Enhancement of Arabic Text Classification Using Semantic Relations with Part of Speech Tagger

SUHAD A. YOUSIF
Mathematics and Computer Science Department
Faculty of Science
Beirut Arab University
LEBANON
Suhadmail75@Yahoo.Com

VENUS W. SAMAWI
Computer Information System Department
Amman Arab University
JORDAN
venus@aau.edu.jo

ISLAM ELKABANI*
Mathematics and Computer Science Department
Faculty of Science
Beirut Arab University
LEBANON
islam.kabani@bau.edu.lb

RACHED ZANTOUT
Electrical and Computer Engineering Department
Rafik Hariri University
LEBANON
zantoutrm@rhu.edu.lb

Abstract:
When it comes to Arabic text documents, Text Categorization (TC) becomes a challenge. TC is needed for clustering purposes in order to complete Text Mining (TC). Based on the nature of Arabic language, extracting roots or stems from the breakdown of multiple Arabic words and phrases is important task before applying TC. The results obtained by applying the proposed algorithm are compared with the results of three popular algorithms. These algorithms are Khoja stemmer, Light stemmer, and Root extractor. The performance of these three techniques are evaluated and compared based on the accuracy of Naive Bayesian classifier. The obtained result demonstrates that these techniques are not as promising as expected. Therefore, we decided to consider the position tagger and conceptual representation to answer the question, which approach enhances the Arabic TC performance? Arabic WordNet (AWN) is used as a lexical and semantic resource. The performance of new relation "Has- hyponym", suggested in this work, is compared with other already used relations like Synset, term+ Synset, all Synsets, and Bag of words representation to demonstrate its effectiveness. From the experimental results, it was found that the new suggested relation improved the Arabic text classification, at which the macro average F1 is raised to 0.75437 compared with the performance of the other approaches.

Key-Words: -Arabic Text classification, Stemmer, Part of Speech, Conceptual features, Semantic relations.

1. Introduction
The process entitled Text Categorization has a significant aim to classify a recent document into one or multiple categories. It is performed through the utilization of prearranged and already classified documents as training set, thus making it a supervised classification technique. Such a technique in Text Classification has become an important tool to process huge amount of data on the web [1,2].
Text preprocessing for Arabic documents is considered a challenging task especially in information retrieval, text mining, and natural language processing where the processing task include different stages as stop words removal and stemming. The reasons behind that is that Arabic Language is considered a Semitic complicated language compared to English language, which is a highly inflected language. Due to this complexity it needs a set of preprocessing procedures to be ready for manipulation [3]. In fact, text processing techniques might have a positive or negative impact on the accuracy of any text categorization, thus the enhancement of preprocessing stage will lead by necessity to the improvement of any text categorization.

Scientists and Researchers have developed many stemming algorithms for different languages including English, Malay, Latin, Indonesian, Swedish, Dutch, German, Italian French, Slovene, Turkish, Bangla, and Chinese [4]. Yet for Arabic Language, three main different Stemming approaches are used: Root-based approach (Khoja [5]); Light stem-based approach (Larkey [6]), and the statistical stemmer approaches (Root Extractor [7]). However, still there is no complete stemmer for Arabic language.

The aim of this paper is twofold: 1) to compare the accuracy of the three existing techniques used for stemming Arabic text and identify the technique that generates the best results in terms of accuracy, 2) to exploit the Arabic WordNet (AWN) and use it as a lexical and semantic resource in the conceptual representation approach. Moreover, we incorporate it in a comparative study with other representation modes in order to evidently comprehend its effect.

A new relation "Has-hyponym" is suggested to be used in addition to other previously used relations like “Synset”, “term+ Synset”, and “all Synsets”. The main contribution of this research is to answer the question, is it enough to use preprocessing operation (find roots or stems) with Bag of words to get good classification results? What is the effect of using part of speech tagger on the classification accuracy for Arabic language?

Does conceptual representation enhance the Arabic classification performance? Finally, which semantic relation positively affects the classification accuracy?

The rest of this paper is organized in five sections. Section two discusses the related works. The suggested approach is explained in section three. System evaluation and effectiveness measure are illustrated in section four. In section five, the assessment of the experimental results is discussed. Finally, we concluded in section six.

2. Related Works

There are considerable amount of work that have been recently conducted to study. According to [3], there’s another productive and efficient technique specialized for Arabic text stemming while enhancing its accuracy. This new strategy merges three known Stemmers known as Khoja, Light stemmer and n-Gram. In addition to that, they use the Naïve Bayesian (NB) algorithm to stratify all the texts. Eventually, they ended up with a Macro F1 average or classification of 0.83.

The authors [4] developed an effective hybrid approach where numerous stemming algorithms used in the pre-processing of the texts. It works by breaking down words into their roots and stem. A function similarity performed by the root-based approach (e.g. Khoja), the stem-based approach (e.g. Larkey) as well as the statistical approach (e.g. Root Extractor). The previously suggested hybrid approach turned out to be of a superior efficiency compared to the other stemming ones.

As for the author [8], he proposed a feature reduction method that aims to enhance the effectiveness of the Arabic text classification through artificial neural network and support vector machines. Its main goal is to reduce the number of features for the process. Multiple experiments were conducted yielding significant results that emphasize the superiority of artificial neural networks over support vector machines with an executive function computed by macro averaging F1 measure.

By benchmarking these three stemming strategies according to their classification accuracy, the dictionary-lookup stemming surpassed the root-based stemming and light-stemming methods for ANN classifier. As for the SVM classifier, the light stemming turned out to be of a higher efficacy compared to the root-based stemming and dictionary-lookup stemming methods.

In [9], the author examines an approach for document classification based on the WordNet notion if the text representation is limited to a set of words, it can omit possible terminologies. This strategy works by selecting the generic and nonexclusive terminologies from the text to integrate them at the next stage with the terms in different ways to form a new representation. This method was tested in multiple experiments using the multivariate chi-square test to reduce the dimensionality. It was concluded that this approach
has a significant impact particularly on raising the macro-averaged F1 value.

In [10], the author presented multiple new methodologies for automated categorization of Arabic text documents. These methodologies combine the well-known Bag-of-Words (BOW) as well as the Bag-of-Concepts (BOC) text representation patterns alongside Wikipedia as a source of knowledge. Three distinctive instrument learning based classifiers were used. The efficacy of the models was assessed by a standard BOW scheme and a concept-based scheme.

In [11], the authors generated a conceptual framework of texts through the WordNet. Their model was constructed by clear and genuine terminologies derived from the documents. The manipulated the terminologies concepts of WordNet and their combination. To apply text categorization, they utilized three algorithms: SVM, Decision trees, and KNN. They experimented their model on two distinctive corpora: the first one consisted of 11 categories of Reuters for a total of 21578 articles. As for the second one, it consisted of 7 groups with 20 other documents. They concluded that a combination between terminologies and concepts yielded significant results concerning the three training algorithms. This conclusion is especially significant for the decision trees algorithm.

3. Arabic Text Classifier: the Developed Approach

Most text classification systems mainly consist of preprocessing phase, feature selection phase, and classification phase. Fig. 1 illustrates the developed Arabic Text classification approach.

3.1 Preprocessing Phase

In preprocessing phase, the size of the document needed to be classified is significantly reduced. The main preprocessing task is removing punctuation marks, numbers and words written in different languages, in addition to stop words (prepositions and pronouns), in order to enhance the text classification technique. This phase also includes normalizing the documents by replacing letters ("\1\1") with (""), the letter ("\c") with (""), and the letter ("\c") with (""). The rest of the words are kept and called “keywords” or “features”. However, in large files, the number of these keywords is usually large and needed to be filtered. Therefore, their number is reduced by removing redundancy wherever exists.

3.2 Features Extraction/Selection Phase

There are two types of text features, external and internal features. Basically, external features are not related to the content of the text such as author name, publication date, author gender, and so on. On the other hand, internal keywords reflect the text content and are mostly linguistic features, such as lexical items and grammatical categories [1]. In this work, words are manipulated as a feature on four levels: using a bag of words form, word stem (where suffix and prefix were removed), word root (where suffix, prefix, and infixes are removed), and word concept. With all these features, it is important to extract and generate the frequency list of the dataset features (single words) and save it in a training file. As for feature extraction, the output result is a long list of features in which not all of them are necessary for the classification operation. Different techniques were suggested to solve such problem and to help selecting the most representative features for each class. The most popular methods for Arabic text classification are Term Frequency (TF), Chi Squared (CHI), Document Frequency (DF) and their different versions, and Information Gain (IG). In this work, we used Term Frequency (TF) in feature selection by assigning the weight to be equal to the number of occurrences of term \( t \) in document \( d \). This weighting scheme is referred to as term frequency and is denoted TF\(_{t,d}\), with the subscripts denoting the term and the document in order [12].

The main features used in this work are:

A) Arabic Stemming Algorithms

As stated before, stemming algorithms are extremely helpful to breakdown words to one form; this form can be termed as root or stem. Stemming can be explained as the process of removing
prefixes, infixes, or/and suffixes from words to reduce these words to their stem or roots. In this case, resulted roots or stems are called terms. Three different techniques of stemming are applied and tested (Khoja stemmer, light stemmer, and root extractor).

To show different stemming examples, random samples of words from BBC dataset are illustrated (see Table 1).

The results in Table 1 shows that there are some words stemmed to incorrect root, and this is due to the deficiencies in each type of stemmer as illustrated in Table 2.

### Table 1. Stemming examples, from BBC dataset

<table>
<thead>
<tr>
<th>Original Terms</th>
<th>Khoja Stemmer</th>
<th>Light Stemmer</th>
<th>Root Extractor</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Derubuk (wsetathur (alqa'ah)</td>
<td>(Derub) (therar)</td>
<td>(Derubuk) (setathur)</td>
<td>(Derub) (sther)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(Laqaa)</td>
</tr>
</tbody>
</table>

### Table 2. Deficiencies of Stemmers’ Type

<table>
<thead>
<tr>
<th>Stemmer Algorithm</th>
<th>Weakness</th>
</tr>
</thead>
</table>
| Khoja             | • The root Dictionary requires an update to ensure that new detected terms are correctly stemmed  
|                   | • If the root contains a weak letter (i.e. alif, waw or yah), the form of this letter may change during derivation. To deal with this, the stemmer must be checked to see if the weak letter is in the correct form. If it is not, the stemmer produces the correct form of this weak letter, which then gives the correct form of the root.  
|                   | • Replace a weak letter with (و ي ا) (و ي ا) which produces an incorrect root. For example, the word (“munathmat”) is stemmed to (“thama” “ثما”) instead of (“nath’ama” “نثمة”) |
| Light             | • Light stemming removes of affixes, predefined in the list, without checking if the remainder is a stem. And in some cases, truncates it from the word and produces an erroneous stem (e.g.”bustan” “بستان” gives “busta” “بستة”).  
|                   | • There is no standard algorithm for Arabic light stemming; all trails in this field were a set of rules to strip off a small set of suffix and prefixes. Also there is no definite list of these Strippable affixes. (M. Hadni, 2013). |
| Root Extractor    | • Sometimes in the Root Extractor stemmer, the letter with three smallest product values represents the wrong root. For example (“amalieat” للعمليات will produce the root (“la’al” لعال, while the correct root is “a’mal” عمل). |
B) Part-of-Speech Tagging (position tagger)

In this work, we consider terms having noun and adjective part of speech tag only. Obviously not all word forms affect the document's meaning in the same way. For instance, nouns contribute effectively to the meaning while adverbs do not. Hence, we make extensive analysis to every word in text categorization.

C) The Representation "Bag of Words"

Bag of Words representation originated from the vector model framework and it is considered the simplest representation of texts. Within this representation, the text is transformed into vectors of words in a condition of excluding any distance between words [11]. On the other hand, the representation has two important deficiencies, which are polysemy, and synonymy. These occur due to the ambiguity of words and the insufficient information about word's relations. Therefore, we used this work conceptual representation.

D) Representation Based on Concepts

We relied in this work on vector formalism in which vector elements are now related to text concepts rather than text terms. In order to use such representation, we needed to project the terms on a lexicon such as WordNet [11]. By definition,AWN is considered as a lexical reference system whose design was formed by modern psycholinguistics theories examining human semantic system[11]. In AWN nouns, verbs, adverbs, and adjectives are arranged into sets of synonyms (synsets) in which every set represents a lexical concept. Through conceptual associations, every single synset is connected to a different one. The most common association in WordNet can be known as hypernymy, and hyponymy. Hypernymy class includes the building notions whereby generalizability of the associations can be easily sought. As for Hyponymy, it is the exact opposite of Hyperonymy[11].

3.3 Classifier

There is no text classification algorithm considered as the best and absolute one; every algorithm has its own uniqueness as well as its own pros and cons. However, the most popular classifiers are C4.5 decision tree, SVM, K-NN and Naive Bayes algorithms which are applied for text classification [1,14]. The Naïve Bayes classifier is chosen for text classification, which is a classifier that is known as a practical probabilistic and has been applied in many applications. The Naïve Bayes algorithm is based on Bayes rule and conditional probability. Previous researches have proven that the Naïve Bayesian classifier is one of the most efficient and effective classifiers in terms of computation. It can be easily used in data mining applications[4].

4. Evaluation and Effectiveness Measures

To evaluate the developed approach, first, a suitable dataset is needed. Second, effective measure should be specified. A dataset or a corpus is a group of text documents that is categorized under various classes. Lately, it has become highly significant to create an Arabic Corpus as it provides help for all current and future researchers in linguistics topics. As a result, we used BBC dataset as a benchmark dataset for Arabic Language [15]. BBC Arabic Corpus was collected from BBC Arabic website (bbcArabic.com.). The corpus includes 5258 text files. Each text file is assigned to 1 of 6 categories (Middle East News, World News, Business & Economy, Sports, Religions, Science and Law). The dataset is linearly non-separable.

The adopted efficient evaluation was Macro-averagedF1 tested. It is a setup from F1 measure which combines recall and precision in an equally weighted manner.

\[
Precision = \frac{tp}{tp+fp} \tag{1}
\]

\[
Recall = \frac{tp}{tp+fn} \tag{2}
\]

\[
F1 = 2 \frac{Precision \times Recall}{Precision + Recall} \tag{3}
\]

K-fold cross-validation is used in this research to ensure that the system produces reliable results. K is set to 10 in keeping to the precedent established in prior research. The MacroF1 is the harmonic average of the F1 for all distinctive categories, where all the categories are tested in an equal manner. As a result, it can simply be affected by the rare categories [5].

5. Experimental Results: Assessments

This section is concerned with the experimental results and the assessment of these results. Three different document representations are used (Bag of words, position tagger, and concept based) with NB classifier. The classification accuracy of each representation form and combination of them will be illustrated and discussed.
A) Bag of Word
In this work, Bag of Word (BoW) is adopted with multiple stemmers, with and without position taggers to test the possibility of enhancing the accuracy in different cases.
Table 3 shows the results for applying Bag of Word with the existing three types of stemmers (Khoja, Light, Root Extractor). By comparing the performance of the three mentioned stemmers, we observed that Root Extractor is the best stemmer since it improves the accuracy by 6.2%.
Table 4 illustrates the results for applying Bag of Word with the three types of stemmers (Khoja, Light, Root Extractor), and position tagger. Based on the achieved outcomes, we can clearly note that using the position tagger with the root extractor improves the classification results by 5.6% compared to using Bag of Word with pos. tagger.

Table 3. Results of BoW without and with stemmers

<table>
<thead>
<tr>
<th>BoW with different stemmers</th>
<th>Macro F1 Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>BoW</td>
<td>0.68008</td>
</tr>
<tr>
<td>BoW+khoja stemmer</td>
<td>0.71357</td>
</tr>
<tr>
<td>BoW+Light stemmer</td>
<td>0.72123</td>
</tr>
<tr>
<td>BoW +Root Extractor stemmer</td>
<td><strong>0.74250</strong></td>
</tr>
</tbody>
</table>

Table 4. Results of BoW with and without position tagger

<table>
<thead>
<tr>
<th>BoW with different stemmers and Tagger</th>
<th>Macro F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>BoW +Pos. tagger</td>
<td>0.69096</td>
</tr>
<tr>
<td>BoW +Pos. tagger+khoja stemmer</td>
<td>0.71134</td>
</tr>
<tr>
<td>BoW +Pos. tagger +Light stemmer</td>
<td>0.7140</td>
</tr>
<tr>
<td>BoW +Pos. tagger + Root Extractor stemmer</td>
<td><strong>0.74698</strong></td>
</tr>
</tbody>
</table>

B) Concept Base Representation
Relation between concepts is considered very important in capturing the ideas in texts. Recent researches showed that replacing terms with concepts without taking into consideration the relation did not improve the accuracy significantly [9]. Based on that, we suggest using “Has-Hyponym” relation, and then we added the frequency of “Hyponym” relation to the concept frequency to enhance text classification accuracy.
Tables 5 summarizes the results of the used approach while varying different features such as Synonym (concept), terms + synonym, set of all synonym (bag of concept) and the proposed feature Has-Hyponym. The results showed that the best performing feature is the new “Has-Hyponym” relation without pos tagger as it improves the accuracy by 7.4% compared to Bag of word representation.
Table 5 illustrates the results of applying different features such as Synonym (concept), terms + synonym, set of all synonym (bag of concept) and the proposed feature Has-Hyponym with position tagger. By comparing the obtained results, we observe that with position tagger, Has-Hyponym provided the best results and improved the accuracy by 7.8%.

Table 5. Results of Concept Relations without Position tagger

<table>
<thead>
<tr>
<th>Semantic Features</th>
<th>Macro F1 Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Synset (Synonym)</td>
<td>0.71489</td>
</tr>
<tr>
<td>Term + First Synset</td>
<td>0.72165</td>
</tr>
<tr>
<td>Bag of Concepts (List of synsets)</td>
<td>0.74799</td>
</tr>
<tr>
<td>Has-Hyponym</td>
<td><strong>0.75437</strong></td>
</tr>
</tbody>
</table>

Table 6. Results of Concept Relations with Position tagger

<table>
<thead>
<tr>
<th>Semantic Features and Tagger</th>
<th>Macro F1 Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Synset (Synonym) +pos tagger</td>
<td>0.72538</td>
</tr>
<tr>
<td>Term + First Synset +pos tagger</td>
<td>0.718096</td>
</tr>
<tr>
<td>Bag of Concepts (List of synsets)+pos tagger</td>
<td>0.72067</td>
</tr>
<tr>
<td>Has-Hyponym +pos tagger</td>
<td><strong>0.7589</strong></td>
</tr>
</tbody>
</table>

6. Conclusion and Future Work
In this work, a comparison between different types of stemmer presented. The root extractor with the position tagger showed the best performance among all other stemming approaches. In addition to that, the conceptual representation using WordNet concepts is used. In this approach, the “Has-Hyponym” relation out performs the other relations, especially when position tagger is combined with it. As future work, we will suggest new rooting approach based on Arabic WordNet. We also
suggest generating and trying more combinations between conceptual representation relations.

References


