BENHA@IDAT: Improving Irony Detection in Arabic Tweets using Ensemble Approach

Hamada A. Nayel[1] [0000−0002−2768−4639], Walaa Medhat[2], and Metwally Rashad[2]

[1] Department of Computer Science
Faculty of Computers and Artificial Intelligence, Benha University, Egypt
[2] Department of Information Systems
Faculty of Computers and Artificial Intelligence, Benha University, Egypt
{hamada.ali,walaa.medhat,metwally.rashad}@fci.bu.edu.eg

Abstract. This paper describes the methods and experiments that have been used in the development of our model submitted to Irony Detection for Arabic Tweets shared task. We submitted three runs based on our model using Support Vector Machines (SVM), Linear and Ensemble classifiers. Bag-of-Words with range of n-grams model have been used for feature extraction. Our submissions achieved accuracies of 82.1%, 81.6% and 81.1% for ensemble based, SVM and linear classifiers respectively.

Keywords: Irony Detection · Arabic NLP · Ensemble Based Classifiers · SVM. [3]

1 Introduction

Analyzing social media is an important research area due to the huge amount of information streaming from online social networking and microblogging platforms such as Twitter, Facebook and Instagram. One of the attractive tasks is irony detection which can be defined as the conflict of using the verbal meaning of a sentence and its intended meaning [15]. Twitter platform comprises of text communications with a high percentage of ironic messages. Devices and platforms monitoring the sentiment in Twitter messages are faced with the problem of wrong polarity classification of ironic messages [16].

Irony is studied by various disciplines, such as linguistics, philosophy, and psychology [18], but it is difficult to define it in formal terms. In computational linguistics, irony is often used as a concept of sarcasm, although some researchers differentiate between irony and sarcasm, considering that sarcasm tends to be harsher, humiliating, degrading and more aggressive [4, 8].

Irony detection is not a straightforward problem, since ironic statements are used to express the contrary of what is being said [20], therefore it is difficult to be solved by current systems. Being a creative form of language, there is no agreement in the literature on how verbal irony should be defined. Recently irony detection has been studied from a computational perspective as one of classification problem that separates ironic from non-ironic statements [16].

Arabic is an important natural language having a huge number of speakers. The research in Natural Language Processing (NLP) for Arabic is constantly increasing. However, there is still a need to handle the complexity of NLP tasks in Arabic. This complexity arises from different aspects, such as orthography, morphology, dialects, short vowels and word order [1]. Irony detection in Arabic is a challenging task. The following example : "أحنا ندعم خالد على وننتخب حسني مبارك " from Twitter illustrates how the author ironically employs a positive opinion word "ندعم" (which means support) towards Khaled Ali, the former presidential candidate of Egypt to express a negative opinion and take opposite action " ننتخب حسني مبارك " (which means then elect Hosni Mubarak).

In this paper we have developed a model for detecting ironic Arabic tweets using Machine Learning (ML) approach. The proposed model classifies a given tweet as either ironic or non-ironic. The paper is organized as follows: section 2 introduces the related work and section 3 contains a description of our model. Section 4 gives a brief overview about dataset and the performance evaluation metric is given in section 5. Results and future work are tackled in section 6 and section 7 respectively.

2 Related Work

Irony detection is a very challenging task that encountered a lot of development through the years. There are many research works that have been done on English language and fewer research on other natural languages specially the Arabic language. Here are some of the recent research works that contribute into the problem.

Reyes and Rosso [14] have focused on identifying the key components to detect irony in English customer reviews via computational point of view. Reviews that were posted by means of an online viral effect have been selected. They have designed a model with six categories namely, n-grams, POS ngrams, funny profiling, positive/negative profiling, affective profiling, and pleasantness profiling to represent irony from different linguistic layers. They achieved good results using three different classifiers in terms of accuracy and F-score.

In [19], some patterns associated to ironic statements in Brazilian Portuguese have been analyzed and implemented. A common ground between the author of the tweets and their audience is required in order to establish some background information on the text. Features like the city, time, and genre have been considered while analysing the patterns. Results showed that patterns related to symbolic language, such as laughter marks and emoticons are the best hints to irony and sarcasm. In addition, results illustrated that heavy punctuations are clues to ironic statements and patterns related to static expressions are bad search choices that have given low output results.

Barbieri and Saggion [2] casted the automatic detection of irony as a classification problem. They have proposed a model capable of detecting irony in the social network Twitter based on lexical features. Tweets that have hashtag irony and some other topics have been selected to create a linguistically motivated set of features. The features take into account frequency, written/spoken differences, sentiments, ambiguity, intensity, synonymy and structure. Results showed that their model outperforms bag-of-words approach across-domains.

emotIDM [5], a model for irony detection in Twitter, has been developed by formulation of the task as a classification problem. It was evaluated on a set of representative Twitter corpora that included samples of ironic and not ironic messages, which were different along various dimensions: size, balanced vs imbalance distribution, collection methodology and criteria. Results showed good performances in classification terms across all these dimensions. It performed better in cases of datasets with balanced distribution, where a self-tagging methodology has been applied.

Arabic tweets irony detection has been addressed in [7], by designing a binary classifier based system for irony detection in Arabic tweets. The classifier uses four groups of features, among which surface, sentiment, shifter, and contextual features. Tweets with irony hashtags in Arabic language have been collected. Results showed that state-of-the art features can be successfully applied to Arabic with accuracy of 72.76%.

In this paper, we developed a ML-based model for irony detection in Arabic tweets. Our model uses Term Frequency/Inverse Document Frequency (TF/IDF) with ranges of n-grams for extracting features. We registered for Irony Detection for Arabic Tweets (IDAT) shared task and have submitted three runs using different classification algorithms namely, Support Vector Machine (SVM), Linear classifier that uses Stochastic Gradient Descent (SGD) as an optimizer and Ensemble classifier.

3 Model Description

Given a set of tweets $\mathcal{T} = \{\mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_n\}$ and a set $\mathcal{C} = \{I, N\}$ of classes representing ***I**ronic* and ***N**on-ironic* tweets respectively, the task of detecting ironic tweets can be formalised as a simple binary classification problem that assigns one of two predefined classes of $\mathcal{C}$ to an unlabelled tweet $\mathcal{T}_k$.

The general structure of our model is shown in Fig. 1. The first step in our model is preprocessing which aims at cleaning the data and removing vain parts from it. The second step is feature extraction which is necessary for both training and testing purposes. The third step is training the classification algorithm. The following subsections give details of each step.

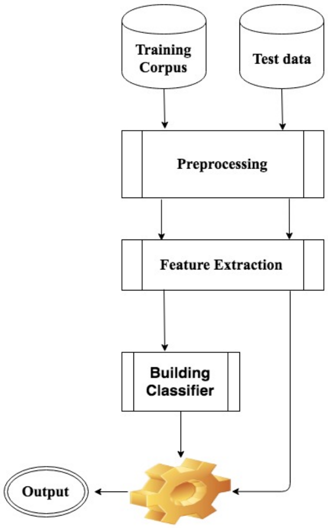

Fig. 1. The general structure of our model.

3.1 Preprocessing

Preprocessing is a key step in building models for Arabic language. In this step, each tweet $\mathcal{T}_k$ has been tokenized into a set of words or tokens to get n-gram bag of words. The following processes have been implemented to each tweet:

Punctuation Elimination We removed punctuation marks such as {'+', '_', '#', '$'.. }, which are increasing the dimension of feature space with redundant features. Example of redundancy, the following tokens {#صباح _العربية , صباح _العربية, صباح العربية } pronounced as "Sabah Al Arabiya" and means "morning of Al Arabiya"(Al Arabiya[4] is a news agency). Existence of "_" and "#" will add redundant features {#صباح _العربية , صباح _العربية} which affects the performance of irony detection.

Tweet Cleaning Twitter users usually do not follow the standard rules of the language especially Arabic language. A common manner of users is to repeat a specific letter in a word. Cleaning the tokens from this redundant letters helps in feature space reduction. In our experiments, the letter is assumed to be redundant if it is repeated more than two times. For example the words "ههههههه" ("*hahahah*" i.e. giggles) and "عااااااجل" (*i.e. "urgent"*) containing redundant letter and will be reduced to "هه" and "عااجل" respectively.

[4] https://www.alarabiya.net/

3.2 Features Extraction

TF/IDF with range of n-grams has been used to represent tweets as vectors. If $\lessdot w_1, w_2, \ldots, w_k \gtrdot$ are the tokenized words in a tweet $\mathcal{T}_j$, the vector associated to the tweet $\mathcal{T}_j$ will be represented as $\lessdot v_{j1}, v_{j2}, \ldots, v_{jk} \gtrdot$ 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 tweet $\mathcal{T}_j$, df_i is the number of tweets in which the token w_i occurs and N is the total number of tweets.

We used range of 2-grams model, i.e. unigram and bigram. For example sentence ”اجل اسقطت ساركوزى” (Yes I beat Sarkozy) has following set of features {”اجل”, ”اسقطت”, ”ساركوزى”, ”اجل اسقطت”, ”اسقطت ساركوزى”}.

3.3 Building Classifiers

Three classifiers have been trained for our model mainly SVM [17], Linear classifier [17] and Ensemble classifier. SVM is a binary classifier which has been used for different NLP tasks effectively [12, 11]. Linear classifier uses linear discriminant function.

Ensemble approach uses a set of classifiers as base classifiers and combines the output of these base classifiers to get the final output [10]. Ensemble approach has been implemented in different NLP tasks such as Native Language Identification [11] and Named Entity Recognition [10, 13]. Random Forests (RF) classifier is a supervised algorithm which involves multiple decision trees and each tree is built using independently sampled random vector [3]. Multinomial Bayes classifier is an instance of Naive Bayes classifier that takes in account word frequency in documents [9].

SVM has been used to for the first submission. Linear classifier uses SGD as an optimization algorithm; has been implemented for second submission. For third submission, an ensemble-based model that uses four models as base classifiers, which are: RF, Multinomial Bayes, linear, and SVM classifiers. The structure of base classifiers is same as the structure shown in Fig. 1.

4 Dataset

The data provided by organizers was taken from Twitter regarding political issues and Middle East events from 2011 to 2018 and is divided into training set and testing set [6]. The training set contains 4023 labeled tweets and testing set contains 1005 unlabelled tweets.

5 Results

F-score has been used to evaluate the performance of all submissions. F-score is a harmonic mean of Precision (P) and Recall (R) [10] and calculated as follow:

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

We have used 5-fold cross-validation technique. The cross validation accuracy of all training classifiers for all submissions is given in Table 1. It is clear that SVM gives the best accuracy with higher standard deviation while development, while linear classifier gives the worst accuracy with minimum standard deviation.

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

Classifier	SVM	Linear	Ensemble
Cross Validation Accuracies	78.86%	76.74%	77.74%
	79.35%	77.24%	79.35%
	82.07%	77.58%	81.44%
	82.94%	80.32%	82.56%
	77.56%	76.56%	77.68%
	Mean = 80.15%	Mean = 77.69%	Mean = 79.75%
	STD = 2.02%	STD = 1.36%	STD = 1.96%

Among 26 submissions received by shared task organizers, our submissions achieve 5th, 11th and 13th ranks as shown in Table 2. It is clear that ensemble-based classifier gives better F1 score of our submissions.

Table 2. Final result for test data

Rank	Classifier	F1 score
5	Ensemble Based Classifier	82.1%
11	SVM Based Classifier	81.6%
13	Linear Based Classifier	81.1%

6 Conclusion and Future Work

In this work we used a simple TF/IDF with range of n-grams model to extract feature for training the classifiers. The classifiers used are SVM, linear and Ensemble-based which are used to create the submitted runs. This work can be extended by using word embeddings as features. In addition, Artificial Neural Network based classifiers can be used as a classification algorithm.

References

1. Alayba, A.M., Palade, V., England, M., Iqbal, R.: Improving sentiment analysis in arabic using word representation. 2018 IEEE 2nd International Workshop on Arabic and Derived Script Analysis and Recognition (ASAR) (Mar 2018). https://doi.org/10.1109/asar.2018.8480191
2. Barbieri, F., Saggion, H.: Modelling irony in twitter. In: Proceedings of the Student Research Workshop at the 14th Conference of the European Chapter of the Association for Computational Linguistics. pp. 56–64. Association for Computational Linguistics, Gothenburg, Sweden (Apr 2014), https://www.aclweb.org/anthology/E14-3007
3. Breiman, L.: Random forests. Machine Learning **45**(1), 5–32 (Oct 2001), https://doi.org/10.1023/A:1010933404324
4. CLIFT, R.: Irony in conversation. Language in Society **28**(4), 523–553 (1999). https://doi.org/10.1017/S0047404599004029
5. Farías, D.I.H., Patti, V., Rosso, P.: Irony detection in twitter: The role of affective content. ACM Trans. Internet Technol. **16**(3), 19:1–19:24 (Jul 2016), http://doi.acm.org/10.1145/2930663
6. Ghanem, B., Karoui, J., Benamara, F., Moriceau, V., Rosso, P.: Idat@fire2019: Overview of the track on irony detection in arabic tweets. In: Mehta, P., Rosso, P., Majumder, P., Mitra, M. (eds.) Working notes of the Forum for Information Retrieval Evaluation (FIRE 2019). CEUR Workshop Proceedings, CEUR-WS.org (2019), Kolkata, India, December 12-15
7. Karoui, J., Zitoune, F.B., Moriceau, V.: Soukhria: Towards an irony detection system for arabic in social media. Procedia Computer Science **117**, 161 – 168 (2017), http://www.sciencedirect.com/science/article/pii/S1877050917321622
8. Lee, C.J., Katz, A.N.: The differential role of ridicule in sarcasm and irony. Metaphor and Symbol **13**(1), 1–15 (1998)
9. McCallum, A., Nigam, K.: A comparison of event models for naive bayes text classification. In: AAAI-98 workshop on learning for text categorization. pp. 41–48. The COLING 2012 Organizing Committee (1998)
10. Nayel, H.A., Shashirekha, H.L.: Improving NER for clinical texts by ensemble approach using segment representations. In: Bandyopadhyay, S. (ed.) Proceedings of the 14th International Conference on Natural Language Processing, ICON 2017, Kolkata, India, December 18-21, 2017. pp. 197–204. NLP Association of India (2017), https://aclweb.org/anthology/papers/W/W17/W17-7525/
11. Nayel, H.A., Shashirekha, H.L.: Mangalore-university@inli-fire-2017: Indian native language identification using support vector machines and ensemble approach. In: Majumder, P., Mitra, M., Mehta, P., Sankhavara, J. (eds.) Working notes of FIRE 2017 - Forum for Information Retrieval Evaluation, Bangalore, India, December 8-10, 2017. CEUR Workshop Proceedings, vol. 2036, pp. 106–109. CEUR-WS.org (2017), http://ceur-ws.org/Vol-2036/T4-2.pdf

12. Nayel, H.A., Shashirekha, H.L.: Mangalore university inli@fire2018: Artificial neural network and ensemble based models for INLI. In: Mehta, P., Rosso, P., Majumder, P., Mitra, M. (eds.) Working Notes of FIRE 2018 - Forum for Information Retrieval Evaluation, Gandhinagar, India, December 6-9, 2018. CEUR Workshop Proceedings, vol. 2266, pp. 110–118. CEUR-WS.org (2018), http://ceur-ws.org/Vol-2266/T2-10.pdf
13. Nayel, H.A., Shashirekha, H.L., Shindo, H., Matsumoto, Y.: Improving multi-word entity recognition for biomedical texts. CoRR **abs/1908.05691** (2019), http://arxiv.org/abs/1908.05691
14. Reyes, A., Rosso, P.: Mining subjective knowledge from customer reviews: A specific case of irony detection. In: Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis (WASSA 2.011). pp. 118–124. Association for Computational Linguistics, Portland, Oregon (Jun 2011), https://www.aclweb.org/anthology/W11-1715
15. Reyes, A., Rosso, P.: On the difficulty of automatically detecting irony: Beyond a simple case of negation. Knowl. Inf. Syst. **40**(3), 595–614 (Sep 2014), http://dx.doi.org/10.1007/s10115-013-0652-8
16. Reyes, A., Rosso, P., Veale, T.: A multidimensional approach for detecting irony in twitter. Lang. Resour. Eval. **47**(1), 239–268 (Mar 2013), http://dx.doi.org/10.1007/s10579-012-9196-x
17. Theodoridis, S., Koutroumbas, K.: Chapter 3 - linear classifiers. In: Pattern Recognition (Fourth Edition), pp. 91 – 150. Academic Press, Boston, fourth edition edn. (2009), http://www.sciencedirect.com/science/article/pii/B9781597492720500050
18. Utsumi, A.: A unified theory of irony and its computational formalization. In: COLING 1996 Volume 2: The 16th International Conference on Computational Linguistics (1996), https://www.aclweb.org/anthology/C96-2162
19. Vanin, A.A., Freitas, L.A., Vieira, R., Bochernitsan, M.: Some clues on irony detection in tweets. In: Proceedings of the 22Nd International Conference on World Wide Web. pp. 635–636. WWW '13 Companion, ACM, New York, NY, USA (2013), http://doi.acm.org/10.1145/2487788.2488012
20. Zhang, S., Zhang, X., Chan, J., Rosso, P.: Irony detection via sentiment-based transfer learning. Information Processing and Management **56**(5), 1633 – 1644 (2019), http://www.sciencedirect.com/science/article/pii/S0306457318307428