Automatic Text Tagging of Arabic News Articles Using Ensemble Deep Learning Models

Ashraf Elnagar, Omar Einea and **Ridhwan Al-Debsi**
Machine Learning and Arabic Language Processing Research Group
Dept. Computer Science
University of Sharjah
Sharjah, UAE
{ashraf,oeinea,raldebsi}@sharjah.ac.ae

Abstract

Automatic document categorization gains more importance in view of the plethora of textual documents added constantly on the web. Text categorization or classification is the process of automatically tagging a textual document with most relevant label. Text categorization for Arabic language become more challenging in the absence of large and free datasets. We propose new, rich and unbiased dataset for the single-label (SANAD) text classification, which is made freely available to the research community on Arabic computational linguistics. In contrast to the majority of the available categorization systems of Arabic text, we offer several deep learning classifiers. With deep learning, we eliminate the heavy pre-processing phase usually used to on the data. Our experimental results showed solid performance on SANAD corpus with a minimum accuracy of 93.43%, achieved by CGRU, and top performance of 95.81%, achieved by HANGRU. In pursuit of superior performance, we implemented an ensemble model to combine best deep learning models together in a majority-voting paradigm.

1 Introduction

As a result of the rise of the Internet and Web 2.0, unimaginable amount of data is constantly on the rise, which is produced by several sources including social media users. The presence of such unstructured data makes a great resource for data processing and management in order to extract useful information. One important task is text classification and clustering, which is a field of research that gained much momentum in the last few years. The recent advances in machine learning paved the road for proposing successful text categorization systems.

The terms text categorization and text classification are used interchangeably to indicate the process of predicting predefined categories or domains to a given document. The automated categorization process may report the most relevant single category or multiple close ones (Figure 1). For the huge amount of available documents (or text) on the internet, manual classification by domain experts becomes ineffective and unfeasible. Therefore, automated classifiers had become not only an alternative but a necessity utilizing machine learning algorithms. However, the unstructured nature of the textual documents necessitates the need of machine learning algorithms to represent the data in a compatible format such as using numeric vectors. Text categorization is a key prerequisite to several evolving applications in different areas such as language (and dialects) identification (Lulu and Elnagar, 2018), sentiment analysis (Elnagar and Einea, 2016; Elnagar et al., 2018b,a), genre classification (Onan, 2018), and spam filtering (Li et al., 2018) to list few.

Text categorization is well studied in several languages and in particular the English language. Despite of the importance of Arabic language being the fourth used language on the Internet and 6th official language reported by United Nations ((Eldos, 2003)), few research attempts are reported on the Arabic language text classification as detailed in the next section. According to Wikipedia, as of 2018, there are 25 independent nations where Arabic is an official language and the number of Arabic speakers reach 380 million. With the rise of Arabic data on the internet, the need for an effective and robust automated classification system becomes a must. The research attempts at addressing this problem for Arabic text are limited to using shallow deep learning classifiers and were conducted on small and mostly unavailable datasets. As a result, we report the construction of a dataset for Arabic categorization tasks collected from news sources. The dataset is made free to use for the research community. In addition and unlike previous research works, we utilize deep learning models for investigating both single-label Arabic text categorization and provide comparative results of the different models.

We constructed a new corpus for the Arabic classification tasks, namely, SANAD (Single-label Arabic News Articles Dataset), (Einea et al., 2019). This corpus consists of more than one dataset. It is made available on Mendely[1]. It is our objective to make the dataset accessible for the research community.

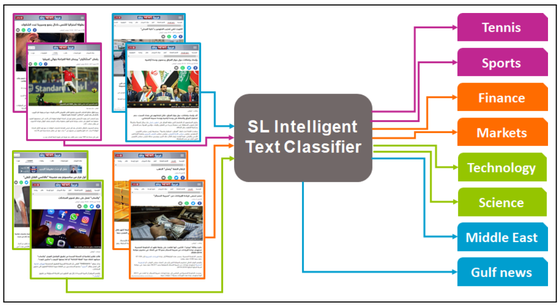

Figure 1: Single-label text classifier.

Several reported works proposed robust text classifiers but mostly designed for English text. As for Arabic, reported works are conducted on small datasets. Besides, the reported accuracies of such solutions have a big room for improvement. We implement nine robust deep neural network based classifiers that are tested on large datasets and yield high accuracy on single-label categorization tasks.

The remainder of this paper is organized as follows. In Section 2, we describe previous research work on Arabic text categorization. Next, we describe the datasets in detail in Section 3. In Section 4, we list the deep learning models implemented for the Arabic categorization task. In Section 5, we demonstrate the performance and improvement of our models over existing systems on SANAD as well as a recently reported benchmark dataset. Finally, we conclude our research in Section 6.

[1]http://dx.doi.org/10.17632/57zpx667y9.1

2 Literature Review

Numerous papers addressed the problem of automatic text categorization proposing different techniques and solutions. This is mainly true for the English language. Comprehensive surveys already exist and provide a thorough coverage of text categorization classifiers (Sebastiani, 2002; Aggarwal and Zhai, 2012; Korde and Mahender, 2012; Joachims, 2002). A relatively recent good survey on Arabic text categorization is available in (Hmeidi et al., 2014).

As our emphasis, in this work, is Arabic language, we pay more attention to research work on Arabic text categorization. The early work of Saad, (Saad, 2010), used several shallow learning supervised classifiers including Decision Tree, KNN, SVM, and Naïve Bayes. He studied the impact of pre-processing on text categorization results. For this purpose, he used the widely spread, but relatively small, BBC and CNN Arabic news datasets. Similarly, the effect of pre-processing of Arabic text in order to reduce the feature spaces are reported in (Duwairi et al., 2009; Al-Kabi et al., 2011; Yaseen and Hmeidi, 2014) in which the authors investigated the impact of stemming, light stemming, and synonyms-clustering on the features space reduction and accuracy. For the same purpose, Feature Subset Selection (FSS) metrics, (Mesleh, 2011), were used with SVM classifier to categorize text. Although the training time is reduced, accuracy deteriorates as well.

Furthermore, Maximum Entropy (ME) is used to classify news articles, (Sawaf et al., 2001). The work concluded that the Dice measures with N-gram produce better results than the Manhattan distance. Combining both ME and pre-processing is reported in (A, 2007). The author showed that the use of normalization and stop-words removal has enhanced F1-measure.

The use of Neural Networks (NN) for Arabic text categorization was first reported in (Umer and Khiyal, 2007) using Learning Vector Quantization (LVQ) classifier and self-organization Maps (SOM). Good accuracy results were reported while using a relatively small dataset. Similarly, the authors of (Harrag et al., 2011) showed that NN outperforms SVM after reducing the features space.

The majority of reported research on Arabic text classification used classical supervised machine learning classifiers such as NB (El Kourdi et al., 2004; Mesleh, 2007; Hadi et al., 2008; Joachims, 1998; Alsaleem, 2011; Khorsheed and Al-thubaity, 2013), SVM (Mesleh, 2007; Joachims, 1998; Alsaleem, 2011; Khorsheed and Al-thubaity, 2013), Rocchio (Joachims, 1998), KNN (Mesleh, 2007; Hadi et al., 2008; Joachims, 1998), and decision trees (Joachims, 1998; Khorsheed and Al-thubaity, 2013; Harrag et al., 2009). The results mostly conclude that SVM is reported as the top classifier for categorizing Arabic texts followed by NB and decision trees.

Different from the previous research works, El-Mahdaouy et al (El Mahdaouy et al., 2017) performed Arabic document classification using Word and document Embedding rather than relying on text pre-processing and word counting representation. It was shown that document Embedding outperformed text pre-processing techniques either by learning them using Doc2Vec or averaging word vectors. The results are in line with the conclusions reported by Baroni et al. (Baroni et al., 2014) which evaluated the use of word embedding against classical approaches that rely on pre-processing or word counting on an array of applications such as concept categorization on the English language. Besides, it has been shown that neural network based models are more robust when it comes for sensitivity to parameters settings.

In our work, we introduce new benchmark datasets for both single-label and multi-label Arabic text categorization. However, the datasets may serve the research community on Arabic computational linguistics working on other supervised learning tasks. Therefore, the datasets are publicly available. Moreover, we investigate the use of nine deep learning models to solve the single-label as well as the multi-label Arabic text categorization problem.

3 Dataset

We use three different datasets that we collected using web scraping (Python Selenium, Requests and BeautifulSoup or PowerShell), from three popular news websites (alarabiya.net, alkhaleej.ae and akhbarona.com). All datasets have the categories [Finance, Medical, Politics, Sports, Tech, Culture and Religion] except alarabiya.net; it does not have the last 2 categories. As these datasets were collected from news portals, the articles are expressed in modern standard Arabic, so there are no dialects involved. Since all datasets are tagged with single labels, we grouped them in one corpus called SANAD. We partitioned the datasets into training and testing sets, Table 1 details the number of articles and categories in each one of them.

Source	Categories	Train	Test
Alarabiya.net	5	22203	4075
Khaleej.ae	7	42000	3500
Akhbarona.com	7	42000	4900

Table 1: Number of articles in SANAD.

The scraped articles are cleaned by removing Latin alphabet and punctuation marks. In the sequel, we describe each one of the 3 datasets that make SANAD:

3.0.1 alarabiya.net

All scraped articles were initially grouped into 7 categories. However, 2 of the categories did not have much data (i.e., 'Culture' and 'Iran News') when compared with the rest of the categories. We merged 'Iran News' with the 'Politics' category and dropped the 'Culture' set. The articles collected are until early 2018. Figure 2 shows the distribution of the five resulting categories of this dataset.

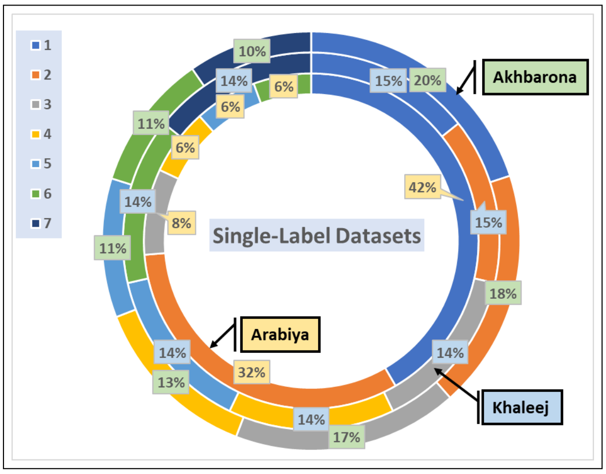

Figure 2: Distribution of categories in the proposed single-label datasets.

3.0.2 alkhaleej.ae

We collected around 1.2M (4GB) articles since 2008 until 2018. However, the tagging in the news portal was incomplete and vague. Therefore, we had to manually tag a reasonable amount of articles in each one of the aforementioned seven categories. This is a balanced dataset in which each category has 6.5k articles. the total size of the dataset is 45.5k articles. Figure 2 shows the balanced distribution of the 7 categories.

3.0.3 akhbarona.com

We scraped a large number of articles in the 7 categories. However, the 'Religion' category had half as much as other categories did. In order to increase the number, we scraped the remaining half of this category from a similar newspaper portal, which is `Alanba.com`. Figure 2 shows the resulting distribution of the seven categories of this dataset (Table 1).

4 Deep Leaning Models

- CNN The hierarchy of our CNN model consists of a dropout layer, followed by 3 CNN layers with kernel size of 5, and 128 filters, followed by global max-pooling with default values, and another dropout layer.

- RNN We used both GRU and LSTM models. The GRU model consists of 2 GRU layers. While our LSTM model consists of 1 LSTM layer. This selection has been determined by trying out different methods until we obtained the best accuracy. Both RNN layers are an improvement on the basic RNN layer to involve memory capabilities, where GRU has a memory, but LSTM was introduced to solve the Vanishing Gradient Problem (Hochreiter et al., 2001).

- BiRNN Both RNN models mentioned above were also wrapped around with a Bidirectional wrapper, giving us 2 more models; BiGRU and BiLSTM. Both models are composed of 1 BiRNN layer. The reason for implementing the bidirectional strategy is because of the nature of text, where each word is defined by the preceding and the proceeding words. Bidirectional wrappers allow the layers to go over the data in both directions, resulting in a vector that is 2 times as big as a uni-directional layer.

- Attention The attention mechanism was added only to the RNN models, as it was noted in (Raffel and Ellis, 2015) that it will solve the long term memory issues, hence it was applied to GRU and LSTM only. The attention models simply have an attention layer after the RNN model producing more models.

- CNN+RNN For our final 2 models, we used a combination of CNN and RNN layers to produce CRNNs (Convolution Recurrent Neural Networks). The hierarchy of the network consists of a dropout layer, followed by one CNN layer, one RNN layer, global max pooling, and another dropout layer.

5 Experimental Results and Discussion

5.1 Setup and Pre-processing

Our objective is to explore the success of using DNN models to classify Arabic news categories. We conducted several experiments involving categorization of Arabic text on different datasets. Our experiments involve single-label classification on our own three constructed datasets (`Arabiya`, `Khaleej`, and `Akhbarona`.

We split all datasets into 80% for training, 10% for cross-validation, and 10% for testing. We report the accuracy on testing datasets for each of the nine implemented deep learning models. It should be noted that embeddings are initialized at random for the input layer in all experiments. We chose Tensorflow and Keras frameworks for the implementation of all DNN models.

Simple text pre-processing is used to clean the dataset by filtering out non-Arabic content. This is particularly important when dealing with data collected from the web. Although Arabic character set is somehow unique, it is easy to eliminate non-Arabic characters. We further eliminate all diacritics, elongation (i.e., "جميـــل" is reduced to "جميل", punctuation marks, extra spaces, etc. Another widely adopted practice is to apply normalization on some Arabic characters. This involves replacing the letters "إ" , "أ" , and "آ" with letter "ا" , letter "ة" with "ه" , and letter "ي" with "ى" . In contrast with the majority of research works on Arabic computational linguistics, we argue that the normalization step is not required; we believe it can affect the contextual meaning for some words such as "فأر" and "فار" or "كرة" and "كره" . This is clear when producing word embedding models. As a result, we did not normalize the Arabic text.

5.2 Single-label Text Classification

We implemented 9 DNN models. Namely, 4 RNN models (GRU, BiGRU, LSTM, BiLSTM), 2 attention models (HANGRU, HANLSTM), and 3 CNN based models (CNN, CGRU, CLSTM). We trained the 9 models on each of the 3 training datasets of SANAD. Then, we tested the resulting trained model on each of the 3 datasets. For example, we trained the BIGRU model on Arabiya training dataset and tested it on Arabiya testing dataset, Khaleej testing dataset, and Akhbarona testing dataset. The resulting accuracy scores of this comprehensive testing is depicted in Figures 4 and 3 for each of the 9 DNN models.

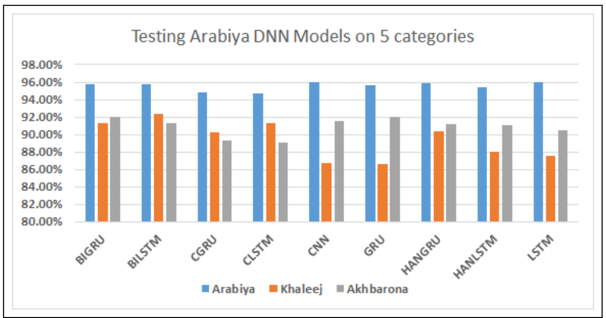

Figure 3: Performance evaluation of the 9 models on the 'Arabiya' datasets.

Figure 3 summarizes the accuracy results on our three constructed datasets. In the first experiment, we trained the nine DNN models on Arabiya dataset and then tested the models on Arabiya, Khaleej, and Akhbarona datasets on five categories. When testing on Arabiya dataset, six models out of the nine produced close results between 95.63% and 96.05% (CNN), one model (HANLSTM) reported 95.63, which is around average, and two models, CLSTM and CGRU, performed below average with accuracy scores of 94.67% and 94.87%, respectively. We further tested the Arabiya-trained model on totally different testing data from Khaleej and Akhbarona datasets. On Khaleej testing dataset, the best and worst results are reported by BiLSTM model with accuracy of 92.40% and GRU model with accuracy of 86.64%. As for Akhbarona test dataset, the best and worst results are reported by GRU model with accuracy of 92.00% and CLSTM model with accuracy of 89.05%. In the second experiment (Figure 4), we trained the nine DNN models on Khaleej dataset and then tested the models on Arabiya, Khaleej, and Akhbarona datasets on seven categories. The

Table 2: Performance of the 9 DNN models on the datasets (AR-5, KH-7, and AB-7. Best and worst performing DL model is shown in Bold font for each dataset.

	AR-5	KH-7	AB-7
BIGRU	95.78%	95.00%	92.94%
BILSTM	95.75%	93.91%	93.53%
CGRU	94.87%	94.23%	**91.18%**
CLSTM	**94.67%**	94.57%	92.55%
CNN	**96.05%**	95.89%	93.94%
GRU	95.63%	**93.86%**	93.37%
HANGRU	95.85%	**96.94%**	**94.63%**
HANLSTM	95.36%	95.49%	94.08%
LSTM	95.95%	95.23%	93.26%

results on Khaleej test dataset ranged between 93.85% and 96.94%. Whereas the results on the other two test datasets ranged between 75.04% and 87.12% for Arabiya and 66.38% and 76.40% for Akhbarona. In the third experiment (Figure 4), we trained the nine DNN models on Akhbarona dataset and tested against the three datasets. The results ranged between 78.43% and 89.79% for Arabiya; and 70.14% and 80.46% for Khaleej; and 91.18% and 94.63% for Akhbarona. The results of these experiments show that Arabiya-training model is the best one to use for single-label classification of Arabic news articles.

The performance of the DNN models vary in the above set of experiments. While some DNN models produce above average results, few others are trailing behind. Figure 5 reflects the level of performance of each model in the experiments. For example, BiLSTM model yielded accuracy above average in eight of the nine experiments. Similarly, both HANGRU and BiGRU models were successfully producing solid results around or above average. However, GRU performed poorly compared to the rest.

Table 2 depicts the results on SANAD datasets. namely, Arabiya with 5 categories (AR), Khaleej with 7 categories (KH-7), and Akhbarona with 7 categories (AB-7).

5.3 Ensemble Models

To further enhance the accuracy results of the deep learning models, we employed the ensemble concept to produce better classifiers. Ensemble modeling is the process of combining more than one model together while producing a single accuracy

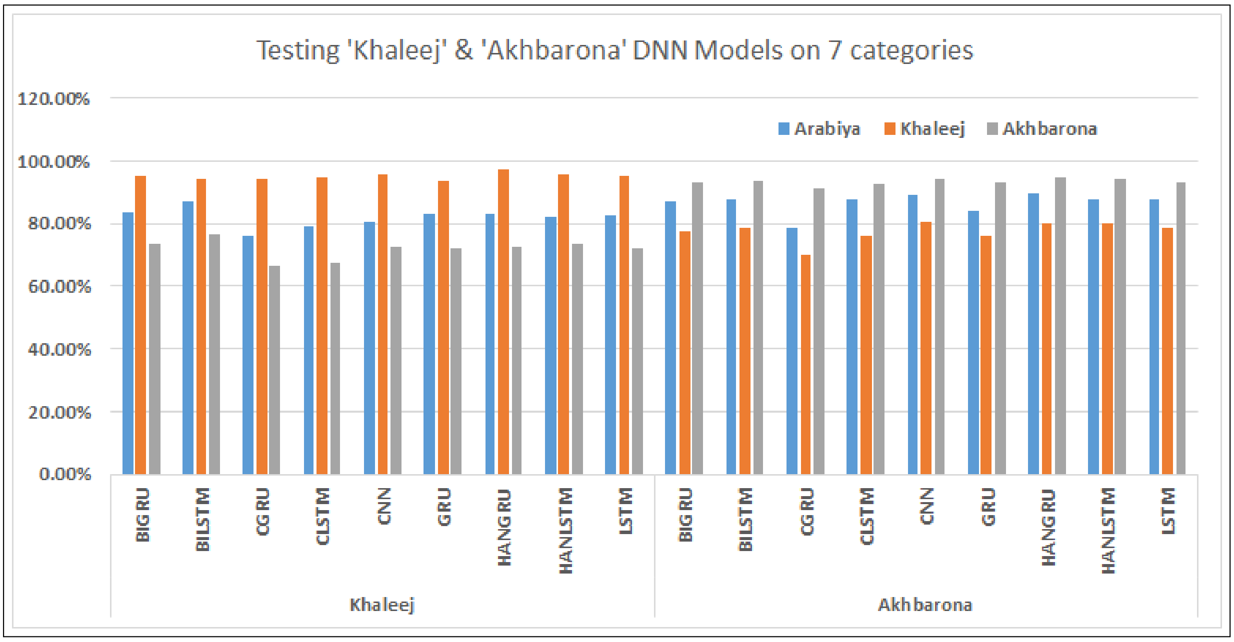

Figure 4: Performance evaluation of the 9 models on the datasets: Khaleej and Akhbarona.

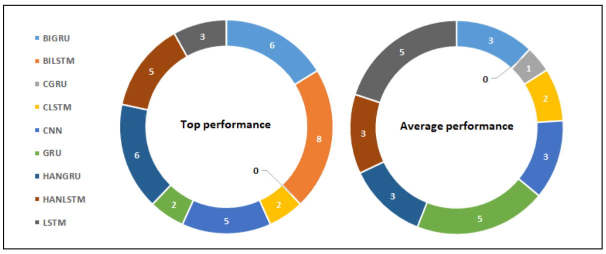

Figure 5: Top and average performing deep learning models in all experiments.

score. We use the majority voting principle to compute such score. By combining different models, we anticipate eliminating drawbacks of some models such as biases and high variability of data.

We performed a greedy ensemble on all combinations of DNN models. We solicited models that produce higher accuracy than the best single model reported above. As expected, a combination of two or more models outperformed the top individual model's accuracy. Although the number of generated ensemble models reached 459 models in some cases, the improvement in accuracy scores did not exceed 2.1%. This is was achieved when testing Khaleej models against Akhbarona dataset. On the other hand, no single ensemble model beat the top single model of testing Khaleej models on Khaleej dataset. It is worth noting that the impact is little because the reported accuracy scores of individual models are already high. Figure 6 compares top ensemble models with top individual model for all nine tests.

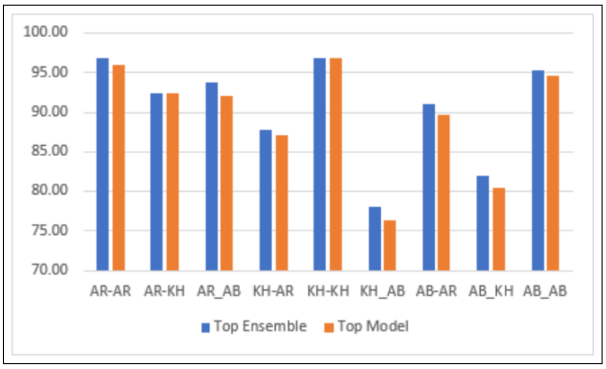

Figure 6: Top ensemble model vs. top individual model.

We observed that some of the DNN models had more contribution in the successful selected ensemble models than others. The model that had the most contribution is HANGRU, which appeared in 7 experiments out of the 9 ones; major contributor to the top ensemble models. CNN appeared 6 times. However, the BiGRU model is the least contributor (only once). Figure 7 shows the contribution percentages of each model in the top ensemble models.

6 Conclusion

In this work, we described a new large corpus for single-label Arabic text categorization tasks as a contribution to the research community on Arabic computational linguistics. SANAD is collected

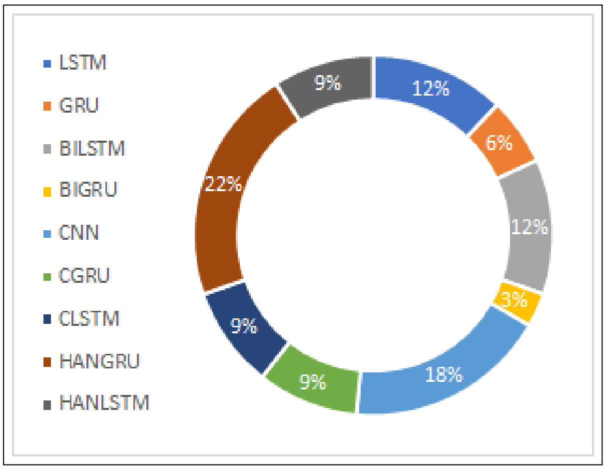

Figure 7: Contributions of DNN models to the top ensemble models.

from annotated Arabic news articles, and consists of 3 datasets; 2 (Arabiya and Akhbarona) are imbalanced while Khaleej dataset is a balanced one. The total number of Arabic articles amount to 200k, which makes it the largest freely available benchmark. The articles are classified into a maximum of seven categories.

We further implemented a variety of deep learning Arabic text classifiers and tested them thoroughly on SANAD corpus. Our treatment is different from existing Arabic single-label text systems that adopt standard machine learning classifiers with heavy pre-processing phase to prepare the data. Besides, we eliminated the heavy pre-processing requirements. Our experimental results showed that DNN models performed very well on SANAD corpus with a minimum accuracy of 93.43%, achieved by CGRU, and top performance of 95.81%, achieved by HANGRU. Furthermore, we introduced ensemble modeling to boost the performance, which resulted in enhancing the results in 8 experiments out of the 9 ones.

References

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

Charu C. Aggarwal and Cheng Xiang Zhai. 2012. *Mining Text Data*. Springer Publishing Company, Incorporated.

Mohammed N. Al-Kabi, Qasem A. Al-Radaideh, and Khalid W. Akkawi. 2011. Benchmarking and assessing the performance of arabic stemmers. *Journal of Information Science*, 37(2):111–119.

Saleh Alsaleem. 2011. Automated arabic text categorization using svm and nb. *International Arab Journal of eTechnology*.

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 238–247. Association for Computational Linguistics.

Rehab Duwairi, Mohammad Nayef Al-Refai, and Natheer Khasawneh. 2009. Feature reduction techniques for arabic text categorization. *Journal of the American Society for Information Science and Technology*, 60(11):2347–2352.

Omar Einea, Ashraf Elnagar, and Ridhwan Al Debsi. 2019. Sanad: Single-label arabic news articles dataset for automatic text categorization. *Data in Brief*, page 104076.

Mohamed El Kourdi, Amine Bensaid, and Tajje-eddine Rachidi. 2004. Automatic arabic document categorization based on the naÏve bayes algorithm. In *Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages*, Semitic '04, pages 51–58, Stroudsburg, PA, USA. Association for Computational Linguistics.

Abdelkader El Mahdaouy, Eric Gaussier, and Saïd Ouatik El Alaoui. 2017. Arabic text classification based on word and document embeddings. In *Proceedings of the International Conference on Advanced Intelligent Systems and Informatics 2016*, pages 32–41, Cham. Springer International Publishing.

T.M. Eldos. 2003. Arabic text data mining: a root-based hierarchical indexing model. *International Journal of Modelling and Simulation*, 23(3):158–166.

A. Elnagar and O. Einea. 2016. Brad 1.0: Book reviews in arabic dataset. In *2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA)*, pages 1–8.

Ashraf Elnagar, Yasmin S. Khalifa, and Anas Einea. 2018a. *Hotel Arabic-Reviews Dataset Construction for Sentiment Analysis Applications*, pages 35–52. Springer International Publishing.

Ashraf Elnagar, Leena Lulu, and Omar Einea. 2018b. An annotated huge dataset for standard and colloquial arabic reviews for subjective sentiment analysis. *Procedia Computer Science*, 142:182 – 189. Arabic Computational Linguistics.

Wael Hadi, S Al Hawari, Fadi Thabtah, and Jafar Ababneh. 2008. Naive bayesian and k-nearest neighbour to categorize arabic text data. In *The 22nd annual European Simulation and Modelling Conference (ESM'2008)*, pages 196–200.

F. Harrag, E. El-Qawasmeh, and P. Pichappan. 2009. Improving arabic text categorization using decision trees. In *2009 First International Conference on Networked Digital Technologies*, pages 110–115.

Fouzi Harrag, Eyas El-Qawasmah, and Abdul Malik S. Al-Salman. 2011. Stemming as a feature reduction technique for arabic text categorization. *2011 10th International Symposium on Programming and Systems*, pages 128–133.

Ismail Hmeidi, Mahmoud Al-Ayyoub, Nawaf Abdulla, Abdalrahman Almodawar, Raddad Abooraig, and Nizar A. Ahmed. 2014. Automatic arabic text categorization: A comprehensive comparative study. *Journal of Information Science*, 41:114–124.

Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, Jürgen Schmidhuber, et al. 2001. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies.

Thorsten Joachims. 1998. Text categorization with support vector machines: Learning with many relevant features. In *Machine Learning: ECML-98*, pages 137–142, Berlin, Heidelberg. Springer Berlin Heidelberg.

Thorsten Joachims. 2002. *Learning to Classify Text Using Support Vector Machines: Methods, Theory and Algorithms*. Kluwer Academic Publishers, Norwell, MA, USA.

Mohammad Khorsheed and Abdulmohsen Al-thubaity. 2013. Comparative evaluation of text classification techniques using a large diverse arabic dataset. *Language Resources and Evaluation*, 47:513–538.

SVandana Korde and C. Namrata Mahender. 2012. Text classification and classifiers: A survey. *IJAIA Journal*, 3(2):85–99.

Yuancheng Li, Xiangqian Nie, and Rong Huang. 2018. Web spam classification method based on deep belief networks. *Expert Systems with Applications*, 96:261 – 270.

Leena Lulu and Ashraf Elnagar. 2018. Automatic arabic dialect classification using deep learning models. *Procedia Computer Science*, 142:262 – 269. Arabic Computational Linguistics.

Abdelwadood Moh'd Mesleh. 2011. Feature sub-set selection metrics for arabic text classification. *Pattern Recogn. Lett.*, 32(14):1922–1929.

Abdelwadood Moh'd A Mesleh. 2007. Chi square feature extraction based svms arabic language text categorization system. *Journal of Computer Science*, 3(6):430–435. Exported from https://app.dimensions.ai on 2019/02/03.

Aytuğ Onan. 2018. An ensemble scheme based on language function analysis and feature engineering for text genre classification. *Journal of Information Science*, 44(1):28–47.

Colin Raffel and Daniel PW Ellis. 2015. Feed-forward networks with attention can solve some long-term memory problems. *arXiv preprint arXiv:1512.08756*.

Motaz Saad. 2010. The Impact of Text Preprocessing and Term Weighting on Arabic Text Classification. Master's thesis, Computer Engineering Dept., Islamic University of Gaza, Palestine.

Hassan Sawaf, Jörg Zaplo, and Hermann Ney. 2001. Statistical classification methods for arabic news articles. In *Arabic Natural Language Processing in ACL2001*, pages 1–6.

Fabrizio Sebastiani. 2002. Machine learning in automated text categorization. *ACM Comput. Surv.*, 34(1):1–47.

Muhammad Fahad Umer and M. Sikander Hayat Khiyal. 2007. Classification of textual documents using learning vector quantization. *Information Technology Journal*, 6:154–159.

Qussai Yaseen and Ismail Hmeidi. 2014. Extracting the roots of arabic words without removing affixes. *Journal of Information Science*, 40(3):376–385.