TEXT CLASSIFICATION FOR ARABIC WORDS USING REP-TREE

Hamza Naji and Wesam Ashour

Department of Computer Engineering, Islamic University, Gaza, Palestine

ABSTRACT

The amount of text data mining in the world and in our life seems ever increasing and there's no end to it. The concept (Text Data Mining) defined as the process of deriving high-quality information from text. It has been applied on different fields including: Pattern mining, opinion mining, and web mining. The concept of Text Data Mining is based around the global Stemming of different forms of Arabic words. Stemming is defined like the method of reducing inflected (or typically derived) words to their word stem, base or root kind typically a word kind. We use the REP-Tree to improve text representation. In addition, test new combinations of weighting schemes to be applied on Arabic text data for classification purposes. For processing, WEKA workbench is used. The results in the paper on data set of BBC-Arabic website also show the efficiency and accuracy of REP-TREE in Arabic text classification.

KEYWORDS

Data mining, Text classification, Text data mining, Arabic text classification, Pre-processing.

1. INTRODUCTION

Text Mining is an important basic process because of huge availability of text documents which located in various formats. [1] On the other hand, the process of understanding text at human level by machines is basically difficult. The Arabic is employed by quite three hundred million folks in over twenty countries [2]. The Arabic expressive style is additionally employed by several alternative languages like Persian, Urdu, Iranian language and alternative regional languages of Pakistan, Afghanistan and Persia. Following Latin script, it's the second most generally used script within the world. Text data mining has the same targets as data mining including [3], text classification, clustering, document recapitulation, and extracting useful trends. Text mining must predominance difficulty that there is no explicit structure. The unique nature of Arabic language morphological principles called for few of the literature in the field of classification of Arabic texts. Arabic could be a difficult language for variety of reasons [4, 5, 6, 7, 8]:

1- Bound mixtures of characters is written in numerous techniques.
2- Advanced morphology recording as compare to West Germanic.
3- Short vowels which give different pronunciation.
4- A huge number of Arabic synonyms.

We will study the impact of text pre-processing and totally different term weight
Schemes combos on Arabic text as a result of scarceness of literature during this regard. The study will be on a dataset of Arabic words collecting from the BBC-Arabic website.
The rest of the paper is organized as follows: Section 2 reviews related work. Section 3 shows proposed work. Section 4 presents the results, and finally, we tend to conclude the paper in Section 5.

2. RELATED WORK

In the discussion below, we focus on the works addressing Arabic TC. Since the amount and quality of options wont to categorical texts have a direct impact on classification algorithms, the subsequent discusses the major goal of feature reduction and choice and their impact on TC. Duwairi et al. [9] compared between three reduction techniques (stemming, light stemming, and word cluster). K Nearest Neighbor was chosen for (training samples) and testing and therefore the results showed that light stem yielded the highest accuracy and lowest time of model building.

Another study [10] compared 3 Feature Subset Selection (FSS) metrics. They applied a comparative study check the result of the feature choice metrics in terms of exactness. The outcome in general disclosed that Odd magnitude relation (OR) worked higher than the others. Some studies used alternative techniques like N-gram and totally different distance measures and verified their effects on Arabic TC. For example, [11] used a statistical method called Maximum Entropy (ME) for the classification of Arabic words. The author showed that the Dice measures using N-gram outperforms using the Manhattan distance. Al-Zoghby [12] used Association Rules for Arabic text classification, and also he used CHARM algorithm with soft-matching over hard big O exact matching. Data sets consisting of 5524 records. Each record is a snippet of emails having the subject nuclear. The vocabulary size is 103,253 words. Similar classifier was used in [13], but different selection and reduction techniques were applied. The author used normalization, stop words removal to increase the ultimate accuracy. Most of Previous research added to the literature used tiny datasets, and applied one or 2 classifiers to classify one corpus that isn't enough to gauge Arabic TC. In this paper, we provide a comprehensive study for Arabic text classification. We examine the impact and therefore the advantages of employing completely different Arabic morphological techniques with different weight schemes applied on seven corpora by exploitation REP-Tree with combos of weight schemes.

3. PROPOSED WORK

In this section we proposed our work by Implementing and integrating an Arabic morphological analysis tools (khoja light stemming) into leading open source machine learning tools (Weka). The tool is available publically accessible freely at [14]. The implemented Arabic morphological analysis tools were applied on BBC-ARABIC data set. We will apply the REP-Tree as a classification algorithm with a combination of weighting schemes: (Term Frequency, Inverse Document Frequency, and Term Frequency-Inverse Document Frequency).

3.1 Pre-processing

Text pre-processing is a necessary component of any natural language process (NLP) system, since the characters, words, and sentences known at this stage are the basic units passed to any or all more text classification stages, from analysis and tagging elements, like morphological analyzers and part- of-speech taggers, through applications, like data retrieval and artificial intelligence systems.The most used method for text mining presentations is displaying text as a bag-of-tokens (words, n-grams). So we can already reduce, classify, cluster, and compute participate stats over text. These are useful to classify and managing height dimension amount text. The reason that natural language processing normally is so complicated is that text is extremely ambiguous. Linguistic communication is supposed for human consumption and infrequently contains ambiguities beneath the idea that humans are going to be ready to develop context and interpret the supposed meaning. Weighting schemes functionality lies in enhance text document representation as feature vector. Popular term weighting schemes are the following [15]:

Term Frequency: A simple way to start out is by eliminating documents that do not contain all three words "the", "brown", and "cow", but this still leaves many documents. To additional characterize them, we would count the amount of times every term happens in each document and total the all together; the number of times a term happens through document is named its term frequency.

Inverse Document Frequency: - However, as a result of the term "the" is so popular, this may tend to incorrectly emphasize documents that happen to use the word "the" a lot of oftentimes, while not giving enough weight to the additional significant terms "brown" and "cow".The term "the" isn't a decent keyword to characterize relevant and non-relevant documents and terms, in contrast to the less common words "brown" and "cow". Therefore associate inverse document frequency factor is included that diminishes the terms weighting that occur terribly oft within the document set and will increase the burden of terms that seldom occur.

Term Frequency-Inverse Document Frequency: A high weight in tf–idf is reached by a high term frequency (in the given document) and a low document frequency of the term in the whole collection of documents; the weights hence tend to filter out common terms. Since the quantitative relation within the idf's log function is usually larger than or adequate to one, the value of idf (and tf-idf) is bigger than or adequate to zero. As a term seems in additional documents, the quantitative relation within the log approaches one, transfer the idf and tf-idf nearer to zero. In several things, short documents tend to be diagrammatic by short vectors, whereas a lot of larger-term sets are assigned to the longer documents. Normally, all text documents must have an equivalent importance for text mining aims. this means that a standardization factor to be included into the term-weighting to equalize the length of the document vectors [16]. In most cases it's so difficult of the morphological variants to recognize by matching only. The text recognition process need additional algorithm called stemming. Stemming is the term used in linguistic morphology and information retrieval to describe the process for reducing inflected (or sometimes derived) words to their word stem, base or root form generally a written word form. The term after stemming needs not to be typical to the morphological root of the original term; it is usually sufficient that closed terms map to the same stem, even if new term is not in itself a valid stem. Stemming approaches are studied in technology since the Nineteen Sixties. several search engines treat words with an equivalent stem as synonyms as a type of inquiry growth, a method referred to as conflation. For the needs of Stemming, Khoja algorithmic rule are going to be used [17] and it's a popular Arabic Stemmer.Weka (Waikato Environment for Knowledge Analysis) [18] is a popular suite of machine learning software written in Java, developed at the University of Waikato. It is free package obtainable underneath the GNU General Public License.Weka gives a large combination of classification, clustering, and visualization algorithms for data mining, which can be derived through a familiar Graphical User Interface. By using weka software we can choose the 'String To Word Vector' tool with totally different combos, we tend to setup the term weighting combos given in Table one to be passed to our reduced error pruning decision tree 'REP'. Then we make a list of combination between the weighting schemes: Term Frequency, Inverse Document Frequency, and Term Frequency-Inverse Document Frequency (TF, IDF, and TFIDF). Activating the counting words property (CW) and apply the previse weighting schemes, the resulting combinations (described in Table 2) are tfcw, idfcw, tf-idfcw, norm-cw, minFreq3cw, norm-minFreq3cw, and all-minFreq3cw. The term (freq) refers to the number of letters in the word during the classification stage determined by the user as he like.

Table 1: Show Weka weighting schemes- String to Word Vector options.[19]

Weighting schemes	Description
tf	*Log (1+fij),* where *fij* is the frequency of word *i* in document *dj*.
Idf	*fij*log(num of Docs/num of Docs with word i),* where *fij* is the frequency of word *i* in document *dj*.
tf idf	*log(1 + fij) * log (num of Docs/num of Docs with word i),* where *fij* is the frequency of word *i* in document *dj*.
normalizeDocLength	Sets whether if the word frequencies for a document should be normalized or not.
minTermFreq	Sets the minimum term frequency (apply term pruning)
outputWordCounts	Output word counts rather than Boolean 0 or 1(indicating absence or presence of a word).

Two major combinations are used; Bag of Tokens (BOT) (without Khoja stemming algorithm), and term Stemming. Symbols used in the preprocessing combinations for Stem and BOT are (shown in Table 2).

Table 2: Symbols used in experiment setup pre-processing combos for weka weighting schemes.

Symbol	Explanation
CW	Output word counts
tf CW	Apply *TF* transformation on word count
Idf CW	Apply *IDF* transformation on word count
tf idf CW	Apply *TFIDF* transformation on word count
Norm CW	Apply document normalization on word count
minFreq3CW	Apply term pruning on word count that less than 3
Norm minFreq3CW	Apply normalization and term pruning on word count that less than 3

3.2 Problem with encoding in Weak Tool

When you deal with Arabic files in weka such as: Arrf and CSV formats you will face problem with the misunderstanding of the Arabic texts which present if like symbols. The solution is to use a unique encoding from Java to display them under Windows (= "Cp1252").If you change the file encoding to "utf-8" in the system.ini file.

4. EXPERIMENTAL RESULTS AND ANALYSIS

Experiments had been applied on Arabic dataset collected from BBC- Arabic website (http:// www.BBc.co.uk/arabic/).The dataset contains 110 text documents belonging to one of the seven categories (Middle East news, Sport, Health, Compute Technology, Varieties and Communications).For text classification; we use REP-tree with specific training set of 66% cross-validation and comparing them by the G-graph tree. Table 3 shows text mining classification for BOT and Stemmed terms using weighting schemes combinations described in Table 2.

Table 3.A: Text Classification result by REP-Tree for BOT and Stemmed term using different text preprocessing combos.

Weighting Schemes	BOT	Stemmed By Khoja Algorithm	Correctly BOT %	Correctly Stemmed %	Time BOT(sec)	Time Stemmed(sec)
CW	4452	2424	92.3581	97.8581	90.8	34.45
tf CW	4452	2424	92.3581	97.8581	98.6	30.24
IdfCw	4452	2424	92.3581	97.8581	98.6	29.17
TfidfCw	4452	2424	92.3581	97.8581	98.6	29.17
Norm Cw	4452	2424	92.3581	97.8581	74.4	28.21
minFreq3 Cw	337	314	93.2463	95.9234	7.7	3.55
Norm minFreq3Cw	1120	534	88.3552	98.3451	17.5	9.77

Table 3.A and 3.B describe the reduction of dataset text using different weighting schemes combinations. Comparison between BOT and Stemmed term using REP-Tree compared with G-graph-Tree, using term stemming lead to reduce dimensionality for all weighting schemes combinations because stemming reduces the size of the huge text dataset, which have many morphological variants, to their root. The results show that the REP-Tree is more correctly than G-graph-Tree.

Table 3.B: Text Classification result by G-graph-Tree for BOT and Stemmed term using different text preprocessing combos

Weighting Schemes	BOT	Stemmed By Khoja Algorithm	Correctly BOT %	Correctly Stemmed %	Time BOT(sec)	Time Stemmed(sec)
CW	4423	2439	90.8521	95.9234	100.5	38.58
tf CW	4423	2439	90.8521	95.9234	110.6	39.29
IdfCw	4423	2439	90.8521	95.9234	110.6	36.17
TfidfCw	4423	2439	90.8521	95.9234	110.6	39.76
Norm Cw	4423	2439	90.8521	95.9234	99.4	38.54
minFreq3 Cw	328	320	91.5647	93.8854	8.4	6.77
Norm minFreq3Cw	1090	540	89.5372	96.7789	22.5	11.34

Figure 1: is configured from Tables (3.A &3.B) describes the reduction of document data set using different weighting schemes combinations. Comparison between BOT and Stemmed Term, using term stemming lead to reduce dimensional for all weighting schemes combinations because stemming reduce the size of the huge text data set, which have many morphological variants, to their root. Dimensionality dramatically reduced using term pruning with minimum frequency of 3

because there are many infrequent terms in the document collection. Stemming with REP-Tree gives the less value or the optimum value.

Figure 1: Text dataset dimensionality for different text preprocessing combos.

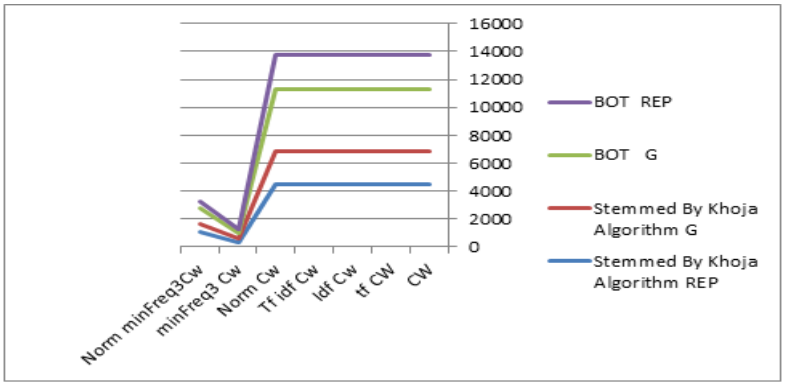


Figure 2 shows classification accuracy for different text preprocessing combinations, cw-norm-minFreq3 gives highest accuracy for REP-Tree Stemmed terms, while cw-minFreq3. Obviously, pruning infrequent terms enhance classification accuracy. The accuracy for REP-Tree stemmed terms is better than G-graph for all preprocessing combinations. Stemming enhance term weighting and this affect classification accuracy.

Figure 2: REP-TREE and G-graph stemmed classification accuracy for each text preprocessing combos.

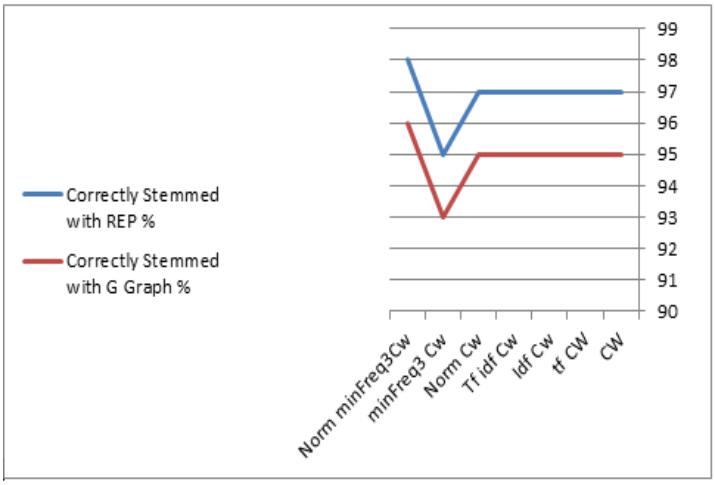
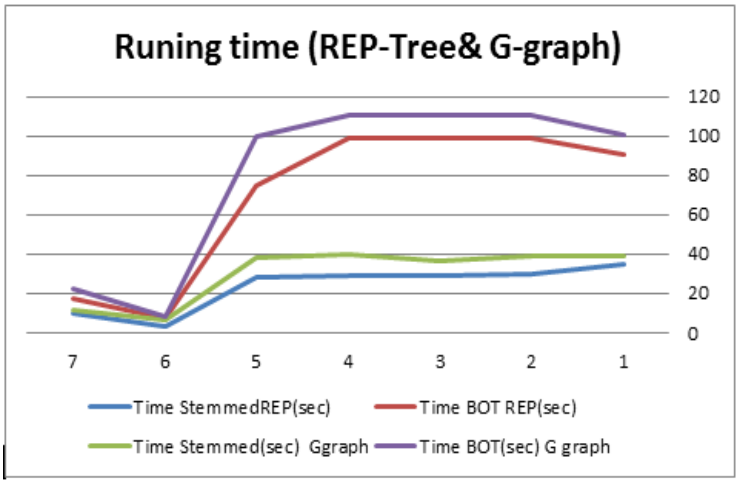


Figure 3 depicts the running time for classification process. Shortest running time is achieved when use term pruning with minimum 3 occurrences. Again, running time for REP-Tree stemmed terms is shorter than BOT for all pre-processing combinations. Term stemming and pruning dramatically reduce dimensionality and enhance classification accuracy and performance. The empirical results also show that cw-norm and cw-tfidf give good accuracy and performance; this may vary from dataset to another. Furthermore, it is known that document normalization and TFIDF work well for large text dataset [20]. Running time with REP-Tree gives the less value or the optimum value.

Figure 3: Text classification running time for REP-Tree and G-graph tree.



5. CONCLUSIONS

Text preprocessing is a core step in text data mining. There are many preprocessing weight schemes combination that can be used for text preprocessing, but it is very difficult to determine the best preprocessing and term weighting. In this paper we have a tendency to examine the pre-processing phase as a main step in Arabic text mining. Empirical results showed term stemming and pruning, document normalization, and term coefficient dramatically cut back spatiality, enhance text illustration and directly impact text mining performance.

REFERENCES

1. Frakes, W. B. Stemming algorithms. In Information retrieval: Data structures and algorithms, W. B. F. a. R. Baeza-Yates, Ed. Englewood Cliffs, NJ: Prentice Hall, chapter 8, 1992.
2. S.al-Emami and M.Usher, "On-line Recognition of Handwritten Arabic Characters ," Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 12, pp. 704-710, 1990.
3. Feldman R., Sanger J., The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data. Cambridge University Press, 2007, PP 320.
4. S. Basu, R. J. Mooney, K. V. Pasupuleti, and J. Ghosh. Evaluating the novelty of text-mined rules using lexical knowledge. In Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2001), pages 233–239, San Francisco, CA, 2001.
5. Al-Marghilani A., Zedan H., Ayesh A., Text Mining Based on the Self-Organizing Map Method for Arabic-English Documents. Proc. of the 19th Midwest Artificial Intelligence and Cognitive Science Conf. (MAICS 2008), Cincinnati, USA, pp. 174-181 , 2008.
6. El-Halees A., A Comparative Study on Arabic Text Classification. Egyptian Computer Science Journal Vol. 20 no. 2 May, 2008.
7. Ghwanmeh S., Applying Clustering of Hierarchical K-means-like Algorithm on Arabic Language. Int. Journal of Information Technology Vol. 3 No 3. 2005.
8. Taghva, K., Elkhoury, R., Coombs, J.: Arabic stemming without a root dictionary. Information Technology: Coding and Computing, ITCC, Vol. 1, pp 152 – 157, 2005.
9. R. Duwairi, M. N. Al-Refai, and N. Khasawneh, "Feature reduction techniques for arabic text categorization," Journal of the American society for information science and technology, vol. 60, no. 11, pp. 2347–2352, 2009.
10. A. Mesleh, "Feature sub-set selection metrics for arabic text classification," Pattern Recognition Letters, vol. 32, no. 14, pp. 1922–1929, 2011.
11. L. Khreisat, "Arabic text classification using n-gram frequency statistics a comparative study," in Conference on Data Mining— DMIN'06, 2006, p. 79.

12. Al-Zoghby A., Eldin AS., Ismail NA., Hamza T., "Mining Arabic Text Using Soft Matching association rules", In the Int. Conf. on Computer Engineering & Systems, ICCES'07, 2007.
13. A. El-Halees, "Arabic text classification using maximum entropy," The Islamic University Journal (Series of Natural Studies and Engineering), vol. 15, pp. 157–167, 2007.
14. WesamAshour, Motaz K. Saad, "Open Source Arabic Language and Text Mining Tools", (2010, August), [Online]. Available: http://sourceforge.net/projects/ar-text-mining.
15. M.M. Gaber, Scientific Data Mining and Knowledge Discovery — Principles and Foundations, Springer, New York, 2010.
16. Said D., Wanas N., Darwish N., Hegazy N.: A Study of Arabic Text preprocessing methods for Text Categorization. 2nd Int. conf. on Arabic Language Resources and Tools, Cairo, Egypt, 2009.
17. Quinlan R., C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, San Mateo, CA. 1993.
18. R.J. McQueen, D.L. Neal, R. DeWar, S.R. Garner and C.G. Nevill-Manning, "The WEKA machine learning workbench: its application to a real world agricultural database," Proc Canadian Machine Learning Workshop, Banff, Canada, 1994.
19. Jing L., Huang H., Shi H.: Improved feature selection approach TFIDF in text mining. Proc. of the 1st int. conf. of machine learning and cybernetics, Beijing, 2002.
20. Said D., Wanas N., Darwish N., Hegazy N.: A Study of Arabic Text preprocessing methods for Text Categorization. 2nd Int. conf. on Arabic Language Resources and Tools, Cairo, Egypt, 2009.

AUTHORS

WesamAshour: WesamAshour is an associate professor at Islamic University of Gaza. He is an active researcher at The Faculty of Engineering. He got his Master and Doctorate degrees from UK. His research interests include data mining, artificial intelligence, reinforcement learning and neural networks.

HamzaNajihas graduated in 2013 with B.Sc. in Software Engineering from University of Palestine Gaza, then start studding M.Sc. of Computer Engineering in the Islamic University of Gaza in 2014/2015.