algorithms

MDPI

Article

The Effect of Preprocessing on Arabic Document Categorization

Abdullah Ayedh [1], Guanzheng TAN [1,*], Khaled Alwesabi [1] and Hamdi Rajeh [2]

[1] School of Information Science and Engineering, Central south University, Changsha 410000, China; abdullah_ayedh@csu.edu.cn (A.A.); k_alwesabi@csu.edu.cn (K.A.)

[2] College of Computer Science and Electrical Engineering, Hunan University, Changsha 410000, China; hamdiahmed919@gmail.com

* Correspondence: tgz@csu.edu.cn; Tel.: +86-139-7318-8535

Academic Editor: Tom Burr

Received: 21 January 2016; Accepted: 12 April 2016; Published: 18 April 2016

Abstract: Preprocessing is one of the main components in a conventional document categorization (DC) framework. This paper aims to highlight the effect of preprocessing tasks on the efficiency of the Arabic DC system. In this study, three classification techniques are used, namely, 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. Choosing appropriate combinations of preprocessing tasks provides significant improvement on the accuracy of document categorization depending on the feature size and classification techniques. Findings of this study show that the SVM technique has outperformed the KNN and NB techniques. The SVM technique achieved 96.74% micro-F1 value by using the combination of normalization and stemming as preprocessing tasks.

Keywords: document categorization; text preprocessing; stemming techniques; classification techniques; Arabic language processing

1. Introduction

Given the tremendous growth of available Arabic text documents on the Web and databases, researchers are highly challenged to find better ways to deal with such huge amount of information; these methods should enable search engines and information retrieval systems to provide relevant information accurately, which had become a crucial task to satisfy the needs of different end users [1].

Document categorization (DC) is one of the significant domains of text mining that is responsible for understanding, recognizing, and organizing various types of textual data collections. DC aims to classify natural language documents into a fixed number of predefined categories based on their contents [2]. Each document may belong to more than one category.

Currently, DC is widely used in different domains, such as mail spam filtering, article indexing, Web searching, and Web page categorization [3]. These applications are increasingly becoming important in the information-oriented society at present.

The document in text categorization system must pass through a set of stages. The first stage is the preprocessing stage which consists of document conversion, tokenization, normalization, stop word removal, and stemming tasks. The next stage is document modeling which usually includes the tasks such as vector space model construction, feature selection, and feature weighting. The final stage is DC wherein documents are divided into training and testing data. In this stage, the training process uses the proposed classification algorithm to obtain a classification model that will be evaluated by means of the testing data.

The preprocessing stage is a challenge and affects positively or negatively on the performance of any DC system. Therefore, the improvement of the preprocessing stage for highly inflected language such as the Arabic language will enhance the efficiency and accuracy of the Arabic DC system. This paper investigates the impact of preprocessing tasks including normalization, stop word removal, and stemming in improving the accuracy of Arabic DC system.

The rest of this paper is organized as follows. Related works are presented in Sections 2 and 3 provides an overview of the Arabic language structure. An Arabic DC framework is described in Section 4. Experiment and results are presented in Section 5. Finally, conclusion and future works are provided in Section 6.

2. Related Work

DC is a high valued research area, wherein several approaches have been proposed to improve the accuracy and performance of document categorization methods. This section summarizes what has been achieved on document categorization from various pieces of the literature.

For the English language, the impact of preprocessing on document categorization are investigated by Song *et al.* [4]. It is concluded that the influence of stop-word removal and stemming are small. However, it is suggested to apply stop-word removal and stemming in order to reduce the dimensionality of feature space and promote the efficiency of the document categorization system.

Toman *et al.* [5] examined the impacts of normalization (stemming or lemmatization) and stop-word removal on English and Czech datasets. It is concluded that stop-word removal improved the classification accuracy in most cases. On the other hand, the impact of word normalization (stemming and lemmatization) was negative. It is suggested that applying stop-word removal and ignoring word normalization can be the best choice for document categorization.

Furthermore, the impact of preprocessing tasks including stop-word removal, and stemming on the English language are studied on trimmed versions of Reuters 21,578, Newsgroups and Springer by Pomikálek *et al.* [6]. It is concluded that using stemming and stop-word removal has very little impact on the overall classification results.

Uysal *et al.* [7] examined the impact of preprocessing tasks such as tokenization, stop-word removal, lowercase conversion, and stemming on the performance of the document categorization system. In their study, all possible combinations of preprocessing tasks are evaluated on two different domains, namely e-mail and news, and in two different languages, namely Turkish and English. Their experiments showed that the impact of stemming, tokenization and lowercase conversion were overall positive on the classification accuracy. On the contrary, the impact of the stop word removal task was negative on the accuracy of classification system.

The use of stop-word removal, stemming on spam email filtering, are analyzed by Méndez *et al.* [8]. It is concluded that performance of SVM is surprisingly better without using stemming and stop-word removal. However, some stop-words are rare in spam messages, and they should not be removed from the feature list in spite of being semantically void.

Chirawichitchai *et al.* [9] introduced the Thai document categorization framework which focused on the comparison of several feature representation schemes, including Boolean, Term Frequency (TF), Term Frequency inverse Document Frequency (TFiDF), Term Frequency Collection (TFC), Length Term Collection (LTC), Entropy, and Term Frequency Relevance Frequency (TF-RF). Tokenization, stop word removal and stemming were used as preprocessing tasks and chi-square as feature selection method. Three classification techniques are used, namely, naive bayes (NB), Decision Tree (DT), and support vector machine (SVM). The authors claimed that using TF-RF weighting with SVM classifier yielded the best performance with the F-measure equaling 95.9%.

Mesleh [10,11] studied the impact of six commonly used feature selection techniques based on support vector machines (SVMs) on Arabic DC. Normalization and stop word removal techniques were used as preprocessing tasks in his experiments. He claimed that experiments proved that the stemming technique is not always effective for Arabic document categorization. His experiments

showed that chi-square, the Ng-Goh-Low (NGL) coefficient, and the Galavotti-Sebastiani-Simi (GSS) coefficient significantly outperformed the other techniques with SVMs.

Al-Shargabi *et al.* [12] studied the impact of stop word removal on Arabic document categorization by using different algorithms, and concluded that the SVM with sequential minimal optimization achieved the highest accuracy and lowest error rate.

Al-Shammari and Lin [13] introduced a novel stemmer called the Educated Text Stemmer (ETS) for stemming Arabic documentation. The ETS stemmer is evaluated by comparison with output from human generated stemming and the stemming weight technique (Khoja stemmer). The authors concluded that the ETS stemmer is more efficient than the Khoja stemmer. The study also showed that disregarding Arabic stop words can be highly effective and provided a significant improvement in processing Arabic documents.

Kanan [14] developed a new Arabic light stemmer called P-stemmer, a modified version of one of Larkey's light stemmers. He showed that his approach to stemming significantly enhanced the results for Arabic document categorization, when using naive Bayes (NB), SVM, and random forest classifiers. His experiments showed that SVM performed better than the other two classifiers.

Duwairi *et al.* [15] studied the impact of stemming on Arabic document categorization. In their study, two stemming approaches, namely, light stemming and word clusters, were used to investigate the impact of stemming. They reported that the light stemming approach improved the accuracy of the classifier more than the other approach.

Khorsheed and Al-Thubaity [16] investigated classification techniques with a large and diverse dataset. These techniques include a wide range of classification algorithms, term selection methods, and representation schemes. For preprocessing tasks, normalization and stop word removal techniques were used. For term selection, their best average result was achieved using the GSS method with term frequency (TF) as the base for calculations. For term weighting functions, their study concluded that length term collection (LTC) was the best performer, followed by Boolean and term frequency collection (TFC). Their experiments also showed that the SVM classifier outperformed the other algorithms.

Ababneh *et al.* [17] discussed different variations of vector space model (VSM) to classify Arabic documents using the k-nearest neighbour (KNN) algorithm; these variations are Cosine coefficient, Dice coefficient, and Jacaard coefficient and use the inverse document frequency (IDF) term weighting method for comparison purposes. Normalization and stop word removal techniques were suggested as preprocessing tasks in their experiment. Their experimental results showed that the Cosine coefficient outperformed Dice and Jaccard coefficients.

Zaki *et al.* [18] developed a hybrid system for Arabic document categorization based on the semantic vicinity of terms and the use of a radial basis modeling. Normalization, stop word removal, and stemming techniques were used as preprocessing tasks. They adopted the hybridization of *N-gram* + *TFiDF* statistical measures to calculate similarity between words. By comparing the obtained results, they found that the use of radial basis functions improved the performance of the system.

Most of the previous studies on Arabic DC have proposed the use of preprocessing tasks to reduce the dimensionality of feature vectors without comprehensively examining their contribution in promoting the effectiveness of the DC system, which makes this study a unique one that addressed the impact of preprocessing tasks on the performance of Arabic classification systems.

To clarify the differences of the present work from the previous studies, the investigated preprocessing tasks and experimental settings are comparatively presented in Table 1. In this table, normalization, stop word removal, and stemming are abbreviated as NR, SR, and ST, respectively. The experimental settings include feature selection (FS in the table), dataset language, and classification algorithms (CA in the table).

Table 1. Comparison of the characteristics of this study with those of the previous studies.

Study	NR	SR	ST	FS	Language	CA
Uysal *et al.* [7]	-	√	√	√	Turkish and English	SVM
Pomikálek *et al.* [6]	-	√	√	√	English	KNN, NB, SVM, NN, *etc.*
Méndez *et al.* [8]	-	√	√	√	English	NB, SVM, Adaboost, *etc.*
Song *et al.* [4]	√	√	√	√	English	SVM
Toman *et al.* [5]	√	√	√	-	English and Czech	NB
Chirawichitchai *et al.* [9]	-	√	√	√	Thai	NB, DT, SVM
Mesleh [10,11]	√	√	-	√	Arabic	SVM
Duwairi *et al.* [15]	-	√	√	-	Arabic	KNN
Kanan [14]	-	√	√	-	Arabic	SVM, NB, RF
Zaki *et al.* [18]	√	√	√	-	Arabic	KNN
Al-Shargabi *et al.* [12]	-	√	-	-	Arabic	NB, SVM, J48
Khorsheed *et al.* [16]	-	√	-	√	Arabic	KNN, NB, SVM, *etc.*
Ababneh *et al.* [17]	√	√	-	-	Arabic	KNN
The proposed study	√	√	√	√	**Arabic**	**NB, KNN, SVM**

3. Overview of Arabic Language Structure

The Arabic language is a semantic language with a complicated morphology, which is significantly different from the most popular languages, such as English, Spanish, French, and Chinese. The Arabic language is a native language of the Arab states and the secondary language in a number of other countries [19]. More than 422 million people are able to speak Arabic, which makes this language the fifth most spoken language in the world, according to [14]. The alphabet of the Arabic language consists of 28 letters:

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

The Arabic language is classified into three forms: Classical Arabic (CA), Colloquial Arabic Dialects (CAD), and Modern Standard Arabic (MSA). CA is fully vowelized and includes classical historical liturgical text and old literature texts. CAD includes predominantly spoken vernaculars, and each Arab country has its dialect. MSA is the official language and includes news, media, and official documents [3]. The direction of writing in the Arabic language is from right to left.

The Arabic language has two genders, feminine (مؤنث) and masculine (مذكر); three numbers, singular (مفرد), dual (مثنى), and plural (جمع); and three grammatical cases, nominative (الرفع), accusative (النصب), and genitive (الجر). In general, Arabic words are categorized as particles (ادوات), nouns (اسماء), or verbs (افعال). Nouns in Arabic including adjectives (صفات) and adverbs (ظروف) and can be derived from other nouns, verbs, or particles. Nouns in the Arabic language cover proper nouns (such as people, places, things, ideas, day and month names, *etc.*). A noun has the nominative case when it is the subject (فاعل); accusative when it is the object of a verb (مفعول) and the genitive when it is the object of a preposition (مجرور بحرف جر) [20]. Verbs in Arabic are divided into perfect (صيغة الفعل التام), imperfect (صيغة الف الناقص) and imperative (صيغة الامر). Arabic particle category includes pronouns(الضمائر), adjectives(الصفات), adverbs(الاحوال), conjunctions(العطف), prepositions (حروف الجر), interjections (صي التعجب) and interrogatives (علامات الاستفهام) [21].

Moreover, diacritics are used in the Arabic language, which are symbols placed above or below the letters to add distinct pronunciation, grammatical formulation, and sometimes another meaning to the whole word. Arabic diacritics include hamza (ء), shada (ّ), dama (ُ), fathah (َ), kasra (ِ), sukon (ْ), double dama (ٌ), double fathah (ً), double kasra (ٍ) [22]. For instance, Table 2 presents different pronunciations of the letter (Sad) (ص):

Table 2. Presents different pronunciations of the letter (Sad) (ص).

صْ	صّ	صِ	صً	صَّ	صُّ	صّ	صِ	صَ	صُ
/s/	/sun/	/sin/	/san/	/ssa/	/ssu/	/ss/	/si/	/sa/	/su/

Arabic is a challenging language in comparison with other languages such as English for a number of reasons:

- The Arabic language has a rich and complex morphology in comparison with English. Its richness is attributed to the fact that one root can generate several hundreds of words having different meanings. Table 3 presents different morphological forms of root study (درس).

Table 3. Different morphological forms of word study (درس).

Word	Tense	Pluralities	Meaning	Gender
درس	Past	Single	He studied	Masculine
درست	Past	Single	She studied	Feminine
يدرس	Present	Single	He studies	Masculine
تدرس	Present	Single	She studied	Feminine
درسا	Past	Dual	They studied	Masculine
درستا	Past	Dual	They studied	Feminine
يدرسان	Present	Dual	They study	Masculine
تدرسان	Present	Dual	They study	Feminine
يدرسا	Present	Dual	They study	Masculine
تدرسا	Present	Dual	They study	Feminine
درسوا	Past	Plural	They studied	Masculine
درسن	Past	Plural	They studied	Feminine
يدرسوا	Present	Plural	They study	Masculine
تدرسن	Present	Plural	They study	Feminine
سيدرس	Future	Single	They will study	Masculine
ستدرس	Future	Single	They will study	Feminine
سيدرسا	Future	Dual	They will study	Masculine
ستدرسا	Future	Dual	They will study	Feminine
سيدرسون	Future	Plural	They will study	Masculine
ستدرسون	Future	Plural	They will study	Feminine

- 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. For example, in English, the word write is the root of word writer. In Arabic, the word writer (كاتب) is derived from the root write (كتب) by adding the letter Alef (ا) inside the root. In these cases, it is difficult to distinguish between infix letters and the root letters.
- Some Arabic words have different meanings based on their appearance in the context. Especially when diacritics are not used, the proper meaning of the Arabic word can be determined based on the context. For instance, the word (علم) could be Science (عِلْم), Teach (عَلَّمْ) or Flag (عَلَمْ) depending on the diacritics [23].
- Another challenge of automatic Arabic text processing is that proper nouns in Arabic do not start with a capital letter as in English, and Arabic letters do not have lower and upper case, which makes identifying proper names, acronyms, and abbreviations difficult.
- There are several free benchmarking English datasets used for document categorization, such as 20 Newsgroup, which contains around 20,000 documents distributed almost evenly into 20 classes; Reuters 21,578, which contains 21,578 documents belonging to 17 classes; and RCV1 (Reuters Corpus Volume 1), which contains 806,791 documents classified into four main classes.

Unfortunately, there is no free benchmarking dataset for Arabic document classification. In this work, to overcome this issue, we have used an in-house dataset collected from several published papers for Arabic document classification and from scanning the well-known and reputable Arabic websites.

- In the Arabic language, the problem of synonyms and broken plural forms are widespread. Examples of synonyms in Arabic are (تقدم, تعال, أقبل, هلم) which means (Come), and (منزل, دار, بيت, سكن) which means (house). The problem of broken plural forms occurs when some irregular nouns in the Arabic language in plural takes another morphological form different from its initial form in singular.
- In the Arabic language, one word may have more than lexical category (noun, verb, adjective, *etc.*) in different contexts such as (wellspring, "عين الماء"), (Eye, "عين الانسان"), (was appointed, "عين وزيرا للتجارة").
- In addition to the different forms of the Arabic word that result from the derivational process, there are some words lack authentic Arabic roots like Arabized words which are translated from other languages, such as (programs, "برامج"), (geography, "جغرافية"), (internet, "الإنترنت"), *etc.* or names, places such as (countries, "البلدان"), (cities, "المدن"), (rivers, "الانهار"), (mountains, "الجبال"), (deserts," الصحارى"), *etc*.

As a result, the difficulty of the Arabic language processing in Arabic document categorization is associated with the complex nature of the Arabic language, which has a very rich and complicated morphology. Therefore, the Arabic language needs a set of preprocessing routines to be suitable for classification.

4. Arabic Document Categorization Framework

An Arabic Document Categorization Framework usually consists of three main stages: the preprocessing stage, the document modeling stage, and document classification. The preprocessing stage involves document conversion, tokenization, stop word removal, normalization, and stemming. The document modeling stage includes vector space model construction, term selection, and term weighting. The document classification stage covers classification model construction and classification model evaluation. These phases will be described in details in the following subsections.

4.1. Data Preprocessing

Document preprocessing, which is the first step in DC, converts the Arabic documents to a form that is suitable for classification tasks. These preprocessing tasks include a few linguistic tools such as tokenization, normalization, stop word removal, and stemming. These linguistic tools are used to reduce the ambiguity of words to increase the accuracy and effectiveness of the classification system.

4.1.1. Text Tokenization

Tokenization is a method for dividing texts into tokens. Words are often separated from each other by blanks (white space, semicolons, commas, quotes, and periods). These tokens could be individual words (noun, verb, pronoun, article, conjunction, preposition, punctuation, numbers, and alphanumeric) that are converted without understanding their meanings or relationships. The list of tokens becomes input for further processing.

4.1.2. Stop Word Removal

Stop word removal involves elimination of insignificant words, such as so لذا, for لاجل, and with مع, which appear in the sentences and do not have any meaning or indications about the content. Other examples of these insignificant words are articles, conjunctions, pronouns (such as he هو, she هي, and they هم), prepositions (such as from من, to الى, in في, and about حول), demonstratives, (such as this هذا, these هولاء, and there اولئك), and interrogatives (such as where اين, when متى, and whom

لمن). Moreover, Arabic circumstantial nouns indicating time and place (such as after بعد, above فوق, and beside بجانب), signal words (such as first اولا, second ثانيا, and third ثالثا) as well as numbers and symbols (such as @, #, &, %, and *) are considered insignificant and can be eliminated. A list of 896 words was prepared to be eliminated from all the documents.

4.1.3. Word Normalization

Normalization aims to normalize certain letters that have different forms in the same word to one form. For example, the normalization of "ء" (hamza), "آ" (aleph mad), "أ" (aleph with hamza on top), "ؤ" (hamza on waw), "إ" (alef with hamza at the bottom), and "ئ" (hamza on ya) to "ا" (alef). Another example is the normalization of the letter "ى" to "ي" and the letter "ة" to "ه". We also remove the diacritics such as {ْ, ٍ, ٌ, ً, ُ, ِ, ّ, َ} because these diacritics are not used in extracting the Arabic roots and not useful in the classification. Finally, we duplicate the letters that include the symbols ""ّ الشدة, because these letters are used to extract the Arabic roots and removing them affects the meaning of the words.

4.1.4. Stemming

Stemming can be defined as the process of removing all affixes (such as prefixes, infixes, and suffixes) from words. Stemming reduces different forms of the word that reflect the same meaning in the feature space to single form (its root or stem). 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"). All these words share the same abstract meaning of action or movement. Using the stemming techniques in the DC makes the processes less dependent on particular forms of words and reduces the potential size of features, which, in turn, improve the performance of the classifier. For the Arabic language, the most common stemming approaches are the root-based stemming approach and the light stemming approach.

The root-based stemmer uses morphological patterns to extract the root. Several root-based stemmer algorithms have been developed for the Arabic language. For example, the author in [24] has developed an algorithm that starts by removing suffixes, prefixes, and infixes. Next, this algorithm matches the remaining word against a list of patterns of the same length to extract the root. The extracted root is then matched against a list of known "valid" roots. However, the root-based stemming technique increases the word ambiguity because several words have different meanings but have stems from the same root. Hence, these words will always be stemmed to this root, which, in turn, leads to a poor performance. For example, the words مقصود, قاصد, الاقتصادية have different meanings, but they stemmed to one root, that is, "قصد", this root is far abstract from the stem and will lead to a very poor performance of the system [25].

The light stemmer approach is the process of stripping off the most frequent suffixes and prefixes depending on a predefined list of prefixes and suffixes. The light stemmer approach aims not to extract the root of a given Arabic word; hence, this approach does not deal with infixes or does not recognize patterns. Several light stemmers have been proposed for the Arabic language such as [26,27]. However, light stemming has difficulty in stripping off prefixes or suffixes in Arabic. In some cases, removal of a fixed set of prefixes and suffixes without checking if the remaining word is a stem can lead to unexpected results, especially when distinguishing extra letters from root letters is difficult.

According to [26], which compared the performance of light stemmer and root base stemmer and concluded that the light stemmer significantly outperforms the root base stemmer, we adopted the light stemmer [27] as the preprocessing task for Arabic document categorization in this study.

Preprocessing Algorithm:

Step 1: Remove non-Arabic letters such as punctuation, symbols, and numbers including {., :, /, !, §,&,_ , [, (, ,—, |,—,ˆ,),], }=,+, $, ∗, . . .}.
Step 2: Remove all non-Arabic words and any Arabic word that contains special characters.
Step 3: Remove all Arabic stop words.
Step 4: Remove the diacritics of the words such as {ً, ٌ, ٍ, َ, ُ, ِ, ّ}, except " ّ " الشدة.
Step 5: Duplicate all the letters that contain the symbols " ّ "الشدة.
Step 6: Normalize certain letters in the word, which have many forms to one form. For example, the normalization of "ء" (hamza), "آ" (aleph mad), "أ" (aleph with hamza on top), "ؤ" (hamza on waw), "إ" (alef with hamza at the bottom), and "ئ" (hamza on ya) to "ا" (alef). Another example is the normalization of the letter "ى" to "ي" and the letter "ة" to "ه" when these letters appear at the end of a word.
Step 7: Remove words with length less than three letters because these words are considered insignificant and will not affect the classification accuracy.
Step 8: If the word length = 3, then return the word without stemming because attempting to shorten words with more than three letters can lead to ambiguous stems.
Step 9: Apply the light stemming algorithm for the Arabic words list to obtain an Arabic stemmed word list.

4.2. Document Modeling

This process is also called document indexing and consists of the following two phases:

4.2.1. Vector Space Model Construction

In VSM, a document is represented as a vector. Each dimension corresponds to a separate word. If a word occurs in the document, then its weighting value in the vector is non-zero. Several different methods have been used to calculate terms' weights. One of the most common methods is *TFiDF*. The *TF* in the given document measures the relevance of the word within a document, whereas the *DF* measures the global relevance of the word within a collection of documents [28].

In particular, considering a collection of documents D containing N documents, such that $D = \{d_0, \ldots \ldots, d_{n-1}\}$ each document d_i that contains a collection of terms t will be represented as vectors in VSM as follows:

$$d_{ij} = \left(t_{1j}, t_{2j}, \ldots\ldots, t_{ij}\right), \; j = 1, \ldots, m, \tag{1}$$

where m is the number of distinct words in the document *di*.

The *TFiDF* method uses the *TF* and *DF* to compute the weight of a word in a document by using Equations (2) and (3):

$$IDF\,(t) = \log\left(\frac{N}{DF\,(d,t)}\right) \tag{2}$$

$$TFiDF\,(t,d) = TF\,(t,d) * IDF\,(d,t) \tag{3}$$

where *TF* (*t*,*d*) is the number of times term *t* occurs in document *d*, *DF* (*d*,*t*) is the number of documents that contain term *t* , and *N* is the total of all documents in the training set.

4.2.2. Feature Selection

In VSM, a large number of terms (dimensions) are irrelevant to the classification task and can be removed without affecting the classification accuracy. The mechanism that removes the irrelevant feature is called feature selection. Feature selection is the process of selecting the most representative subset that contains the most relevant terms for each category in the training set based on a few criteria and of using this subset as features in DC [29].

Feature selection aims to choose the most relevant words that distinguish (one topic or class from other classes) between classes in the dataset. Several feature selection methods have been introduced for DC. The most frequently used methods mainly include DF threshold [30], information gain [31], and chi-square testing (x^2) [11]. All these methods organize the features according to their importance or relevance to the category. The top ranking features from each category are then chosen and represented to the classification algorithm.

In our paper, we suggested chi-square testing (x^2), which is defined as a well-known discrete data hypothesis testing method from statistics; this technique evaluates the correlation between two variables and determines whether these variables are independent or correlated [32]. x^2 value for each term t in a category c can be defined by using Equations (4) and (5) [33].

$$x^2\left(t_k, c_i\right) = \frac{|Tr| \cdot \left[P\left(t_k, c_i\right) * P\left(\overline{t_k}, \overline{c_i}\right) - P\left(t_k, \overline{c_i}\right) * P\left(\overline{t_k}, c_i\right)\right]^2}{P\left(t_k\right) * P\left(\overline{t_k}\right) * P\left(c_i\right) * P\left(\overline{c_i}\right)} \tag{4}$$

and is estimated using

$$x^2\left(t, c\right) = \frac{N * (AD - CB)^2}{(A + C) * (B + D) * (A + B) * (C + D)} \tag{5}$$

where A is the number of documents of category c containing the term t; B is the number of documents of other category (not c) containing t; C is the number of documents of category c not containing the term t; D is the number of documents of other category not containing t; and N is the total number of documents. The chi-square statistics show the relevance of each term to the category. We compute chi-square values for each term in its respective category. Finally, highly relevant terms are chosen.

4.2.3. Document Categorization

In this step, the most popular statistical classification and machine learning techniques such as NB [34,35], KNN [36,37], and SVM [11,38] are suggested to study the influence of preprocessing on the Arabic DC system. The VSM that contains the selected features and their corresponding weights in each document of the training dataset are used to train the classification model. The classification model obtained from the training process will be evaluated by means of testing data.

5. Experiment and Results

5.1. Arabic Data Collection

Unfortunately, there is no free benchmarking dataset for Arabic document categorization. For most Arabic document categorization research, authors collect their own datasets, mostly from online news sites. To test the effect of preprocessing on Arabic DC and to evaluate the effectiveness of the proposed preprocessing algorithm, we have used an in-house corpus collected from the dataset used in several published papers for Arabic DC and gathered from scanning well-known and reputable Arabic news websites.

The collected corpus contains 32,620 documents divided into 10 categories of News, Economy, Health, History, Sport, Religion, Social, Nutriment, Law, and Technology that vary in length and number of documents. In this Arabic corpus, each document must be assigned to one of the corresponding category directories. The statistics of the corpus are shown in Table 4.

Figure 1 shows the distribution of documents in each category in our corpus. The largest category contains around 6860 documents, whereas the smallest category contains nearly 530 documents.

Table 4. Statistics of the documents in the corpus.

Category Name	Number of Documents
News	6860
Economy	4780
Health	2590
History	3230
Sport	3950
Religion	3470
Social	3600
Nutriment	2370
Law	1240
Technology	530
Total	**32,620**

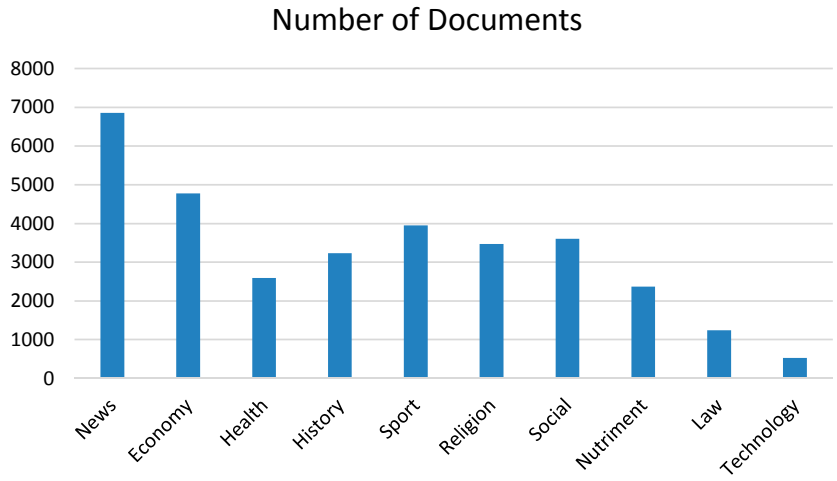

Figure 1. Distribution of documents in each category.

5.2. Experimental Configuration and Performance Measure

In this paper, the documents in each category were first preprocessed by converting them to UTF8 encoding. For stop word removal, we used a file containing 896 stop words. This list includes distinct stop words and their possible variations.

Feature selection was performed using chi-square statistics. The number of features for building the vectors representing the documents included 50, 100, 500, 1000, and 2000. In doing so, the effect of preprocessing task can be comparatively observed within a wide range of feature size. Feature vectors were built using the function *TFiDF* as described by Equations (2) and (3).

In this study, SVM, NB, and KNN techniques were applied to observe the preprocessing effect on improving classification accuracy. Cross-validation was used for all classification experiments, which partitioned the complete collection of documents into 10 mutually exclusive subsets called folds. Each fold contains 3262 of documents. One of the subsets is used as the test set, whereas the rest of the subsets are used as training sets.

We have developed our application using JAVA Programming for implement preprocessing tasks on the dataset. We used RapidMiner 6.0 software (HQ in Boston MA USA) to build the classification model that will be evaluated by means of the testing data. RapidMiner is an open-source software which provides an implementation for all classification algorithms used in our experiments.

The evaluation of the performance for classification model to classify documents into the correct category is conducted by using several mathematic rules such as recall, precision, and F-measure, which are defined as follows:

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

where TP is the number of documents that are correctly assigned to the category, TN is the number of documents that are correctly assigned to the negative category, FP is the number of documents a system incorrectly assigned to the category, and FN are the number of documents that belonged to the category but are not assigned to the category. The success measure, namely, micro-F1 score, a well-known F1 measure, is selected for this study, which is calculated as follows:

$$\text{Micro-F1} = \frac{2 * Precision \times Recall}{Precision + Recall} \tag{8}$$

5.3. Experimental Results and Analysis

First, we ran three different experiments to study the effect of each of the preprocessing tasks. The three experiments were conducted using four different representations of the same dataset. The original dataset without any preprocessing task was used in the first experiment, and this experiment was presented as the baseline. In the second experiment, normalization was used as the preprocessing task of documents. The stop word removal was used in the third experiment. In the fourth and last experiment, we used light stemmer [27]. The results of the experiments for preprocessing task when applied individually on the three classification algorithms are illustrated in Table 5 and Figure 2.

Table 5. F1 Measure scores for preprocessing tasks when applied individually.

Classification Algorithm	Feature Size	Baseline (BS)	Normalization (NR)	Stop Word Removal (SR)	Light Stemmer (LS)
NB	50	0.65	0.6678	0.6511	0.7311
	100	0.7737	0.7763	0.7881	0.8107
	500	0.8841	0.8852	0.8896	0.8807
	1000	0.9089	0.9093	0.9130	0.8889
	2000	0.9252	0.9287	0.9319	0.9022
KNN	50	0.6211	0.6541	0.6319	0.7681
	100	0.7319	0.7552	0.7411	0.8196
	500	0.7611	0.7807	0.7741	0.8507
	1000	0.7600	0.7674	0.7230	0.8378
	2000	0.6359	0.6663	0.6085	0.8119
SVM	50	0.7074	0.7219	0.7141	0.8019
	100	0.7781	0.8063	0.7915	0.8767
	500	0.9311	0.9330	0.9348	0.9507
	1000	0.9559	0.9530	0.9585	0.9630
	2000	0.9648	0.9619	0.9657	0.9663

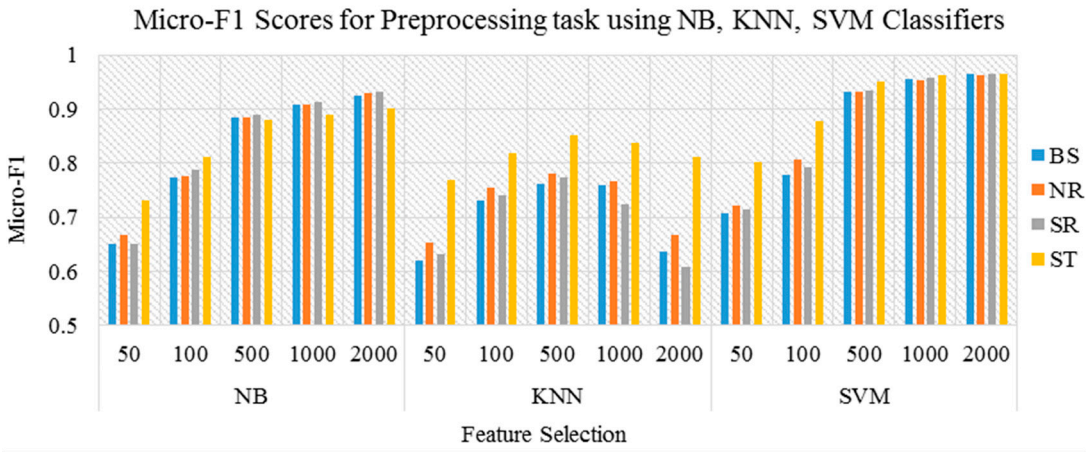

Figure 2. Experimental results preprocessing task using NB, KNN, SVM classifiers.

According to the proposed method, applying normalization, stop word removal, and light stemming has a positive impact on classification accuracy in general. As shown in Table 3, applying light stemming alone has a dominant impact and provided a significant improvement in classification accuracy with SVM and KNN classifiers but has a negative impact when used with NB classifiers and when feature size increases. Similarly, stop word removal provided a significant improvement in classification accuracy with NB and SVM classifiers, but has a negative impact when used with KNN classifiers and when feature size increases. The latter conclusion is surprising because most studies on DC in literature apply stop words directly by assuming them irrelevant.

The results showed that the normalization helped to improve the performance and provided a slight improvement in the classification accuracy. Therefore, normalization should be applied without depending on feature size and classification algorithms. Given that normalization helps in grouping the words that contain the same meaning, a smaller amount of features with further discrimination are achieved. However, stop word removal and stemming status may change depending on the feature size and the classification algorithms.

Combining Preprocessing Tasks

In the following experiments, we studied the effect of combining preprocessing tasks on the accuracy of Arabic document categorization. All possible combinations of the preprocessing tasks listed in Table 6 are considered during the experiments to reveal all possible interactions between the preprocessing tasks. Stop word removal, Normalization, Light stemming (abbreviated in Table 6 as NR, SR, and LS, respectively) are either 1 or 0 which refer to "apply" or "not-apply".

Table 6. Combinations of preprocessing tasks.

No.	Preprocessing Tasks Combinations
1	Normalization (NR):1 ǀ Stop-word removal (SR):1 ǀ Stemming (LS):0
2	Normalization (NR):1 ǀ Stop-word removal (SR):0 ǀ Stemming (LS):1
3	Normalization (NR):0 ǀ Stop-word removal (SR):1 ǀ Stemming (LS):1
4	Normalization (NR):1 ǀ Stop-word removal (SR):1 ǀ Stemming (LS):1

In these experiments, three classification algorithms, namely, SVMs, NB, and KNN, were applied to find the most suitable combination of preprocessing tasks and classification approaches to deal with Arabic documents. The results of the experiments on the three classification algorithms are illustrated in Table 7 and Figure 3. The minimum and maximum Micro-F1 and the corresponding combinations of preprocessing tasks at different feature sizes are also included.

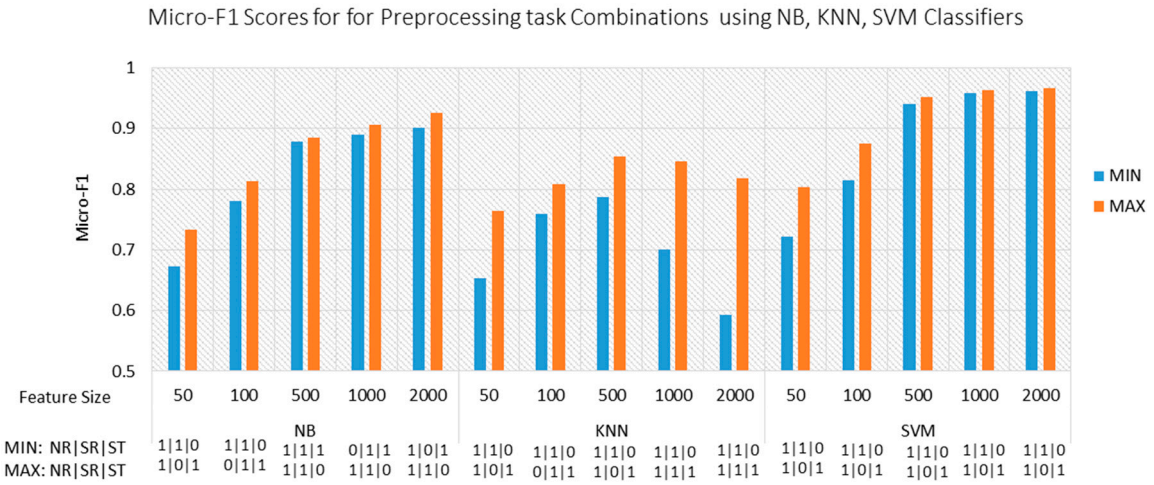

Figure 3. Experimental results and the corresponding combinations of preprocessing tasks for NB, KNN, SVM Classifiers.

Table 7. Experimental results and the corresponding combinations of preprocessing tasks.

Classifier	Feature Size	Max *vs.* Min	Micro-F1	Preprocessing Combination		
				NR	SR	LS
NB	50	Min	0.6730	1	1	0
		Max	0.7326	1	0	1
	100	Min	0.7804	1	1	0
		Max	0.8141	0	1	1
	500	Min	0.8793	1	1	1
		Max	0.8859	1	1	0
	1000	Min	0.8896	0	1	1
		Max	0.9063	1	1	0
	2000	Min	0.9022	1	0	1
		Max	0.9252	1	1	0
KNN	50	Min	0.6526	1	1	0
		Max	0.7648	1	0	1
	100	Min	0.7593	1	1	0
		Max	0.8278	0	1	1
	500	Min	0.7874	1	1	0
		Max	0.8537	1	0	1
	1000	Min	0.7007	1	1	0
		Max	0.8467	1	1	1
	2000	Min	0.5937	1	1	0
		Max	0.8185	1	1	1
SVM	50	Min	0.7222	1	1	0
		Max	0.8030	1	0	1
	100	Min	0.8152	1	1	0
		Max	0.8756	1	0	1
	500	Min	0.9404	1	1	0
		Max	0.9522	1	0	1
	1000	Min	0.9589	1	1	0
		Max	0.9633	1	0	1
	2000	Min	0.9626	1	1	0
		Max	0.9674	1	0	1

Considering the three classification algorithms, the difference between the maximum and minimum Micro-F1 for all preprocessing combinations at each feature size ranged from 0.0044 to 0.2248. In particular, the difference was between 0.0066 and 0.0596 in NB classifier, between 0.0663 and

0.2248 in KNN classifier, and between 0.0044 and 0.0808 in SVM classifier. The amount of variation in accuracies proves that the appropriate combinations of preprocessing tasks depending on the classification algorithm and feature size may significantly improve the accuracy. On the contrary, inappropriate combinations of preprocessing tasks may significantly reduce the accuracy.

The impact of preprocessing on Arabic document categorization was also statistically analyzed using a one-way analysis of variance (ANOVA) over the maximum and minimum Micro-F1 at each feature size with an alpha value = 0.05. p-values were obtained as 0.001258, 0.004316, and 0.203711 for NB, SVM, and KNN, respectively. The results have supported our hypothesis that performance differences are statistically significant with a significance level of 0.05.

We also compared the preprocessing combinations, which provided the maximum accuracy for each classification technique, and the preprocessing combinations, which provided the maximum accuracy at minimum feature size for each classification technique. As shown in Table 8, normalization and stemming should be applied to achieve either maximum accuracy or minimum feature size with the maximum accuracy for KNN and SVM classifiers regardless of the feature size. On the contrary, the statuses of preprocessing tasks were opposite for the NB classifier.

Table 8. Preprocessing tasks (maximum accuracy *vs.* minimum feature size).

Classification Technique	Preprocessing Tasks (Max. Accuracy)			Preprocessing Tasks (Min. Feature Size)		
	Feature Size	Max. Micro-F1	Preprocessing Tasks	Feature Size	Max. Micro-F1	Preprocessing Tasks
Naïve Bayes	2000	0.9319	NR: 0 ǀ SR: 1 ǀ LS: 0	50	0.7326	NR: 1 ǀ SR: 0 ǀ LS: 1
KNN	500	0.8537	NR: 1 ǀ SR: 0 ǀ LS: 1	50	0.7648	NR: 1 ǀ SR: 0 ǀ LS: 1
SVM	2000	0.9674	NR: 1 ǀ SR: 0 ǀ LS: 1	50	0.8030	NR: 1 ǀ SR: 0 ǀ LS: 1

Furthermore, the findings of the proposed study in terms of contribution to the accuracy were compared against the previous works mentioned in the related works section. The comparison is presented in Table 9, where "+" and "−" signs, respectively, represent positive and negative effects, and "N/F" indicate that the corresponding analysis is not found.

Table 9. The comparison of the influence of the preprocessing task in various studies and proposed study.

Study	NR	SR	LS
Uysal *et al.* [7]	N/F	-	+ (often)
Pomikálek *et al.* [6]	N/F	+	+
Méndez *et al.* [8]	N/F	-	-
Chirawichitchai *et al.* [9]	N/F	N/F	N/F
Song *et al.* [4]	N/F	+	+
Toman *et al.* [5]	N/F	+	-
Mesleh [10,11]	N/F	N/F	-
Duwairi *et al.* [15]	N/F	N/F	+
Kanan [14] 2015	N/F	N/F	+
Zaki *et al.* [18]	N/F	N/F	+
Al-Shargabi *et al.* [12]	N/F	+	N/F
Khorsheed *et al.* [16]	N/F	N/F	N/F
Ababneh *et al.* [17]	N/F	N/F	-
Proposed Study	**+**	**- (often)**	**+ (often)**

In general, the results clearly showed the superiority of the SVM over the NB and KNN algorithms for all experiments, especially when the feature size increases. Similarly, Hmeidi [39] compared the performance of KNN and SVM algorithms for Arabic document categorization and concluded that SVM outperformed KNN and showed a better micro average F1 and prediction time. SVM algorithm is superior because this technique has the ability to handle high dimensional data. Furthermore, this

algorithm can easily test the effect of the number of features on classification accuracy, which ensures robustness to different dataset and preprocessing tasks. By contrast, the KNN performance degrades when the feature size increases. This assumption is due to certain features not clearly belonging to both categories taking part in the classification process.

6. Conclusions

In this study, we investigated the impact of preprocessing tasks on the accuracy of Arabic document categorization by using three different classification algorithms. Three preprocessing tasks were used, namely, normalization, stop word removal, and stemming.

The examination was conducted using all possible combinations of the preprocessing tasks considering different aspects such as accuracy and dimension reduction. The results obtained from the experiments reveal that appropriate combinations of preprocessing tasks showed a significant improvement in classification accuracy, whereas inappropriate combinations may degrade the classification accuracy.

Findings of this study showed that the normalization task helped to improve the performance and provided a slight improvement in the classification accuracy regardless of feature size and classification algorithms. Stop word removal has negative impact when combined with other preprocessing tasks in most cases. Stemming techniques provided a significant improvement in classification accuracy when applied alone or combined with other preprocessing tasks in most cases. The results also clearly showed the superiority of the SVM over the NB and KNN algorithms for all experiments.

As a result, the significant impact of preprocessing techniques on the classification accuracy is as important as the impact of feature extraction, feature selection, and classification techniques, especially with a highly inflected language such as the Arabic language. Therefore, for document categorization problems using any classification technique and in any feature size, researchers should consider investigating all possible combinations of the preprocessing tasks instead of applying them simultaneously or applying them individually. Otherwise, classification performance may significantly differ.

Future research will consider introducing a new hybrid algorithm for the Arabic stemming technique, which attempts to overcome the weaknesses of state-of-the-art stemming techniques in order to improve the accuracy of DC system.

Author Contributions: Abdullah Ayedh proposed the idea of this research work, analyzed the experimental results, and wrote the manuscript. Guanzheng Tan led and facilitated the further discussion and the revision. Hamdi Rajeh and Khaled Alwesabi have been involved in discussing and helping to shape the idea, and drafting the manuscript. All authors read and approved the final manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Al-Kabi, M.; Al-Shawakfa, E.; Alsmadi, I. The Effect of Stemming on Arabic Text. Classification: An Empirical Study. *Inf. Retr. Methods Multidiscip. Appl.* **2013**. [CrossRef]
2. Joachims, T. *Text Categorization with Support Vector Machines: Learning with Many Relevant Features*; Springer: Berlin, Germany, 1998.
3. Nehar, A.; Ziadi, D.; Cherroun, H.; Guellouma, Y. An efficient stemming for arabic text classification. *Innov. Inf. Technol* **2012**. [CrossRef]
4. Song, F.; Liu, S.; Yang, J. A comparative study on text representation schemes in text categorization. *Pattern Anal. Appl.* **2005**, *8*, 199–209. [CrossRef]
5. Toman, M.; Tesar, R.; Jezek, K. Influence of word normalization on text classification. *Proc. InSciT* **2006**, *4*, 354–358.
6. Pomikálek, J.; Rehurek, R. The Influence of preprocessing parameters on text categorization. *Int. J. Appl. Sci. Eng. Technol.* **2007**, *1*, 430–434.

7. Uysal, A.K.; Gunal, S. The impact of preprocessing on text classification. *Inf. Proc. Manag.* **2014**, *50*, 104–112. [CrossRef]
8. Méndez, J.R.; Iglesias, E.L.; Fdez-Riverola, F.; Diaz, F.; Corchado, J.M. Tokenising, Stemming and Stopword Removal on Anti-Spam Filtering Domain. In *Current Topics in Artificial Intelligence*; Springer: Berlin, Germany, 2005; pp. 449–458.
9. Chirawichitchai, N.; Sa-nguansat, P.; Meesad, P. Developing an Effective Thai Document Categorization Framework Base on Term Relevance Frequency Weighting. In Proceedings of the 2010 8th International Conference on ICT, Bangkok, Thailand, 24–25 November 2010.
10. Moh'd Mesleh, A. Support Vector Machines Based Arabic Language Text Classification System: Feature Selection Comparative Study. In *Advances in Computer and Information Sciences and Engineering*; Springer: Berlin, Germany, 2008; pp. 11–16.
11. Moh'd A Mesleh, A. Chi square feature extraction based SVMs Arabic language text categorization system. *J. Comput. Sci.* **2007**, *3*, 430–435.
12. Al-Shargabi, B.; Olayah, F.; Romimah, W.A. An experimental study for the effect of stop words elimination for arabic text. classification algorithms. *Int. J. Inf. Technol. Web Eng.* **2011**, *6*, 68–75. [CrossRef]
13. Al-Shammari, E.T.; Lin, J. Towards an Error-Free Arabic Stemming. In Proceedings of the 2nd ACM Workshop on Improving Non English Web Searching, Napa Valley, CA, USA, 26–30 October 2008.
14. Kanan, T.; Fox, E.A. Automated Arabic Text. Classification with P-Stemmer, Machine Learning, and a Tailored News Article Taxonomy. *J. Assoc. Inf. Sci. Technol.* **2016**. [CrossRef]
15. Duwairi, R.; Al-Refai, M.N.; Khasawneh, N. Feature reduction techniques for Arabic text categorization. *J. Am. Soc. Inf. Sci. Technol.* **2009**, *60*, 2347–2352. [CrossRef]
16. Khorsheed, M.S.; Al-Thubaity, A.O. Comparative evaluation of text classification techniques using a large diverse Arabic dataset. *Lang. Resour. Eval.* **2013**, *47*, 513–538. [CrossRef]
17. Ababneh, J.; Almomani, O.; Hadi, W.; Al-omari, N.; Al-ibrahim, A. Vector space models to classify arabic text. *Int. J. Comput. Trends Technol.* **2014**, *7*, 219–223. [CrossRef]
18. Zaki, T.; Es-saady, Y.; Mammass, D.; Ennaji, A.; Nicolas, S. A Hybrid Method N-Grams-TFIDF with radial basis for indexing and classification of Arabic documents. *Int. J. Softw. Eng. Its Appl.* **2014**, *8*, 127–144.
19. Thabtah, F.; Gharaibeh, O.; Al-Zubaidy, R. Arabic text mining using rule based classification. *J. Inf. Knowl. Manag.* **2012**, *11*. [CrossRef]
20. Zrigui, M.; Ayadi, R.; Mars, M.; Maraoui, M. Arabic Text. Classification framework based on latent dirichlet allocation. *J. Comput. Inf. Technol.* **2012**, *20*, 125–140. [CrossRef]
21. Khoja, S. APT: Arabic Part-of-Speech Tagger. In Proceedings of the Student Workshop at NAACL, Pittsburghm, PA, USA, 2–7 June 2001; pp. 20–25.
22. Duwairi, R.M. Arabic Text. Categorization. *Int. Arab J. Inf. Technol.* **2007**, *4*, 125–132.
23. Nwesri, A.F.; Tahaghoghi, S.M.; Scholer, F. Capturing Out-of-Vocabulary Words in Arabic text. In Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006), Sydney, Australia, 22–23 July 2006; pp. 258–266.
24. Khoja, S.; Garside, R. Stemming Arabic Text. Computing Department, Lancaster University: Lancaster, UK, 1999; Available online: http://www.comp.lancs.ac.uk/computing/users/khoja/stemmer.ps (accessed on 14 April 2004).
25. Kanaan, G.; Al-Shalabi, R.; Ababneh, M.; Al-Nobani, A. Building an Effective Rule-Based Light Stemmer for Arabic Language to Inprove Search Effectiveness. In Proceedings of the 2008 International Conference on Innovations in Information Technology, Al Ain, Arab Emirates, 16–18 December 2008; pp. 312–316.
26. Aljlayl, M.; Frieder, O. On Arabic Search: Improving the Retrieval Effectiveness via a Light Stemming Approach. In Proceedings of the Eleventh International Conference on Information and Knowledge Management, McLean, VA, USA, 4–9 November 2002; pp. 340–347.
27. Larkey, L.S.; Ballesteros, L.; Connell, M.E. Light Stemming for Arabic Information Retrieval. In *Arabic Computational Morphology*; Springer: Berlin, Germany, 2007; pp. 221–243.
28. Salton, G.; Buckley, C. Term-weighting approaches in automatic text retrieval. *Inf. Proc. Manag.* **1988**, *24*, 513–523. [CrossRef]
29. Forman, G. An. Extensive empirical study of feature selection metrics for text classification. *J. Mach. Learn. Res.* **2003**, *3*, 1289–1305.

30. Zahran, B.M.; Kanaan, G. Text Feature Selection using Particle Swarm Optimization Algorithm. *World Appl. Sci. J.* **2009**, *7*, 69–74.
31. Ogura, H.; Amano, H.; Kondo, M. Feature selection with a measure of deviations from Poisson in text categorization. *Expert Syst. Appl.* **2009**, *36*, 6826–6832. [CrossRef]
32. Thabtah, F.; Eljinini, M.; Zamzeer, M.; Hadi, W. Naïve Bayesian Based on Chi Square to Categorize Arabic Data. In Proceedings of the 11th International Business Information Management Association Conference (IBIMA) Conference on Innovation and Knowledge Management in Twin Track Economies, Cairo, Egypt, 4–6 January 2009; pp. 4–6.
33. Sebastiani, F. Machine learning in automated text categorization. *ACM Comput. Surv.* **2002**, *34*, 1–47. [CrossRef]
34. El Kourdi, M.; Bensaid, A.; Rachidi, T.-E. Automatic Arabic document categorization based on the Naïve Bayes algorithm. In Proceedings of the Workshop on Computational Approaches to Arabic Script-Based Languages, Geneva, Switzerland, 28 Auguest 2004; pp. 51–58.
35. Al-Saleem, S. Associative classification to categorize Arabic data sets. *Int. J. Acm Jordan* **2010**, *1*, 118–127.
36. Syiam, M.M.; Fayed, Z.T.; Habib, M.B. An intelligent system for Arabic text categorization. *Int. J. Intell. Comput. Inf. Sci.* **2006**, *6*, 1–19.
37. Bawaneh, M.J.; Alkoffash, M.S.; Al Rabea, A. Arabic Text Classification Using K-NN and Naive Bayes. *J. Comput. Sci.* **2008**, *4*, 600–605. [CrossRef]
38. Alaa, E. A comparative study on arabic text classification. *Egypt. Comput. Sci. J.* **2008**, *2*. [CrossRef]
39. Hmeidi, I.; Hawashin, B.; El-Qawasmeh, E. Performance of KNN and SVM classifiers on full word Arabic articles. *Adv. Eng. Inform.* **2008**, *22*, 106–111. [CrossRef]