A Personalized Markov Clustering and Deep Learning Approach for Arabic Text Categorization

Vasu Jindal
University of Texas at Dallas
Richardson, TX 75080
vasu.jindal@utdallas.edu

Abstract

Text categorization has become a key research field in the NLP community. However, most works in this area are focused on Western languages ignoring other Semitic languages like Arabic. These languages are of immense political and social importance necessitating robust categorization techniques. In this paper, we present a novel three-stage technique to efficiently classify Arabic documents into different categories based on the words they contain. We leverage the significance of root-words in Arabic and incorporate a combination of Markov clustering and Deep Belief Networks to classify Arabic words into separate groups (clusters). Our approach is tested on two public datasets giving a F-Measure of 91.02%.

1 Introduction

In the emerging era of big data technology, there has been a widespread increase in information obtained from text documents. Furthermore, with the rapid availability of machine-readable documents, text classification techniques have gained tremendous interest during the recent years. Consequently, automatic categorization of numerous new documents to different categories has become critical for political, social and for news research purposes.

Text categorization techniques have been widely investigated by researchers around the world. However, most of the recent developments in this field are focused on popular Western languages, ignoring Semitic and Middle-Eastern languages like Arabic, Hebrew, Urdu and Persian. As discussed further in related works in Section 2.1, most classification algorithms utilized English and Chinese to validate their methods while works on Arabic are extremely rare. This is primarily due to significant dialect differences between these languages and their complex morphology. Additionally, the presence of various inflections in Arabic as opposed to English makes it difficult for the NLP community to validate techniques on the popular Middle Eastern languages. According to the US Department of Cultural Affairs, Arabic and Urdu are categorized as critical languages and United Nations heavily emphasizes on the social and political importance of these languages. Arabic is listed as one of the six official languages of the United Nations.

In this paper, we present a novel three stage approach to classify Arabic text documents into different categories combining Markov Clustering, Fuzzy-C-means and Deep Learning. To the best of our knowledge, this is the first work that leverages the heavy influence of root-words in Arabic to extract features for both clustering and deep learning to perform classification of Arabic documents. First, we segment each document and extract root-word information. Then we perform clustering with root-word based features using Fuzzy-C-Means and Markov clustering. This allows us to separate documents into unsupervised groups (clusters). We then train a deep belief network (DBN) for each cluster using Restricted Boltzmann Machines. This personalization which is essentially training DBN for each cluster improves the classification accuracy and features extraction. Finally, we generate network graphs of these clusters which can be used for similarity relatedness or summarization in future works. This is the first work to use a modified combination of Markov clustering and personalized deep learning to classify documents into different categories.

The rest of the paper is organized as follows: Section 2 discusses the literature review and Ara-

Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics – Student Research Workshop, pages 145–151,
Berlin, Germany, August 7-12, 2016.

bic morphology. Section 3 focuses on methodology for Markov clustering and deep learning. Section 4 discusses our experimental results and finally, Section 5 summarizes the paper and presents conclusions and future work.

2 Background

2.1 Related Works

As mentioned previously, even though numerous works in text categorization have been proposed for Western languages, works on categorizing Semitic languages like Arabic are very rare in the NLP community.

Most previous works on Arabic text categorization treats documents as a bag-of-words where the text is represented as a vector of weighted frequencies for each of the distinct words or tokens. (Diederich et al. 2003; Sebastiani et al. 2002). We use a similar approach to extract features from documents based on root-word frequency. Early efforts to categorize Arabic documents were performed using Naive Bayes algorithm (El-Kourdi et al., 2004), maximum entropy (Sawaf et al., 2001) and support vector machines (Gharib et al., 2009).

N-gram frequency statistics technique was used with Manhattan distance to categorize documents by Khreisat (Khreisat, 2006) El-Halees described a method based on association rules to classify Arabic documents (El-Halees, 2007). Hmeidi I. uses two machine learning methods for Arabic text categorization: K-Nearest Neighbor (KNN) and support vector machines (SVM) (Hmeidi et al., 2008). An approach for feature selection in Arabic text categorization was proposed using information gain and Chi-square (Yang and Pedersen, 1997). Abu-Errub A. proposed a new Arabic text classification algorithm using Tf-Idf and Chi square measurements (Abu-Errub, 2014). However, all these methods were not very efficient for large datasets giving an accuracy of less than 70% and were unable to classify documents with different diacritics. Diacritics are signs or accents whose pronunciation or presence in a word can result in a different meaning.

Recently, a new technique using a combination of kNN and Rocchio classifier for text categorization was introduced (Mohammad et al., 2016). This approach specifically solves the Word Sense Disambiguation (WSD) problem in a supervised approach using lexical samples of five Arabic words. Although, this method achieved higher accuracy than previous works, usage of only five Arabic words limits usage for larger datasets. Our proposed approach generates multiple roots of each Arabic words addressing the issue of different diacritics in Arabic.

2.2 Arabic Morphology

Arabic belongs to the family of Semitic languages and has significant morphological, syntactical and semantical differences from other languages. It consists of 28 letters and can be extended to 90 by added shapes, marks, and vowels. Furthermore, Arabic is written from right to left and letters have different styles based on the position of its appearance in the word. The base words of Arabic inflect to express eight main features. Verbs inflect for aspect, mood, person and voice. Nouns and adjectives inflect for case and state. Verbs, nouns and adjectives inflect for both gender and number. Arabic morphology consists from a bare root verb form that is trilateral, quadrilateral, or pentaliteral. The derivational morphology can be lexeme = Root + Pattern or inflection morphology (word = Lexeme + Features) where features are noun specific, verb specific or single letter conjunctions. In contrast, in most European languages words are formed by concatenating morphemes. For example in German, 'Zeitgeit'(the spirit of the times) is simply 'Zeit'(time) + 'geist'(spirit) i.e the root and pattern are essentially interleaved.

Stem pattern are often difficult to parse in Arabic as they interlock with root consonants (Abdelali, 2004). Arabic is also influenced by infixes which may be consonants and vowels and can be misinterpreted as root-words. One of the major problem is usage of a consonant, hamza. Hamza is not always pronounced and can be a vowel. This creates a severe orthographic problem as words may have differently positioned hamzas making them different strings yet having similar meaning.

Furthermore, diacritics are critical in categorizing Arabic documents. For example, consider the two words "zhb" mean "to go" and "gold" differing by just one diacritic. The two words can only be distinguished using diacritics. The word "go" may appear in a variety of text documents while "gold" may likely appear in documents containing other finance related words. This is where our personalized deep learning approach is extremely efficient for Arabic as discussed in future sections. For the purpose of clarity, we use the term "root-

words" throughout this paper to represent the roots of an Arabic word.

3 Methodology

Figure 1 presents an overview of our algorithm. In summary, our approach consists of a pre-processing stage, two stages of clustering and finally a learning stage. In the pre-processing stage documents are tokenized and segmented into different words and the Tf-Idf weighted root-word counts are extracted. Subsequently, we cluster the documents by a combination of Markov Clustering and Fuzzy C-means in Step 2. Finally, in step 3, we use deep learning models on each obtained cluster to personalize learning for each root word cluster. Personalization essentially means to train a separate deep belief network for each cluster. Each of these stages is discussed in subsequent sections.

3.1 Pre-Processing

In the pre-processing stage we first remove the punctuation marks, auxiliary verbs, pronouns, conjunctions etc. As we stated in section 2.2, it is very important for processing Arabic to properly use the semantic information provided by root-words (Kanaan et al., 2009). Therefore, representing words presented in a document in the root pattern increases efficiency of classification. Root extraction or stemming for the Arabic dataset is performed using a letter weight and order scheme. For example, the root meaning "write" has the form k-t-b. More words can be formed using vowels and additional consonants in the root word. Some of the words that can be formed using k-t-b are kitab meaning "book", kutub meaning "books", katib meaning "writer", kuttab representing "writers", kataba meaning "he wrote", yaktubu meaning "he writes", etc.

In our method we assign weights to letters of Arabic and subsequently rank them based on their frequency in the document. Root-words of the Arabic word are selected by recurring patterns with the maximum weight. Furthermore, we calculate the standard Tf-Idf frequency of each root word to use as features in clustering and deep learning. Tf-Idf (term frequency-inverse document frequency) is one of the widely used feature selection techniques in information retrieval (Baeza-Yates et al., 1999). Tf measures the importance of a term in a given document while Idf signifies the relative importance of a term in a collection of documents. In the next section, we will discuss the clustering step of our approach.

3.2 Estimation of Initial Number of Clusters

The words frequencies (Tf-Idf) and root-word frequencies are used from the pre-processing stage and grouped into clusters. Once the words are clustered, we consider each document and find which cluster of words it may belong. This is done by rank-matching based approach discussed later. The clustering step is described below.

Consider each document to be P

$$P = \begin{bmatrix} p_{0,0} & p_{0,1} & \cdots & p_{0,W-1} \\ \vdots & \vdots & \cdots & \vdots \\ p_{H-1,0} & p_{H-1,2} & \cdots & p_{H-1,W-1} \end{bmatrix} \quad (1)$$

where $p_{i,0}$ indicate extracted words tokenized from the documents. $p_{i,j}$ are the similar root-words of $p_{i,0}$.

We first estimate the initial number of categories that may be present in a corpus of text documents. The estimation of initial number of clusters is critical for text categorization as many Arabic documents may contain synonyms, different morphologies of Arabic and yet may have a close meaning. We perform estimation by computing the total number of modes found in all eigenvectors of the data. Modes in each eigenvector of the data are detected using kernel density estimation. Then, significance test of the gradient and second derivative of a kernel density estimation is computed. A similar approach was used for bioinformatics data (Pouyan et al., 2015). Briefly, if $E = \{e_1, e_2, ..., e_M\}$ denotes eigenvectors of dataset X, a kernel Gaussian is considered as follows:

$$\kappa(l) = \frac{1}{\sqrt{2\pi}} exp(\frac{-l^2}{2}) \quad (2)$$

where $\kappa(.)$ is the Gaussian kernel and h is the bandwidth. The estimator gradient is written as follows:

$$\Delta\hat{f}(l) = \frac{2}{N \cdot h^2} \cdot \sum_{i=1}^{N} \kappa(\frac{l - e_i}{h}) \cdot (l - e_i) \quad (3)$$

The number of modes is approximated using the number of times, the gradient changes from positive to negative for each projection of data on the eigenvectors. K represents initial number of clusters approximated by summation of all the modes in eigenvectors.

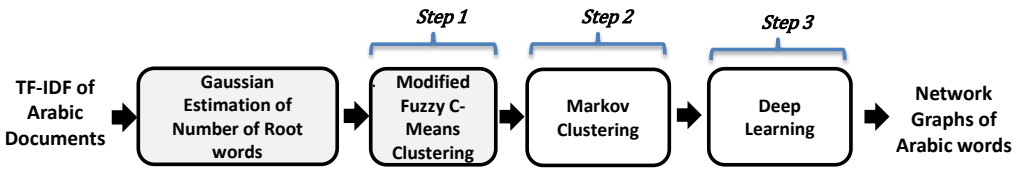

Figure 1: The general view of our proposed method

3.3 Initial Clustering Using Fuzzy-C-Mean

We leverage the heavy influence of root-words in Arabic for clustering words (tf-idf) from the pre-processing stage. As described in section 2.1, new words in Arabic can be formed by filling vowels or consonants. For example, k-t-b is a trilateral root word as it contains a sequence of three consonants. Accordingly, an improved version of Fuzzy-C-Means clustering is developed to calculate the membership probability of each words to its root word. The clusters we obtain are intended to correspond to the different root-words in document. Concisely, $\chi = \{\mu_1, ..., \mu_j, ..., \mu_K\}$ will be centers of K root-words $C = \{c_1, ..., c_j, ..., c_K\}$ which represents potential similarities of N-letter words $X = [x_1, x_2, ..., x_n]^\top$. Words are assigned to different root word clusters by minimizing the following optimization model:

$$\begin{cases} Minimize\ \{J_m = \sum_{i=1}^{N} \sum_{j=1}^{K} u_{ij}^m D_m(x_i, \mu_j)\} \\ \text{subject to:}\ \ \sum_{j=1}^{K} u_{ij} = 1 \qquad \forall i = 1, 2, ..., N \end{cases} \tag{4}$$

where word x_i belongs to root word c_j with the membership probability of u_{ij}. The fuzzification coefficient is selected as $m = 2$ in this work. $D_m(x_i, \mu_j)$ implies the *Mahalanobis Distance* between word x_i and root word c_j. Since membership probability depends on the dispersion of cluster c_j, we use Mahalanobis Distance instead of Euclidean Distance as a distance metric between x_i and population c_j.

A Lagrangian multiplier is used to minimize the optimization problem of Fuzzy-C-Means given in Equation 4. The initial cluster set C will be available after applying the revised Fuzzy-C-Means and used in future steps.

3.4 Merging Clusters Using Markov Clustering

The number of initial clusters may have been overestimated by kernel density estimation in the first stage. We address this issue using Markov clustering, a fast, divisive and scalable clustering algorithm based on stochastic modelling of flow of networks (Van Dongen, 2001). We apply Markov clustering on the initial cluster centers $\mu_1, ..., \mu_K$ to extract the main skeleton of the data cloud. Markov clustering (MCL) has recently emerged as a popular clustering technique for determining cluster networks. The algorithm computes the probability of random walks through a graph by applying two main methods: expansion and inflation. When MCL is applied on centers of initial clusters, the centers corresponding to initial populations will be clustered in the same segments. We extract the final categories of the text documents by merging these clusters. MCL is required as the previous step may have estimated duplicate clusters based on similar diacritics or morphologies. Therefore, these redundant clusters are merged in this stage.

Once the words are clustered, we consider each document and find which most similar cluster of words it may belong to. This is done by rank-matching based approach. We find the maximum match between the 40% most frequent words in the document and every cluster. Cluster of words consisting of the highest number of words from top 40% frequent words in the document is assigned to the document.

3.5 Deep Learning

We further use the state-of-the-art deep learning for extracting features from the clusters and use them for future clustering. We create separate deep belief network classifier for each cluster, which allows us to capture differences in between dialects, topics etc. To reiterate, our novel contribution is a personalized deep learning model for

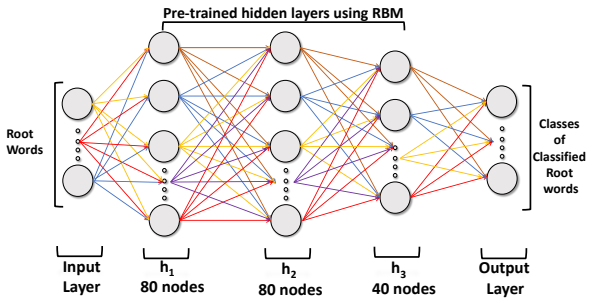

Figure 2: Deep Neural Network for C_1

each root word cluster. Personalized means training of each model for each specific root word, dialects and diacritics. The classifiers get more fine-grained by training one classifier for every cluster.

This personalization poses several advantages: the features learned using deep learning are personalized for each root word. Consequently, the technique is robust against different dialects, scripts and way of writing. Secondly, personalized deep learning models extract features from diacritics. As previously described in section 2.2, diacritics makes Arabic word classification very challenging. By extracting features from diacritics in the context of appeared root-words, our personalized deep learning approach efficiently solves this problem.

We separately train deep belief networks (DBN) on each cluster obtained from Stage 2. For example the deep network contains one input layer, three hidden layers and one output layer for Cluster 1 with hidden layers consisting of 80, 80 and 40 nodes respectively. The input of these classification models are words in each clusters. The activation function of hidden units is the sigmoid function which is traditionally used in nonlinear neural networks. Furthermore, higher number of parameters in neural networks generally makes parameter estimation much more difficult. Therefore, it is inefficient to start training of deep neural networks from random initial weight and bias values. We incorporate Restricted Boltzmann Machines (RBM) to pre-train the network and find good initial weights for training deep belief networks (Le Roux and Bengio, 2008).

Let D_i be a DBN model for cluster C_i. The hidden layers of each D_i are first trained as RBMs using unlabeled inputs. We use Contrastive Divergence-1 (CD-1) algorithm to obtain samples after 1-step Gibbs sampling (Hinton, 2002). CD-1 allows accurate estimation of gradient's direction and minimize reconstruction error. Due to this pre-training, D_i learns an identity function with same desired output as the original input. Furthermore, it enhances the robustness of D_i by learning feature representations of the Arabic words before the final supervised learning stage.

Subsequently, the pre-trained DBN network is fine-tuned by vanilla back-propagation with labeled segments as the input layer. Figure 2 gives the abstract structure of final resulting D_1 after tuning and is also used for identification for cluster C_1. If a new document is added for text categorization then we find it's nearest neighbor cluster and use the deep learning model specific to that cluster. Finally, we plot the network graphs of the extracted cluster words for visualization of the association of these words.

4 Experimental Results

We evaluate our three-stage technique using two popular datasets previously used in Arabic text categorization: 10,000 documents from Al-Jazeera news website (http://www.aljazeera.net) and 6,000 Saudi Press Agency (Al-Harbi et al., 2008) documents. The results are reported using 10-fold cross validation. Our proposed method achieves a precision of 91.2% and recall of 90.9% giving F-measure of 91.02%.

Clustering is performed on a set consisting of the total 12,000 documents, randomly sampled from separately from Al-Jazeera and Saudi Press Agency. We further ran deep learning on each of these clusters and extracted network graphs. Two example networks are presented in Figure 3 and 4. We compare our results with existing approaches in Table 1. We can see that our technique improves substantially on the previous published works. Furthermore, it is capable to categorize different diacritics by using clusters based on root-words. Most misclassified cases in our algorithm due to random outliers and/or mix categories in a document. An example of a random outlier are some recent words which are not influenced by root-words. This can be further improved by using a larger dataset in future works and using new discriminative features for clustering and deep learning.

5 Conclusion and Future Work

This paper presents a novel three-stage technique for Arabic text categorization using a combina-

Table 1: Performance of our algorithm on Al-Jazeera Dataset

Technique	Precision	Recall	Root-words based?	Robust to Diacritcs?
Naive Bayes (El-Kourdi et al., 2004)	62.6%	57.4%	No	No
SVM (Hmeidi et al.,2008)	71.2%	66.7%	No	No
kNN (Mohammad AH et al., 2016)	83.0%	80.2%	No	No
Ours	**91.2%**	**90.9%**	**Yes**	**Yes**

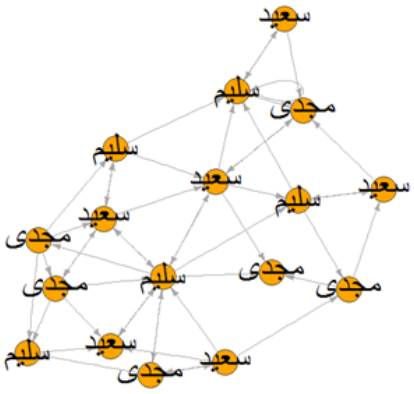

Figure 3: Network of a document's words from Cluster 1

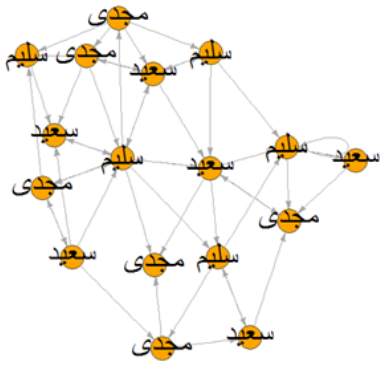

Figure 4: Network of a document's words from Cluster 2

tion of clustering and deep learning. We leverage the influence of root-words in Arabic to extract the features. Our technique is robust against different diacritics and complex morphology of Semitic languages. Furthermore, this procedure can be extended to Persian and Semitic languages like Hebrew which heavily depend on root-words. Future work includes extracting more discriminative features of root-words using deep learning and improving training of our models using larger datasets. We also plan to explore other Arabic morphologies like lemmas used in Arabic dependency parsing (Marton et al., 2003).

6 References

Ahmed Abdelali. 2004. Localization in modern standard Arabic. Journal of the American Society for Information Science and technology, 55(1):23-28.

Aymen Abu-Errub. 2014. Arabic text classification algorithm using tf-idf and chi square measurements. International Journal of Computer Applications, 93(6).

S Al-Harbi, A Almuhareb, A Al-Thubaity, MS Khorsheed, and A Al-Rajeh. 2008. Automatic Arabic text classification. Journes internationals, France, pp. 77–83.

Ricardo Baeza-Yates, Berthier Ribeiro-Neto, et al. 1999. Modern information retrieval, volume 463. ACM press New York.

Joachim Diederich, Jorg Kindermann, Edda Leopold,and Gerhard Paass. 2003. Authorship attribution with support vector machines. Applied intelligence, 19(1-2):109-123.

Alaa El-Halees. 2007. Arabic text classification using maximum entropy. The Islamic University Journal(Series of Natural Studies and Engineering), 15(1):157-167.

Mohamed El Kourdi, Amine Bensaid, and Tajje-eddine Rachidi. 2004. Automatic arabic document categorization based on the Naive bayes algorithm. In Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages, pages 51-58. Association for Computational Linguistics.

Tarek F Gharib, Mena B Habib, and Zaki T Fayed. 2009. Arabic text classification using support vector machines. International Journal of Computer Applications, 16(4):192-199.

Geoffrey E Hinton. 2002. Training products of experts by minimizing Contrastive Divergence. Neural computation, 14(8):1771-1800.

Ismail Hmeidi, Bilal Hawashin, and Eyas El-Qawasmeh. 2008. Performance of kNN and SVM classifiers on full word arabic articles. Advanced Engineering Informatics, 22(1):106–111.

Ghassan Kanaan, Riyad Al-Shalabi, Sameh Ghwanmeh, and Hamda Al-Maadeed. 2009. A comparison of text-classification techniques applied to arabic text. Journal of the American society for Information Science and Technology, 60(9):1836-1844.

Laila Khreisat. 2006. Arabic text classification using N-gram frequency statistics a comparative study. DMIN, 2006:78-82.

Nicolas Le Roux and Yoshua Bengio. 2008. Representational power of restricted boltzmann machines and deep belief networks. Neural computation, 20(6):1631-1649.

Yuval Marton, Nizar Habash, and Owen Rambow. 2013. Dependency parsing of modern standard arabic with lexical and inflectional features. Computational Linguistics, 39(1):161-194.

Adel Hamdan Mohammad, Omar Al-Momani, and Tariq Alwadan. 2016. Arabic text categorization using k-nearest neighbour, decision trees (c4.5) and rocchio classifier: A comparative study.

M Baran Pouyan, V Jindal, J Birjandtalab, and M Nourani. 2015. A two-stage clustering technique for automatic biaxial gating of flow cytometry data. In Proceedings of 2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pages 511-516.

Hassan Sawaf, Jorg Zaplo, and Hermann Ney. 2001. Statistical classification methods for arabic news articles. Natural Language Processing in ACL'2001, Toulouse, France.

Fabrizio Sebastiani. 2002. Machine learning in automated text categorization. ACM computing surveys (CSUR), 34(1):1-47.

Stijn Marinus Van Dongen. 2001. Graph clustering by flow simulation.

Yiming Yang and Jan O Pedersen. 1997. A comparative study on feature selection in text categorization. In ICML, volume 97, pages 412-420.