Effect of Using Light Stemming for Arabic Text Classification. International Journal of Advanced Computer Science and Applications (IJACSA), 12(5), 2021. http://dx.doi.org/10.14569/IJACSA.2021.0120589 .
18. El Kah, A., & Zeroual, I. The effects of pre-processing techniques on Arabic text classification. *Int. J*, *10*(1), Volume 10, No.1, January - February 2021, 1-12.
19. D. Bouchiha, A. Bouziane, and N. Doumi. Machine Learning for Arabic Text Classification: A Comparative Study. *Malaysian J. Sci. Adv. Tech.*, vol. 2, no. 4, pp. 163–173, Oct, 2022.
20. Musab Mustafa Hijazi, Akram M.Zeki, and Amelia Ritahani Ismail. Arabic Text Classification: Review Study. Journal of Engineering and Applied Sciences, 11 (3): 528-536, 2016.
21. Abdullah Y. Muaad, G. Hemantha Kumar, J. Hanumanthappa, J.V. Bibal Benifa, M. Naveen Mourya, Channabasava Chola, M. Pramodha, R. Bhairava. An effective approach for Arabic document classification using machine learning. Global Transitions Proceedings, Volume 3, Issue 1, 2022, Pages 267-271, ISSN 2666-285X, https://doi.org/10.1016/j.gltp.2022.03.003.
22. Khoja S., Garside R. Stemming Arabic text. Computer Science Department, Lancaster University, Lancaster, UK, 1999.
23. Mucherino, A., Papajorgji, P.J., Pardalos, P.M.bk-Nearest Neighbor Classification. In: Data Mining in Agriculture. Springer Optimization and Its Applications, vol 34. Springer, New York, NY, 2009, https://doi.org/10.1007/978-0-387-88615-2_4.
24. Suthaharan, S. Support Vector Machine. In: Machine Learning Models and Algorithms for Big Data Classification. Integrated Series in Information Systems, vol 36. Springer, Boston, MA, 2016, https://doi.org/10.1007/978-1-4899-7641-3_9.
25. Derek A. Pisner, David M. Schnyer. Chapter 6 - Support vector machine. Editor(s): Andrea Mechelli, Sandra Vieira, Machine Learning, Academic Press, 2020, Pages 101-121, ISBN 9780128157398, https://doi.org/10.1016/B978-0-12-815739-8.00006-7.
26. Meyer, David. Support Vector Machines The Interface to libsvm in package e1071, 2001, Vol.1/3, 9.2001, R News. 1.