BFCAI at SemEval-2022 Task 6: Multi-Layer Perceptron for Sarcasm Detection in Arabic Texts

Nsrin Ashraf and **Fathy Elkazaz** and **Mohamed Taha** and **Hamada Nayel**
Department of Computer Science, Benha University, Benha, Egypt
{nisrien.ashraf19,fathy.elkazzaz}@fci.bu.edu.eg
{mohamed.taha, hamada.ali}@fci.bu.edu.eg
Tarek Elshishtawy
Department of Information System, Benha University, Benha, Egypt
t.shishtawy@fci.bu.edu.eg

Abstract

This paper describes the systems submitted to iSarcasm shared task. The aim of iSarcasm is to identify the sarcastic contents in Arabic and English text. Our team participated in iSarcasm for the Arabic language. A multi-Layer machine learning based model has been submitted for Arabic sarcasm detection. In this model, a vector space TF-IDF has been used as for feature representation. The submitted system is simple and does not need any external resources. The test results show encouraging results.

1 Introduction

Analyzing social media becomes a crucial task, due to the frequently usage of social media platforms. Sarcasm detection, the conflict of using the verbal meaning of a sentence and its intended meaning (CLIFT, 1999; Lee and Katz, 1998), is an important task. Sarcasm detection is a challenge, since sarcastic contents are used to express the opposing of what is being said. Recently sarcasm detection has been studied from a computational perspective as one of classification problems that separates sarcastic from non-sarcastic contents(Reyes et al., 2013; Nayel et al., 2021).

Arabic is an important natural language having an extensive number of speakers. The research in Natural Language Processing (NLP) for Arabic is continually increasing. However, there is still a need to handle the complexity of NLP tasks in Arabic. This complexity arises from various aspects, such as orthography, morphology, dialects, short vowels, and word order. Sarcasm detection in Arabic is a particularly challenging task (Alayba et al., 2018).

In this paper, we describe the system submitted to the iSarcasm detection shared task(Abu Farha et al., 2022). The shared task aims at detecting the sarcasm contents in Arabic tweets. In this work, a machine learning framework has been developed and various machine learning algorithms have been implemented. Term Frequency-Inverse Document Frequency (TF-IDF) has been used as vector space model for tweet representation. The rest of this paper is organized as follows: in section 2, a background about sarcasm detection is given. Section 3 and section 4 overview the dataset and the system respectively. Experimental setup and results are given in section 5 and section 6 respectively. Finally, section 7 concludes the proposed work and suggests future work to be continued.

2 Background

The research work have been done on Arabic sarcasm detection were mainly focused on creating datasets and establish a baseline for each created dataset (Ghanem et al., 2020). Karoui et al. (2017) created a corpus of sarcastic Arabic tweets that are related to politics. Distant supervision has been used for the creation of corpus. The authors used keywords that are like sarcastic contents in Arabic to label the tweets as sarcastic tweets. They implemented different machine learning algorithms such as SVM, logistic regression, Naïve Bayes, and other classifiers on the developed corpus.

An ensemble classifier of XGBoost, random forest and fully connected neural networks has been designed by Khalifa and Hussein (2019). They extracted a set of features that consists of sentiment and statistical features, in addition to word n-grams, topic modelling features and word embeddings. Nayel et al. (2019) developed an ensemble-based system for irony detection in Arabic Tweets. A set of classification algorithms namely Random forests, multinomial Naïve Bayes, linear, and SVM classifiers have been used as base-classifiers. In (Nayel et al., 2021), sarcasm detection has been formulated as a binary classification problem and SVM has been implemented.

Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), pages 881 - 884

3 Dataset

A new data collection method has been introduced, where the sarcasm labels for texts are provided by the authors themselves. The author of each sarcastic text rephrased the text to convey the same intended message without using sarcasm. Leggitt and Gibbs (2000) defined a set categories of ironic speech namely; sarcasm, irony, satire, understatement, overstatement, and rhetorical question. Linguistic experts have been asked to further label each text into one of these categories. Each text in the Arabic dataset has the following information attached to it:

- a label specifying the text dialect;
- a label specifying the nature of sarcasm (sarcastic or non-sarcastic), provided by its author;
- a rephrase provided by its author that conveys the same message non-sarcastically.

4 System Overview

In this section, we review the main structure of the proposed model. The proposed system, as shown in figure 1, consists of three phases namely; preprocessing, feature extraction and training the classification algorithms. Then, the resulted model used to predict the unseen test data.

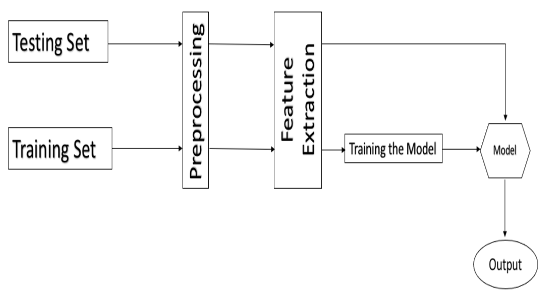

Figure 1: The structure of the proposed model

4.1 Preprocessing

The first stage of developing systems is preprocessing, where unwanted and uninformative piece of text has to be removed, it is also called text cleaning. We performed text cleaning by removing:

- special symbols, such as $\{+, -, =, \$, ...\}$;
- repeated characters such as ("*hhhhhhhh*" will be normalized to "*hh*");
- non-Arabic words, such as English characters or any other language;
- punctuations and Arabic diacritics.

4.2 Features Extraction

To prepare features to build classification model and before feeding the text into the classifier and after performing text cleaning, Term Frequency-Inverse Document Frequency (TF-IDF) technique was used to change over content to vectors and all the algorithms to investigate the best performing algorithm.

TF-IDF has been used to represent comments as vectors. If $< w_1, w_2, \ldots, w_k >$ are the tokenized words of a comment $\mathcal{T}_j$, the vector associated to the comment $\mathcal{T}_j$ will be represented as $<v_{j1}, v_{j2}, \ldots, v_{jk}>$ where v_{ji} is the weight of the token w_i in tweet $\mathcal{T}_j$ which is calculated as:-

$$v_{ji} = tf_{ji} * \log\left(\frac{N+1}{df_i+1}\right)$$

where tf_{ji} is the total number of occurrences of token w_i in the comment $\mathcal{T}_j$, df_i is the number of comments in which the token w_i occurs and N is the total number of comment.

4.3 Methodology

We explored various classification algorithms as well as ensemble approach by combining the output of these classifiers (also known as base classifiers) using hard voting. The base classifiers used in this work are listed below:

- Support Vector Machines (SVMs)
- Random Forest (RF)
- K-Nearest Neighbours (KNN)
- Multinomial Naïve Bayes (M-NB)
- Multi-Layer Perceptron (MLP)
- Stochastic Gradient Descent (SGD)
- AdaBoost Classifier
- Voting Classifier

Table 1: 5-fold Cross-Validation accuracy for all classifiers in the training set

Classifier	Accuracy	STD
SVM-Linear	81.0%	0.055
SVM-RBF	76.6%	0.003
MNB	76.6%	0.005
SGD	80.0%	0.045
MLP	83.6%	0.045
RF	75.8%	0.056
KNN	79.7%	0.058
AdaBoost	75.2%	0.052
Voting	80.4%	0.043

5 Experimental Setup

For feature extraction phase we used unigram model. For the purpose of training the model, we have used 5-fold cross-validation technique to adjust the parameters.

The Scikit-Learn library implementation of classification algorithms were used in the training phase. For SVM, two kernels have been tested: linear kernel and RBF with two parameters $\gamma = 2$ and $C = 1$. While, for SGD classifier the loss function used was Hinge and the maximum iteration was set at 10000 iterations.

The number of nodes in the hidden layer of MLP was set at 20, logistic function was used as activation function and Adam solver was used. The maximum number of decision trees in random forests is set at 300.

5.1 Evaluation Metrics

F1-score has been used to evaluate the performance of all submissions. F1-score is a harmonic mean of Precision (P) and Recall (R) and calculated as follow:

$$F\text{–}score = \frac{2 * P * R}{P + R}$$

F1-score for the sarcastic class (F1-sarcastic) has been used for final evaluation.

6 Results

The cross validation accuracy of all training classifiers for the training set is given in Table 1. It is clear that MLP gives the best accuracy with moderate Standard Deviation (STD) for the five folds while development phase, so we decided to submit the output of this classifier.

Table 2: Results of MLP classifier for the test set

Measure	Value	Rank
F-1 sarcastic	0.3746	14
F-score	0.6024	11
Precision (P)	0.5968	15
Recall (R)	0.6608	17
Accuracy	0.7329	8

Results for test set is given in Table 2. The reported results show that, while training MLP gives better accuracy among implemented machine learning classifiers. Also, it gives better rank in accuracy for the unlabelled test set. While in other metrics, the performance was not satisfied. This may resulted because of using accuracy metric while comparing different classifiers in development phase.

A good suggestion is to use different evaluation metrics while developing the system. In addition, using different word representation models such as word embeddings, which encompasses the semantic meaning of words could improve the performance. Complex models such as deep learning models is promising, but the challenge in such models is the availability of suitable resources.

7 Conclusion

In this work, a classical machine learning framework has been designed for sarcasm detection in Arabic tweets. The proposed framework reported reasonable results. The future work may include applying complex framework such as deep learning structure. In addition, word representation is very important factor that can be used in different manner such as word embeddings and transformers.

References

Ibrahim Abu Farha, Silviu Oprea, Steven Wilson, and Walid Magdy. 2022. SemEval-2022 Task 6: iSarcasmEval, Intended Sarcasm Detection in English and Arabic. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*. Association for Computational Linguistics.

Abdulaziz M. Alayba, Vasile Palade, Matthew England, and Rahat Iqbal. 2018. Improving sentiment analysis in arabic using word representation. *2018 IEEE 2nd International Workshop on Arabic and Derived Script Analysis and Recognition (ASAR)*.

REBECCA CLIFT. 1999. Irony in conversation. *Language in Society*, 28(4):523–553.

Bilal Ghanem, Jihen Karoui, Farah Benamara, Paolo Rosso, and Véronique Moriceau. 2020. Irony detection in a multilingual context. In *Advances in Information Retrieval*, pages 141–149, Cham. Springer International Publishing.

Jihen Karoui, Farah Banamara Zitoune, and Veronique Moriceau. 2017. Soukhria: Towards an irony detection system for arabic in social media. *Procedia Computer Science*, 117:161–168.

Muhammad Khalifa and Noura Hussein. 2019. Ensemble learning for irony detection in arabic tweets. In *Working Notes of FIRE 2019 - Forum for Information Retrieval Evaluation, Kolkata, India, December 12-15, 2019*, volume 2517 of *CEUR Workshop Proceedings*, pages 433–438. CEUR-WS.org.

Christopher J. Lee and Albert N. Katz. 1998. The differential role of ridicule in sarcasm and irony. *Metaphor and Symbol*, 13(1):1–15.

John S Leggitt and Raymond W Gibbs. 2000. Emotional reactions to verbal irony. *Discourse processes*, 29(1):1–24.

Hamada Nayel, Eslam Amer, Aya Allam, and Hanya Abdallah. 2021. Machine learning-based model for sentiment and sarcasm detection. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 386–389, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Hamada A. Nayel, Walaa Medhat, and Metwally Rashad. 2019. Benha@idat: Improving irony detection in arabic tweets using ensemble approach. In *Working Notes of FIRE 2019 - Forum for Information Retrieval Evaluation, Kolkata, India, December 12-15, 2019*, volume 2517 of *CEUR Workshop Proceedings*, pages 401–408. CEUR-WS.org.

Antonio Reyes, Paolo Rosso, and Tony Veale. 2013. A multidimensional approach for detecting irony in twitter. *Lang. Resour. Eval.*, 47(1):239–268.