Arabic Text Dimensionality Reduction Using Semantic Analysis

ENAS SEDKI, ABDELFATTAH ALZAQAH, ARAFAT AWAJAN

Computer Science Department
Princess Sumaya University for Technology
P O Box 1438, Al-Jubaiha, Amman 11941
JORDAN
awajan@psut.edu.jo

Abstract: - An efficient method to compress and reduce the dimensionality of Arabic texts using semantic model-based representation of text is introduced. The proposed system creates equivalence classes, where similar words, generated according to the rich productive morphology of the language and based on the stem-root-pattern paradigm, are grouped together and represented by a class identifier. In addition, synonyms and similarly named entities are regrouped in order to improve the representation and reduce its size. The reduced representation of the text is accessible to most machine learning algorithms and natural language processing applications that require heavy computational complexity. Distributional similarity measures were used to create equivalence classes of similar words. These measures were applied to the word-context matrix associated with the document in order to identify similar words based on a text's context. The results confirmed that the proposed method shows that incorporation of sematic information in vector representation is superior to classical bag-of-words representation, in terms of size reduction and results quality of applications. The best results are achieved for the clustering of words that are semantically similar, based on their stems. In addition, regrouping differently named entities representing the same concepts improved the reduction amount by 5%.

Key-Words: - Semantic vector space model, word-context matrix, Arabic natural language processing, text dimension reduction, semantic feature extraction

1 Introduction

Extraction of useful information from the huge amount of data available in electronic form can be difficult, because it is usually recorded as unstructured free text. By reducing text dimensionality, we can enhance text representation, and consequently improve the results and performances of different natural language processing tasks. In this instance, the learned low dimensional model can be viewed as a compact representation of text that can be used as an input for more complex applications, including important information extraction and text mining [1]. In addition, the compact representation can itself be useful for storing and transmitting text via different communication channels.

Reducing texts dimensionality techniques aim to reduce the space needed for representing texts, reduce the time required for data manipulation, and improve the performance of different natural language processing (NLP) tasks, therefore increasing effective data density. These techniques reduce text dimensionality by eliminating or minimizing text redundancy and producing a unique presentation of similar terms or concepts that occur in the original texts. They are considered to be lossy compression techniques, because it is impossible to regenerate the original texts from their output results.

A variety of approaches for reducing the dimensionality of texts have been investigated. These have generally been based on the representation of textual documents using the vector space model (VSM), where the components represent the different features of the text, principally its terms or words [2]. However, this representation suffers from a lack of semantics information and relationships between its components.

The present study explored a new computational approach to Arabic text representation, based on a shallow semantic analysis. The proposed method transformed the text into a semantic VSM by clustering different words occurring in different sentences that refer to the same concepts, thereby reducing redundancy in text representation. Removal of duplication helps in reducing the size of

texts, and improves the quality and accuracy of NLP applications, such as summarization and keyword extraction. We used statistical approaches to measure semantic similarity between words in order to create a new vector space representation of the texts. Lexical-semantic resources, including thesauri and WordNet, and manually constructed resources of named entities (NE), were incorporated into the proposed system to improve text representation by creating word clusters of similar or related words extracted from the same root or stem, and regrouped along with their synonyms. However, many synonyms, as well as words, generated form the same root may appear semantically similar in some contexts, but not in others. Distributional similarity measures were used to capture the semantic similarity of the words, based on their context, and to create clusters of similar words. The proposed text representation is a reduced representation of the text accessible to most machine learning algorithms and NLP applications that require heavy computational complexity. It significantly improved text representation in terms of dimensionality, cost effectiveness, and quality of results of NLP applications.

This paper is structured as follows. We introduce related studies and the primary approaches used for other languages in section 2. In section 3, we present the particularities and features of the Arabic language that may be exploited to reduce Arabic text dimensionality. A detailed description of the proposed technique and its components is provided in section 4. Section 5 describes the different techniques used to group similar words. Finally, section 6 describes the datasets used for the evaluation of the proposed method and discusses the results of the conducted experiments.

2 Related Studies

In this section, we review related studies of text dimensionality reduction in general, and those dedicated to texts written in Arabic in particular. Dimensionality reduction can take numerous different forms, and can be implemented in a variety of ways, including via text compression techniques, bag-of-words representation, keyword extraction, VSM-based representation, and semantic VSMs. Text compression techniques use data compression algorithms to reduce the size of data, in order to save both storage space and transmission time. However, they generally ignore linguistic features and the semantic relationships between the terms of the text, which is considered their main disadvantage [3]. In the bag-of-words representation, the text is processed to extract the unique words that are present, and the list of these words is considered to be the reduced representation of the text [1]. Keyword representation replaces the text with a very limited list of terms that represent the essence of the topic of a document in condensed form [4][5]. Keywords are widely used to identify text documents in applications such as documents analysis, document indexing, and information retrieval systems, and to improve the functionality and performance of other applications, such as digital library searching, Web content management, document clustering, categorization of large document collections, and text summarization [5].

The most common text dimensionality reduction techniques are based on the representation of textual documents using the VSM, introduced by Salton [2], which was used to represent collections of documents. In a collection of n documents that contain m unique terms, each document is represented by vector Di = <di1, di2, …,dim> of dimension m that represents a point in a space or a vector in the vector space. Component dij represents the frequency or the weight of the jth term in document Di. A collection of documents can then be represented as a term-by-document matrix of column vectors Di, such that the m rows represent terms and the n columns represent documents. In vector Di, the sequential order of the words and the structure of phrases are lost. However, these vectors primarily capture the similarity of documents, which may be viewed as an important aspect of semantics [6].

Recent studies of text dimensionality reduction approaches have examined the incorporation of semantic features with the VSM to construct more efficient representation that can be used with different NLP tasks, such as texts categorization. Turney and Pantel [6] showed text representation using a word-context matrix, where the focus is on the words' vectors, which is used by different authors to measure the similarity of words. In this representation, the context is given by words, phrases, sentences, or such patterns. The context of a word is always very difficult to define, but in written text, it is often given by neighbor words that occur in the same sentence. Hence, it can be can be measured by the co-occurrence frequency [7].

These different approaches to text dimensionality reduction can be classified into two groups: language-independent approaches and language-dependent approaches. Singular value decomposition and latent semantic indexing techniques are VSM language-independent reduction methods, and they reduce the

dimensionality of the vector space by providing a reduced rank approximation in the column and row space of the document matrix [8]. This ignores the linguistic features of the text's language and considers the words as abstract orthogonal dimensions. Language-dependent techniques investigate the use of the linguistic features of the text language to reduce the VSM representation of a text. Van Rijsbergen [9] suggested the use of a language-dependent approach based on stemming techniques for reducing the size of the index term, and therefore achieving a high degree of relevancy in information retrieval. He found that text representation based on stems reduces the size of the document by 20 to 50%, compared with the full words representation [7].

Text dimensionality and text compression algorithms have been successfully developed and implemented for documents in European languages. However, Arabic has received little research attention in this context, and there are few related publications [10][11][12][13]. The main approaches used have been based on the use of stemming techniques, light stemming techniques, and word clustering. Stemming techniques replace the words with their roots, light stemming removes suffixes and prefixes attached to words, and word clustering techniques group synonyms [13]. It was found that the stemming technique gives the best results in terms of size reduction. These studies used different measures to determine the significant terms representing the text. Harrag et al. [10] compared and evaluated five dimension reduction techniques: stemming, light-stemming, document frequency (DF), term frequency-inverse document frequency (TF-IDF), and latent semantic indexing, and found that the DF, TF-IDF, and LSI techniques were the most effective and efficient of these methods.

The published studies on Arabic texts have focused on the amount of reduction realised and have ignored the impact of the reduction on the meaning of the words; hence they have generally ignored the semantic content of texts and do not account for the semantic relationships that may exist between words and terms, such as synonyms and NEs. Therefore, there is a need to develop new techniques that consider the different aspects of text semantics, in addition to the linguistic features of the language, to devise more accurate, and reduced, representations of Arabic texts.

3 Features of the Arabic Language

Arabic is a Semitic language, and is characterized by its complex and rich inflectional and derivational morphology system. Three classes of words exist, namely, derivative words, non-derivative words, and stop words. The majority of Arabic words are derivative words formed by combining two types of morphemes: templatic morphemes and concatenative morphemes [15], and they are generated according to the root-and-pattern scheme or templatic morphemes. Tens of words (surface form) can be derived from one root, according to a predefined list of standard patterns called morphological patterns or balances. A word may then be represented by its root, along with its morphological pattern.

The concatenative morphology of Arabic allows for the creation of an important number of variants of the same single word stem by adding affixes and clitics. The affixes determine the word's various attributes, such as person, gender, number, and tense, while the clitics are symbols of one to three letters, each representing another token, such as a preposition, conjunction, definite article, or object pronoun. There are two types of clitics, namely, proclitics and enclitics. Proclitics precede a word, for example, the definite article. Enclitics follow a word, for example, object pronouns.

The reduction of Arabic texts faces a variety of problems, as follows:

1. The abundance of unique word forms that result from the rich morphology of the language. There are many unique word forms in Arabic, and the token-to-type ratio for Arabic is much lower than that for English [14].
2. Certain Arabic letters can be written in different ways, leading to a more sparse representation. For example, the letter ALIF may appear with or without HAMZA or MADDA.
3. Special marks, called diacritics, are used as short vowels and may appear optionally in texts.
4. Arabic is rich in synonyms, since they are generally appreciated in written Arabic texts. Figure 1 shows this richness by giving 18 possible synonyms of a single word: ثورة θwrħ, which means "revolution" in English.
5. Named entities recognition (NER). This difficulty is mainly due to the lack of capitalization, the lack of uniformity of writing styles, and the shortage of available and annotated resources. Thus, the

morphological analysis of texts should include a component that is capable of detecting and extracting the NE before performing the analysis at the word level.

عصيان، انقلاب، اضطراب، انتفاضة، انفعال، تمرد، شغب، عصيان، مرج، هبة، هرج، هيجان، هبوب، هياج، تقاتل، تحرك، اهتياج، غضب

Fig. 1. Possible synonyms of the Arabic word "ثورة : θwrħ"

4 Proposed Approach

This study was motivated by the importance of representing texts in a compact manner, without losing significant information. The proposed approach was based on statistical measures and semantic similarity of words, in addition to the linguistic features of the language, and was composed of six different phases. In the first phase, the text was tokenized by extracting the individual words and tokens. The second phase recognized the possible NEs existing in the text. The third phase was a morphological analysis phase, aimed at defining the part of speech, as well as the root and pattern of the derivative words and the stems of non-derivative words. The fourth phase was a text-cleaning operation, which consisted of discarding stop words. In the fifth phase, the word context matrix was calculated to reflect the semantic relationships that link neighboring words. Finally, the sixth phase used shallow semantic analysis, based on similarity measures and semantic relationships between words to construct equivalent classes, regrouping similar words after different levels of analysis. The result of these different phases is a new presentation of the text as a vector of entities, where each entry stands for a group of similar words that represent the same concept or meaning.

4.1 Text Tokenization

The proposed system began with text tokenization, in which the proposed tokenizer detected and isolated individual words. The following rules were used to define the tokenizer and its grammar: [16]

- A token is a sequence of Arabic letters and Arabic diacritical marks.
- A token is separated from other tokens by special characters (a space or punctuation marks).
- The different types of clitics (preposition, conjunction, definite article, or object pronoun) are considered as other tokens attached to the word, and they must be taken out of the word.

4.2 Named Entities Recognition

The NER phase aimed to discover entities, such as proper names, temporal and numeric expressions, names of organizations, locations, etc., in a text. This phase was very important, as it eliminated non-useful morphological analysis that may generate errors and inappropriate clustering of words. In this study, we implemented an NER system that uses a specific Arabic gazetteer of NEs. This recognizes and extracts the NEs by comparison with the elements listed in the gazetteer, and replaces the recognized NEs with their reference in the gazetteer.

4.3 Morphological Analysis

The third phase applied a morphological analyzer developed by the authors on the basis of two freely available analysers: the Alkhalil Morph-Syntactic System (AMSS) [17] and the Stanford Arabic Part-of-Speech tagger [18]. This analyser transformed the text into a sequence of tokens, whereby each one was labeled to identify its type: derivative, non-derivative, and stop words, as well as its morphological structure.

The AMSS identified all the possible morphological and syntactic features, specifically proclitics, prefixes, stems, word types, word patterns, word roots, part of speech (POS), suffixes, and enclitics, while the Stanford Arabic parser provided the part of speech tag associated with the words, given that they had already been tokenized. The POS tags were compared with those provided by the AMSS, and only the compatible solutions provided by the AMSS were retained as final features of the word. A simple greedy regular expression-based stemmer was developed to extract the stems of non-derivative words that the AMSS failed to analyze. This stemmer was repeatedly applied until a word stopped changing, producing a new representation of each word as a sequence of clitics, suffixes, and stems. The preprocessing phase assigns to each input word a type (derivative, non-derivative, or stop word), its stem, its POS tag, and the root and pattern of the derivative words. [7]

4.4 Text Cleaning

Although limited in number, stop words are the words that are most frequently found in texts, and generally account for over 40% of their size. They are generally considered uninformative terms with little semantic discrimination power. The corpus created by [19] shows the irregularity of the word distribution in a collection of 1.7 million words (token) and 89,734 distinct words (types) extracted from the BBC Arabic news collection. The 10 most common stop words occur 215,259 times, while 34,637 words, representing 38.6% of the words in the list, occur once each.

The text-cleaning phase removed from the language the most frequent words that are considered uninformative terms. It used a lexicon of stop words and the list of clitics, and this process reduced the size of the text by approximately 35%. There are basically two advantages to removing these words. First, the size of the text representation is reduced, and second, it allows computing similarity between sentences to be more accurate and easier to depict [7].

At the end of this phase, the text was transformed into a basic vector space S, where each entry is given by:

- The stem of the word
- The number of words in the text generated from this stem (its frequency).
- The stem type (derivative or non-derivative),
- The morphological structure of the stem (pattern, root),
- The POS

The delimiters of sentences were saved at this level, because the sentence information is required in the semantic analysis of words, as this analysis was conducted only at the level of sentences.

4.5 Word-context Matrix

The most important step in the proposed system was the calculation of the text word-context matrix. This matrix is a co-occurrence matrix, where row i represents an entry (stem) S_i from the vector S, and column j represents a context C_j. Herein, a context C_j of a stem S_i is given by any other entry S_j of the vector S that occurs with S_i in the same sentence. The value of the matrix element $D_{i,j}$ represents the number of times that a term generated from stem S_i occurs with another term generated from stem S_j.

The direct count of occurrences of terms S_i and S_j was corrected to be replaced by the relative weight of this occurrence, using pointwise mutual information (PMI) and positive pointwise mutual information (PPMI), defined by:

$$PMI(S_i, S_j) = log_2 \frac{P(S_i,S_j)}{P(S_i)P(S_j)} \quad (1)$$

$$PPMI(S_i, S_j) = \begin{cases} 0 & if\ PMI(S_i, S_j) < 0 \\ PMI(S_i, S_j) & if\ not \end{cases} \quad (2)$$

Where

$$P(S_i, S_j) = \frac{D_{i,j}}{\sum_{k=1}^{N}\sum_{l=1}^{N} D_{l,k}},$$

$$P(S_i) = \frac{\sum_{j=1}^{N} D_{i,j}}{\sum_{k=1}^{N}\sum_{l=1}^{N} D_{l,k}},$$

$$P(S_j) = \frac{\sum_{j=1}^{N} D_{i,j}}{\sum_{k=1}^{N}\sum_{l=1}^{N} D_{l,k}}$$

5 Terms Clustering

The primary idea in the present study was to regroup similar words and identities that are present in the text and to represent each group by one form.

The reduction of text dimensionality was realized by regrouping the words on the basis of different types of knowledge. The first type corresponded to the knowledge that can be extracted from the word-context matrix, which primarily measures the distributional similarity of words, based on their contextual appearance in the text. The second was based on knowledge obtained from linguistic resources, mainly the morphological structure of words, the list of potential synonyms in the language, and similarly named entities.

The proposed regrouping process analyzed the semantic similarity of words to decide whether or not two words were similar. It created equivalence classes where similar words that are generated according the rich productive morphology of the language, based on the stem-root-pattern paradigm, were grouped together and represented by the class identifier. In addition, synonyms and similarly named entities were regrouped, in order to improve the representation and reduce its size. Distributional similarity measures were used and applied to the word-context matrix associated with the document in order to identify similar words based on a text's context.

5.1 Similarity Measures

The similarity between pairs of words is a measure of the degree of correspondence between their

contexts, and can be seen as the distance between their two vectors in the context space represented by the word-context matrix [7]. Distance measures, such as Euclidean distance, Cosine distance, Jaccard distance, and Dice distance, can be used to measure the similarity of words, where the distance between two word-vectors is always seen as a measure of their semantic similarity. In [7] Awajan proposed the use of the cosine distance as a measure of the words' similarity that best captured this feature. Cosine similarity encodes the similarity between two words by giving the cosine of the angle between their corresponding vectors. Similar words that are detected on the basis of this measure represent potential semantically similar words, as this measure considers the linguistic features and relationships between words, such as synonyms and POS, to make the final decision on similarity.

5.2 Named Entities Grouping

In this phase, we solved the problem of having the same entities appearing in the same texts in different forms. For example, the three named entities "الأردن", "المملكة الأردنية" and "المملكة الأردنية الهاشمية" refer to the same entity "Jordan" ,but they appear differently in a text. This step replaced them by one surface by referring to a table that regroups these similarly NEs. Table 1 shows other examples selected from the predefined table of NEs.

5.3 Stems-Based Grouping

The production of different words from the same stem was conducted by adding suffixes and clitics to the stem (concatenative morphemes). These additive morphemes primarily change the word's various attributes, such as person, gender, number, and tense, without changing the concept to which the word refers. Different words generated from the same stem were clustered and represented by their stem in the reduced text. For example, the words (المدرس، مدرس، المدرسات، المدرسون، المدرسين) were clustered in one class represented by the stem (مدرس).

5.3 Stems Based Grouping

The production of different words from the same stem is done by adding suffixes and cletics to the stem (**the** concatenative morphemes). These additive morphemes change mainly the word's various attributes, such as person, gender, number, and tense without changing the concept to which the word refer. The different words generated from the same stem are clustered and represented by their stem in the reduced text. For examples, the words (المدرس، مدرس، المدرسات، المدرسون، المدرسين) are clustered in one class represented by the stem (مدرس).

Table **1**: Named Entities Clusters [7]

Named Entity Cluster Entry	English Meaning	Examples of Clustered NE Members
الأردن	Hashemite Kingdom of Jordan	المملكة الاردنية الهاشمية، الأردن، المملكة الاردنية، الدولة الاردنية، ...
الملك السعودي	The King of Saudi Arabia	الملك السعودي، خادم الحرمين، العاهل السعودي، عاهل المملكة العربية السعودية، ...
بن باديس	Abdelhamid Ben Badis	بن باديس، ألامام بن باديس، الشيخ بن باديس، الامام عبد الحميد بن باديس، عبد الحميد بن باديس، ....
يناير	January	كانون أول، جانفيه، يناير ...
الملكية الاردنية	Royal Jordanian	الملكية الاردنية، الخطوط الجوية الملكية الاردنية، شركة الطيران الاردنية، ...
الواشنطن بوست	The Washington Post	صحيفة الواشنطن بوست، جريدة الواشنطن بوست، الواشنطن بوست، صحيفة واشنطن بوست، جريدة الواشنتن بوست، ...

5.4 Root-Based Grouping

The derivational morphology system in Arabic allows for the production of different stems from the same root, according to different patterns or templates. Different stems generated from the same root may or may not carry the same meaning, according to the meaning added by the pattern. For example, the stems (مدرس، دارس، دراسة) are generated from the same root (درس) according to different patterns, and they refer to the same concept. In contrast, the words (كتاب، اكتتاب), subscription and book respectively, carry different meanings, although they are generated from the same root (كتب). Therefore, we must consider the context, since two different stems generated from the same root may have different meanings if they appear in different contexts.

Different stems generated from the same roots are first tested to calculate their similarity. They are grouped together, and their statistics are accumulated if they are found to be similar. This process considerably reduces the number of rows in the word-context matrix without changing the number of contexts, in order to keep the latter as varied as possible. [7]

5.5 Synonym Grouping

A "synonym" is defined by Merriam-Webster On-Line as "one of two or more words of the same language that have the same or nearly the same meaning in some or all contexts". Synonyms encompass all kinds of words (nouns, adjectives, verbs, and adverbs), but their identification is a challenging task, as perfect synonyms are rare. Two words may often be considered to be synonyms in one context, but not in others. Therefore, the process of synonym grouping must address this issue by considering the contextual information measured by the cosine similarity.

The primary resource for synonyms used in this study was the Arabic WordNet (AWN) [20], which is completed by synonyms found in other resources [21][22]. AWN is composed of groups of near-synonyms and instantiates a sense or concept, called synsets (a synonym set). It is constructed according to the development process used for Princeton WordNet and Euro WordNet, and provides a list of synsets related to a given term and its relationships with other concepts, as well as information regarding the corresponding English/Arabic synsets.
Two ways of grouping synonyms were considered in this study: Synonym' grouping using stems and synonyms grouping using roots. In the grouping using stems, we supposed that two words are synonyms of each other if they satisfy the following conditions:

- They are generated from the same stem
- They have the same POS.
- They are linguistically considered potential synonyms.
- They appear in the text in a similar context, i.e., they are associated with the same set of words (they are similar).

In the synonyms grouping using roots, the conditions of grouping were:

- They are generated from the same root
- They have the same POS.
- They are linguistically considered to be potential synonyms.
- They appear in the text in a similar context.

5.6 Text representation

The new text representation is a set of equivalence classes. The text is transformed at the end of this phase into a semantic vector space model, where each entry represents an equivalence class that clusters terms representing an equivalent meaning and concept. Each entry is described by:

- A word base form associated with the list of its synonyms found in the text or,
- A basic named entity associated with a group of similarly named entities.
- A stem that represents all the non-derivative words generated from this stem and found in the text.
- A root that represents all the derivative words generated from this root.

To each one of these entries, we associate its weight represented by the accumulated frequencies of all its members. The entry weight is used to reflect the importance of the concept or meaning associated with the cluster in the document.

6 Experiments

6.1 Methodology

In our experiments, we began with a collection of documents, pre-processed each document, and transformed the text into a list of stems, where each one was associated with the results of the morphological analysis, encoded the documents into

a word-context matrix, then, using this matrix, we built a vector, where each entry represented a class of similar words. Different tables were used in the implementation of the proposed system: a built-in list of stop words, a table of synonyms, and a table of similarly NEs. A variety of regrouping methods for building the vector were tested and compared. Finally, we evaluated the results and analyzed the impact of the selected criteria on the grouping: stems, roots, synonyms, and NEs. The results were compared in terms of the dimension reduction ratio (DRR), defined by:

$$DRR = \frac{Size\ of\ the\ Reduced\ Represntation}{Size\ of\ the\ Text\ without\ Stop\ words} \quad (3)$$

6.2 Data Sets

A data set of text files of different sizes was selected from different sources, including the BBC's Arabic news collection and the Al-Jazeera.Net website. We selected texts belonging to different categories: political news, economy and business, health, and sciences, and technology. The different approaches proposed in this study were tested with texts from the different categories, and the results were compared for text files of variable sizes: 10 KB, 50 KB, 100 Kb, and 150 KB. All the experiments were conducted after the stop words had been eliminated from the texts.

6.3 Results

Table 2 shows the results of the application of the different reduction-based approaches implemented for one of the text categories considered. It represents the amount of reduction for each step of the systems for different sizes of tested texts. Clustering of words according to their roots gave the best results; however, the main disadvantage of this approach is the fact that some words carrying different meanings may be found regrouped together. Therefore, we can conclude that consideration of the NEs and regrouping the words and their synonyms based on their stems represents the best solution to reduce the size of the representation of texts.

Figures 2 and 3 present the main findings of this study. They compare the amount of reduction on the text dimension of different configuration of the system, and show that the category of the text has little impact on the reduction amount. In addition, Figure 2 shows that regrouping similar words based on the analysis of roots improves the reduction amount by 3 to 5%. Figure 3 shows that the reduction was improved by a factor of 5% on average, by regrouping the NEs referring to the same semantic entities, compared with the methods that consider only regrouping synonyms. Finally, the results show the impact of the size of the document on the performance of the different tested methods, and the performance of all the approaches was better for large-sized texts; the performance increased with the increase of the total number of words in the text.

Table 2. Dimension Reduction Amount (Text Category: Economy)

	Without Name Entity				With Name Entity			
	10KB 1 file	50KB 9 files	100KB 25 files	150KB 35 files	10KB 1 file	50KB 9 files	100KB 25 files	150KB 35 files
Total words	860	4608	9443	14072	860	4608	9443	14072
Total words after removing the stop words	625	3082	6165	9182	626	3415	7015	10375
Unique Words	394	1609	2677	3405	410	1608	2612	3299
Unique Roots	303	988	1528	1851	306	912	1317	1575
Unique Stems	394	1609	2677	3405	410	1608	2612	3299
Stem based Synonym Grouping	394	1603	2665	3384	410	1600	2600	3277
With Root Grouping	389	1583	2620	3297	404	1573	2543	3180
Root based Synonym Grouping	388	1580	2608	3282	403	1570	2530	3162

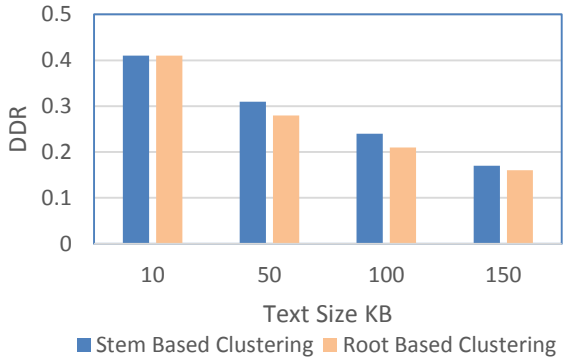

Fig. 2. Clustering based on roots and stems (with NEs)

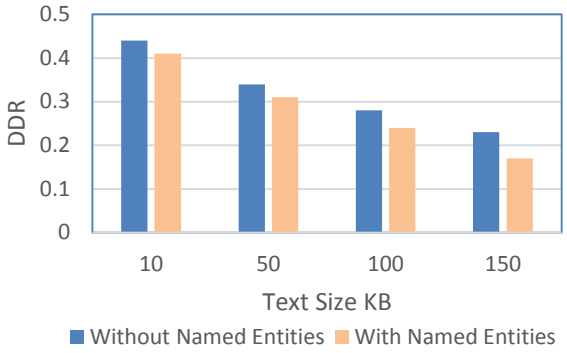

Fig. 3. Impact of clustering similar NEs

7 Conclusions

A new approach for text representation and text dimensionality reduction is presented. This approach was applied for the case of Arabic texts. The dimension of text representation was dramatically reduced by regrouping synonyms, NEs, and words generated from the same roots. The new text representation replaced the original text with a list of terms and semantic features, where each term is representative of an equivalent class of similar terms. Each entry in the new representation was associated with a weight calculated as an accumulation of the weights of synonyms and class members. The conducted experiments show that the proposed method can reduce the dimensionality of texts. Furthermore, it improves the results that can be obtained with different applications, such as keyword extraction and text categorization, in words regarding precision and time efficiency.

References:

[1] C. A. Martins, M. C. Monard, E. T. Matsubara, Reducing the Dimensionality of Bag-of-Words Text Representation Used by Learning Algorithms, Proceedings of 3rd IASTED International Conference on Artificial Intelligence and Applications, Acta Press, 2003, pp. 228–233.

[2] G. Salton, A. Wong, C. S. Yang, A Vector Space Model for Automatic Indexing. *Communication of the ACM*, Vo. 18, No. 11, 1975, pp. 613-620.

[3] Lelewer A. and Hirschberg S., "Data Compression," Computer Journal of ACM Computing Surveys, vol. 19, no. 3, pp. 261-296, 1987.

[4] Rose, S., Engel, D., Cramer, N., and Cowley, W. 2010. Automatic keyword extraction from individual documents. In Text Mining: Applications and Theory, M. W. Berry and J. Kogan (Eds.). John Wiley & Sons. 3–20.

[5] Awajan A. 2015. Keyword Extraction from Arabic Documents using Term Equivalence classes. ACM Transactions on Asian and Low-Resource Language Information Processing. Volume 14, Issue 2, Article 7 (March 2015), 18 pages.

[6] P. D. Turney, and P. Pantel, From Frequency to Meaning: Vector Space Models of Semantics, *Journal of Artificial Intelligence Research*, Vol. 37, 2010, pp. 141-188.

[7] Awajan A. 2015. Semantic Similarity Based Approach for Reducing Arabic Texts Dimensionality. Accepted for publication in the International Journal of Speech Technology (Springer). Volume 18, DOI. 10.1007/s10772-015-9284-6. 13 Pages

[8] K. Baker, Singular Value Decomposition Tutorial. *Note for NLP Seminar*. 2013, pp. 1-24. Accessed from www.ling.ohio-state.edu/~kbaker/pubs/Singular_Value_Decomposition_Tutorial.pdf

[9] C. J. van Rijsbergen, *Information Retrieval*, Butterworths, second edition, 1979.

[10] F. Harrag, E. El-Qawasmah, A. M. Al-Salman, Comparing Dimension Reduction Techniques for Arabic Text Classification Using BPNN Algorithm, *Proceeding of the IEEE First International Conference on Integrated Intelligent Computing*, 2010, pp. 6-11

[11] M. El-Haj, U. Kruschwitz, and C. Fox, Using Mechanical Turk to Create a Corpus of Arabic Summaries. *P ceding of the 7th International*

Language Resources and Evaluation Conference (LREC 2010). pp. 36–39.

[12] H. Froud, A. Lachkar, S. A. Ouatik, A Comparative Study of Root-Based and Stem-Based Approaches for Measuring Similarity Between Arabic Words for Arabic Text Mining Applications, *Advanced Computing: An International Journal (ACIJ)*, 2012, Vol. 3, No. 6.

[13] R. Duwairi, M. N. Al-Refai, N. Khasawneh, Feature Reduction Techniques for Arabic Text Categorization. *Journal of the American Society for Information Science and Technology*, Vol. 60, No. 11, 2009, pp. 2347–2352.

[14] I. Hmeidi, G. Kanaan and M. Evens, Design and Implementation of Automatic Indexing for Information Retrieval with Arabic Documents, *Journal of the American Society for Information Science*, Vol. 48, No. 10, 1997, pp. 867–881.

[15] N. Habash, *Introduction to Arabic Natural Language Processing*, Morgan & Claypool Publishers, USA, 2010.

[16] A. Awajan, Arabic Text Preprocessing for the Natural Language Processing Applications, *Arab Gulf Journal of Scientific Research*, Vol. 25, No. 4, 2007, pp. 179-189.

[17] A. Boudlal, A. Lakhouaja, A. Mazroui, A. Meziane, M. Ould Abdallahi, M. Shoul, Alkhalil Morpho Sys: A Morphosyntactic analysis system for Arabic texts, *International Arab Conference on Information Technology*, 2010, available at http://www.itpapers.info/acit10/Papers/f653

[18] S. Green, C. D. Manning, Better Arabic Parsing: Baselines, Evaluations, and Analysis, *Proceeding of COLING- Beijing*, 2010, pp. 394–402.

[19] M. Saad, Arabic Corpora Statistics found online at http://sourceforge.net/projects/ar-text-mining/files/ArabicCorpora/ accessed on Dec. 2014.

[20] S. Elkateb, W. Black, H. Rodríguez, M. Alkhalifa, P. Vossen, A. Pease, C. Fellbaum, Building a WordNet for Arabic. *In Proceedings of The fifth international conference on Language Resources and Evaluation (LREC 2006)*. Genoa, Italy, May 22-28, 2006.

[21] Almaany 2014. Dictionary and Glossary available at http://www.almaany.com

[22] D. B. Parkinson, Using Arabic Synonyms. Cambridge University Press, 2005.