Arabic Sentiment Analysis Classification

Ahmed Abo Eitta[a], Karim Elezabawy[a], Sayed Omar[a],
Yousef Nafea[a] and Walid Gomaa[ab]

[a] *Cyber-Physical Systems Lab, Egypt-Japan University
of Science and Technology, Alexandria, Egypt*

[b] *Faculty of Engineering Alexandria University, Alexandria, Egypt*

{ahmed.aboeitta, karim.elezabawy, sayed.omar, yousef.nafea, walid.gomaa}@ejust.edu.eg

Abstract

Sentiment analysis is a popular and challenging Natural Language Processing(NLP) application, which helps to capture public opinions on a topic, product, or service. The research state in Arabic sentiment analysis falls behind other high-source languages such as English and Chinese [14]. In this paper, different pipelines with different approaches are done on the ASAD dataset to help analyze the dataset and obtain possible higher accuracy.

Arabic NLP; Sentiment Analysis; Deep Learning; Word Embeddings;

1 Introduction

Nowadays, Sentiment Analysis (SA) is attracting much attention from researchers as it is broadly investigated research areas. SA is known as opinion mining, it is one of natural language processing (NLP) applications that extract and analyse population opinions from the text extracted either from reviews from restaurants,hotels,...,etc or social media posts or any other text resources. SA is a needed element for decision makers, business leaders, political surveys, marketing,...,etc. All those are investing in the research for SA specially in marketing as this investment helps not only to keep their clients satisfied but also to further improve new products and services and attract prospective customers and generally SA has impact on on decision making and on several areas such as politics , economy, and tourism. Arabic sentiment analysis has received less focus due to lack of resources, corpora and lexicons in particular, in addition to the complexity of its morphological and syntactical systems, which incurs ambiguity and requires extensive processing. But now social media has offered a great and wide variety for data needed for doing the Arabic sentiment analysis.

Approaches used for sentiment analysis are either machine learning approaches or semantic approach. Semantic approach depends on semantic words extraction and popularity calculating using lexicon analysis. The machine learning approach which we are using in this paper mainly depends on supervised learning. In this paper we used is a twitter dataset from Arabic sentiment analysis competition represented by KAUST university and we used the machine learning approach with many experiments to get the best results with many combinations between classifiers and features extractors. Our pipeline to build the SA model is a simple pre-processing on the dataset then tokenizing the tweets and used the tokens as an input for a feature extractor like TFID or Aravec or pretrained Word2vec to get word vectors then passing them to a classifier like SVM or LR or LSTM, we get all the possible combinations between the mentioned classifiers and the feature extractors to try to get the best possible results.

2 Related Work

There has been many attempts to tackle down the problem of Arabic sentiment analysis. A recent publication by Al-Ayyoub et al. [3] presents a comprehensive overview of the works done so far about Arabic SA. Farra et al. [13] proposed two approaches for determining Arabic sentence sentiment: (i) a grammatical approach depending on sentence structure and (ii) a lexicon-based approach where words of known sentiment and their frequencies are taken into account. Al-Smadi et al. [4] proposed an aspect-based approach for determining sentiments of Arabic hotel reviews. They used SVM and recurrent neural networks (RNNs). Another work by Shoeb et al. [9] the authors applied SA on tweets by using Naive Bayes (NB) and KNN. The results were relatively good.

In SemEval -International Workshop on Semantic Evaluation - 2017, an Arabic sentiment analysis task was included. El-Beltagy et al. [11] were ranked first in this task. They used a set of hand-engineered and lexicon-based features. The classifier they used was a complement Naive Bayes classifier. The second rank in the same task was by Jabreel et al. [16] who proposed a set of features that are based on bag of words (BoW) model in addition to some features extracted from word embeddings. They used SVM as their classifier.

AraBERT a pre-trained transformer-based BERT model specifically for the Arabic language that uses a Masked Language Modeling (MTM) by Antoun et al. [7]. Their performance was measured by comparing theirs to multilingual BERT from Google and other state-of-the-art approaches. Abu Farha et al. [2] proposed a hybrid model called Mazajak, where CNNs were used for feature extraction, and LSTMs were used for sequence and context understanding. AraVec is a pretrained word embeddings that were trained by Soliman et al. [19] using word2vec framework on a Twitter and Wikipedia corpus. It is the largest dataset used for building Arabic word embeddings using around 67M tweets of different variants (skipgram/ cbow).

CAMel tools by Obeid et al. [20] was implemented by fine-tuning BERT and AraBERT [7] for the sake of Arabic SA. The fine-tuning was made by adding a fully connected linear layer to the last hidden state accompanied by a softmax activation function. Al-Ayyoub et al. [8] created a large lexicon of Arabic words collected from news articles. They built an SA system based on this lexicon and tested it on Twitter collected data.

There have been trials to take advantage of already existing powerful English sentiment analysis systems to assign sentiment analysis to Arabic text. This technique is done by translating Arabic text into English, and using English sentiment resources to assign sentiment polarity to the text. The authors in [15] translated Arabic text, normalized and structured it, and then attempted to find a match in SenticNet, an English sentiment lexicon.

3 Dataset

In the past few years, Arabic sentiment analysis gained much interest and attention which resulted in the publication of a number of Arabic datasets. Those datasets cover several sources which include reviews, newspapers, Wikipedia, and different content from social media platforms e.g Twitter.[12, 1, 6] Work done in this paper is based on A Twitter-based benchmark Arabic Sentiment Analysis Dataset (ASAD) which is a public dataset intended to accelerate research in Arabic NLP generally, and Arabic sentiment classification specially.[5]

ASAD tweets were randomly selected from a pool of tweets that have been collected using the Twitter public streaming API during the period of May 2012 and April 2020. After ASAD content was analyzed, it was found out that around 72% of the tweets located from Saudi Arabia, 13% from Egypt, 7% from Kuwait, 3% from the United States,and 3% from the United Arab Emirates. For dates, around 69% of the tweets were selected from the year 2020 (January 2020 until April 2020), 30% were tweeted in 2019 and the remaining 1% selected in the period between 2012 and

2018. For dialects, 36% of the tweets are written in Modern Arabic, 31% of the tweets used the Khaleeji dialect, 22% used the Hijazi dialect, and the rest 10% tweeted in Egyptian dialect as shown in figure 1.[5]

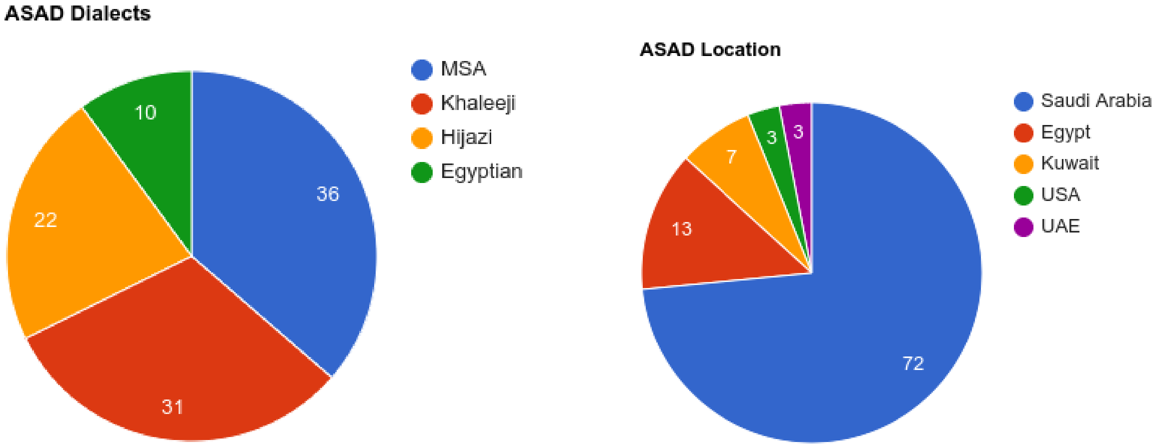

Figure 1: ASAD dialects and locations

ASAD consists of 95,000 annotated tweets, with three-class sentiments labels (positive, negative, and neutral). ASAD is divided into three folds: TRAINING, TEST1 and TEST2. The TRAINING set contains 55K labeled tweets, while TEST1 and TEST2 are composed of 20K tweets each. ASAD has a total of 15,215 positive tweets, 15,267 negative tweets and 64,518 neutral tweets as Shown in Table 1.[5]

	TRAINING		TEST1		TEST2		All	
	No. tweets	(%)	No. tweets	(%)	No. tweets	(%)	No. tweets	(%)
Positive	8821	0.16	3150	0.16	3244	0.16	15215	0.16
Negative	8820	0.16	3252	0.16	3195	0.16	15267	0.16
Neutral	37359	0.68	13598	0.68	13561	0.68	64518	0.68
Total	55000	1.00	20000	1.00	20000	1.00	95000	1.00

Table 1: Class distribution in data splits

4 Methodology

4.1 Preprocessing

The dataset is divided to 50K random tweets for training and 5k random tweets for testing then preprocessing is done to the dataset where tweets are being cleaned from unwanted characters like [”.”,””,” ”,..,etc] and other characters that won’t affect the meaning of the tweet to help in feature extraction process. Some cleaning is done automatically by the aravec model and other done by tokenizing the tweets first to extract every single word then scanning the tokenized tweets and removing all unwanted characters from the tokens. As we said we are doing experiments on many combinations so for the aravec model experiment we depend on the preprocessing that is done automatically by the model other experiments like the pretrained Word2vec model we do the preprocessing using the tokenizer.

4.2 Feature Extraction

Word Embeddings: a word representation allows words with similar sentiment to have a similar vector representation. The idea of word embeddings is representing individual words as real-valued vectors in a predefined vector space. Each word is mapped to one vector and the values of vector are learned in a way that is similar to a neural network. The term ”word embeddings” was first used by Bengio et al. [10] where they proposed a model based on obtaining values for word embeddings by training a neural language model. Mikolov’s et al. [17] work was a break through in word embeddings by the creation of Word2Vec that can be easily used and tuned to output embeddings. Mikolov’s et al. proposed two model architectures for representing word embeddings: continuous bag-of-words and skip-gram model. In the CBOW model, the surrounding words are given and are combined to predict the center word. While in Skip-gram model, it aims to predict the surrounding context words given the input center word.

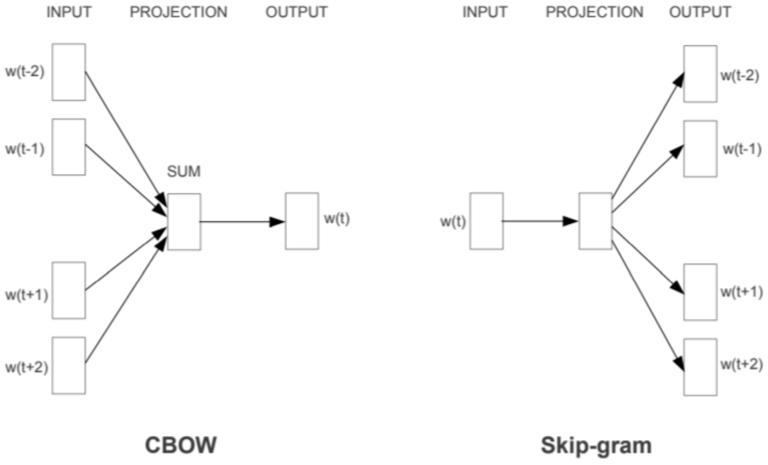

Figure 2: CBOW model vs Skip-gram model [18]

AraVec [19] provides six different word embedding models consisting of three text domains (Tweets, WWW and Wikipedia) and each one of them has two different model architectures; one using CBOW technique and another using the Skip-Gram technique.

TFIDF: a statistical measure that evaluates the relevance of a word to a document in a collection of documents. This is achieved by multiplying two metrics: the frequency of word in a document and the inverse document frequency of the word across a set of documents. TF-IDF is widely used in automated text analysis, and is very useful for qualifying words in natural language processing (NLP).

4.3 Classifiers

Logistic Regression (LR): The second algorithm for classification we used is Logistic regression. It belongs to the family of classifiers known as the exponential or log-linear classifiers. It works by extracting some set of weighted features from the input, taking logs, and combining them linearly. Technically, logistic regression refers to binary classification where the outcome can have only two values, and multinomial logistic regression is used when classifying into three or more classes. In our case, we used multinomial logistic regression as the output is divided into three classes positive, neutral, negative . We use the shorthand logistic regression even when we are talking about more than two classes throughout the paper.

Support Vector Machine (SVM): SVM is a well known supervised ma-chine learning paradigm that depends on finding a decisionboundary that maximizes the separation between the twoclasses. We use the output of the extracted feature vectorfrom the first dense layer of the CNN model, which is a vectorof dimension2048as the input to the SVM, fit the SVM tothe training data and test its efficacy for the given task.

Long Short-Term Memory (LSTM): LSTM is a recurrent neural network (RNN) architecture that remembers values over arbitrary intervals. LSTM is well-suited to classify, process and predict time series which is perfect for any application that has a sequence. And since the word meaning depends on both the following and the preceding words, LSTM was a great choice for the sentiment analysis task.

5 Experiment

In our experiment, we had different pipelines with each one implementing a different approach, these approaches can be classified as following: traditional methods, deeplearning based methods, and sentiment classification tools.
Traditional Methods: We Applied and TFID as a feature extractor, also used Aravec to generate word embeddings and used it as our features, the features were then classified using logistic regression and SVM models.
In order to handle the imbalance of the datasaet, we undersampled the neutral class as a solution, also we implemented a heretical classifier that's explained in details later
50k of the tweets were randomly sampled as the training set, while the other 5k were used for testing, this method is repeated 10 times and the average is recorded to make sure that the results is not produced based on a lucky shuffle

- **TFIDF + SVM** TF-IDF measures how important a particular word is with respect to a document and the entire corpus. The text features were extracted by Tf-Idf, then we applied the logistic SVM classifier with rbf kernel.

- **TFIDF + LR** Similar to the previous model, we used Tf-Idf as a feature extractor, then used logistic regression as a classification tool.

- **Aravec + LR** In this approach, the text features were extracted using aravec unigram twitter cbow model with vector length of 300 to generate one vector for each tweet, where the vector represents the average value of the vectors of each word in the tweet. Then a logistic regression classifier is used.

- **Aravec + SVM(full)** Similar to the previous method, the embeddings are generated, but this time a SVM classifier with rbf kernel is used for classification. this approach uses the full 45k tweet for training.

- **Aravec + SVM(undersampling)** This approach is similar to the previous approach, but the neutral class is undersampled so the number of neutral tweets in the training set be equal to positive and negative tweets, when undersampling was used, the training set was reduced to about 22k tweet on average. Then the same training technique is applied using SVM classifier with rbf kernel.

- **Aravec + SVM(hierarchical)** This approach was also done to target the imbalance of the data, in which the training set was first used to train a Neutral vs All classifier, then using only the positive and negative tweets from the training set another model is trained to distinguish between positive and negative tweets, during testing, the first model would first classify if the tweet is neutral or not, if it's not neutral then the second model would classify it as positive or negative.

for these tests, scikit-sklearn library[21] is used for the classifiers and the evaluation metrics, also for the Tf-Idf feature extractors. For the word embeddings we used the aravec models[19]

Deeplearning based methods: for this part, only one pipeline was implemented in this paper. First the data was preprocessed using a tokenizer, then a word2vec model was used to generate word embeddings out of the preprocessed tweets, then the data was fed to a 2 layers long short-term memory(LSTM) for classification, this model would be refered as **Word2vec + LSTM** further in the results section.

Sentiment classification tools: In this approach, we tried some pre-trained tools for sentiment classification. All the 55k labeled tweets were used in this approach as the tools don't accept fine tuning, the 2 tools that are used are **Mazajak**[2] and **CAMeL Tools** [20]

There are two metrics for our experiments, the primary metric is the average recall of the 3 classes as shown in equation 1. The secondary metric is the average F1 score of both positive and negative classes as shown in equation 2. Both of the metrics are good for imbalanced datasets, but in our case only the first metric will be used for evaluating the models, the secondary metric can help analyze the results more.

$$AvgRec = \frac{1}{3}(R^P + R^N + R^U) \tag{1}$$

$$F1^{PN} = \frac{1}{2}(F1^P + F1^N) \tag{2}$$

Table 2 shows how each model does against the evaluation metrics, as expected, the pre-trained tools had the least scores given that they were trained on other datasets that are not similar in distribution. We also observe that there is a noticable difference between models trained on Tf-Idf feature, and models trained on word embeddings, this is most likely because the word embeddings capture more information of the tweet. Also, SVM as a classifier has always produced better results than LR, with a significant difference when using aravec embeddings. It's also noticeable that the approaches used for handling the imbalance of the dataset didn't do better in terms of *AvgRec*, the hierarchical method almost produced similar results to the best method, but the under sampling method had much lower *AvgRec* this is believed be because under sampling heavily affected the size of the dataset, as it got down to 22k from 45k which is less than half. For the hierarchical approach, we notice that it helped increase the recall of the neutral class R^U using the neutral vs all model which was expected, but the positive and negative recall weren't as high as with other models which is believed to be because there is not enough positve and negative tweets to train the second model effectively.

Table 2: Evaluation of the models

Model	R^P	R^N	R^U	$AvgRec$	$F1^{PN}$
TFIDF + SVM	32.4	20.6	97.0	50.0	38.9
TFIDF + LR	44.5	40.4	89.7	58.2	49.8
Aravec + LR	61.9	44.7	92.0	66.2	60.5
Aravec + SVM(full)	80.5	73.6	82.6	78.9	63.8
Aravec + SVM(undersampling)	61.0	47.5	92.6	67.0	64.5
Aravec + SVM(hierarchical)	76.4	66.6	84.3	75.7	64.4
Word2vec + LSTM	62.8	23.2	96.4	60.8	53.4
Mazajak	32.3	30.8	86.2	49.8	42.9
CAMeL Tools	33.1	38.9	90.9	54.3	47.95

6 Conclusion

In this paper, we've tackled the problem of Arabic NLP, and why it's usefull to have a model that can classify the sentiment of the person based on text. We've also tried a combination of feature extractors and classifiers to come up with a model that achieves very high results based on the ASAD dataset, which is a benchmark for arabic sentiment analysis. The results showed that models that used Aravec word embeddings as features achieved the best results of all the models, Also we've applied methods to handle the imbalance in the dataset but these methods didn't achieve higher results than the standard ones, this was because the training set had low numbers of positive and negative tweets. for our futrue work, we intend to increase the number of positive and negative tweets in the training set using other datasets, but the new training data should follow the distribution of the original dataset in terms of the percentage of each dialect, this would help deal with the problem of imbalance. another approach would be making a model for each dialect, this would require a model with high accuracy to detect the dialect of the tweet, and multiple datasets from each dialect to be able to build a robust model for each dialect.

References

[1] M. Abdul-Mageed and M. Diab. SANA: A Large Scale Multi-Genre, Multi-Dialect Lexicon for Arabic Subjectivity and Sentiment Analysis. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*

(LREC'14), Reykjavik, Iceland, May 2014. European Language Resources Association (ELRA).

[2] I. Abu Farha and W. Magdy. Mazajak: An online Arabic sentiment analyser. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 192–198, Florence, Italy, Aug. 2019. Association for Computational Linguistics.

[3] M. Al-Ayyoub, A. Khamaiseh, Y. Jararweh, and M. Al-Kabi. A comprehensive survey of arabic sentiment analysis. *Information Processing Management*, 56, 09 2018.

[4] M. Al-Smadi, O. Qawasmeh, M. Al-Ayyoub, Y. Jararweh, and B. Gupta. Deep recurrent neural network vs. support vector machine for aspect-based sentiment analysis of arabic hotels' reviews. *J. Comput. Sci.*, 27:386–393, 2018.

[5] B. Alharbi, H. Alamro, M. Alshehri, Z. Khayyat, M. Kalkatawi, I. I. Jaber, and X. Zhang. ASAD: A Twitter-based Benchmark Arabic Sentiment Analysis Dataset. Nov. 2020.

[6] M. Aly and A. Atiya. LABR: A Large Scale Arabic Book Reviews Dataset. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 494–498, Sofia, Bulgaria, Aug. 2013. Association for Computational Linguistics.

[7] W. Antoun, F. Baly, and H. Hajj. AraBERT: Transformer-based model for Arabic language understanding. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 9–15, Marseille, France, May 2020. European Language Resource Association.

[8] M. A. Ayyoub, S. B. Essa, and I. Alsmadi. Lexicon-based sentiment analysis of Arabic tweets. *International Journal of Social Network Mining*, 2(2):101, 2015.

[9] N. Azam, B. Tahir, and A. Mehmood. Sentiment and emotion analysis of text: A survey on approaches and resources. 02 2020.

[10] Y. Bengio, H. Schwenk, J.-S. Senécal, F. Morin, and J.-L. Gauvain. *Neural Probabilistic Language Models*, pages 137–186. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[11] S. R. El-Beltagy, M. E. Kalamawy, and A. B. Soliman. Niletmrg at semeval-2017 task 4: Arabic sentiment analysis, 2017.

[12] H. ElSahar and S. R. El-Beltagy. Building Large Arabic Multi-domain Resources for Sentiment Analysis. In A. Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing*, Lecture Notes in Computer Science, pages 23–34, Cham, 2015. Springer International Publishing.

[13] N. Farra and K. McKeown. SMARTies: Sentiment models for Arabic target entities. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1002–1013, Valencia, Spain, Apr. 2017. Association for Computational Linguistics.

[14] I. Guellil, H. Saâdane, F. Azouaou, B. Gueni, and D. Nouvel. Arabic natural language processing: An overview. *Journal of King Saud University - Computer and Information Sciences*, Feb. 2019.

[15] H. Hassan, H. Bakr, and I. Ziedan. A framework for arabic concept-level sentiment analysis using senticnet. *International Journal of Electrical and Computer Engineering*, 8:4015–4022, 10 2018.

[16] M. Jabreel and A. Moreno. SiTAKA at SemEval-2017 task 4: Sentiment analysis in Twitter based on a rich set of features. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 694–699, Vancouver, Canada, Aug. 2017. Association for Computational Linguistics.

[17] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space, 2013.

[18] T. Mikolov, Q. V. Le, and I. Sutskever. Exploiting similarities among languages for machine translation, 2013.

[19] A. B. Mohammad, K. Eissa, and S. El-Beltagy. Aravec: A set of arabic word embedding models for use in arabic nlp. *Procedia Computer Science*, 117:256–265, 11 2017.

[20] O. Obeid, N. Zalmout, S. Khalifa, D. Taji, M. Oudah, B. Alhafni, G. Inoue, F. Eryani, A. Erdmann, and N. Habash. CAMeL tools: An open source python toolkit for Arabic natural language processing. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 7022–7032, Marseille, France, May 2020. European Language Resources Association.

[21] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.