Emotion based Voted Classier for Arabic Irony Tweet Identification

Nikita Kanwar[1], Rajesh Kumar Mundotiya[2], Megha Agarwal[1], Chandradeep Singh[1]
[1] Pratap Institute of Technology and Science, Sikar
{frathorenik2, megha.agarwal259} {chandradeepsingh904g@gmail.com}
[2] Indian Institute of Technology (BHU), Varanasi
{rajeshkm.mundotiya@gmail.com}

ABSTRACT: *In this paper, we have worked on irony detection in the Arabic language, a task which is organized by FIRE 2019. The tweets have been preprocessed and tokenized to extract the frequency-based, emotion-based features. These features are used to irony identification using the voted classier. The F-score of our proposed approach is 0.807 and the top-ranking developed method having F-score of .037, so the difference between F-score makes our approach better.*

Keywords: Voted Classifier, Emotion Feature

Received: 4 November 2019, Revised 27 January 2020, Accepted 8 February 2020

DOI: 10.6025/jdp/2020/10/2/52-56

1. Introduction

In a couple of years, social media blogs become the best way to exchange pieces of information. Irony used ubiquitously, primarily when acknowledging microblogging platforms like Twitter. These platforms aid users to dispense their opinions and thoughts on a various topic. There are many more natural language processing applications which are often observed while modelling ironic contents and with its complexity [13, 9, 10, 14].

Most of the earlier work on ironic detection using English data has been done using rule-based and feature-based techniques. In previous work of irony detection on English: A rule-based approach depends on the hashtag [12] and linguistic features (exclamations and intensifiers) [16] are used. Feature-based approach includes handcrafted features i.e. emotion features, sentiment lexicon, structural features, subjectivity lexicon as well as distributional vector representation for automatic features [16, 17]. From the natural language processing (NLP) perspective, irony helps to produce and understand human language whereas, in terms of text mining, it is more convenient where semantic analysis has a significant role, such as author profiling and deception detection, sentiment analysis, emotion analysis, online harassment detection [15, 2].

Irony detection in Arabic tweets is a challenging task due to the inclusion of dialects, non-diacritised texts, data sparsity, and code-switching with Arabic dialects, French and English[12]. We have devised a machine learning-based voted classifier tool based on emotion feature and term frequency-inverse document frequency (tf-idf) features of an Arabic tweet. For the emotion feature extraction, we exerted Google translation api to English.

2. Related Work

Irony detection has been a problem scrutinised by various disciplines, such as linguistics, philosophy, and psychology, but their description in formal terms id quite tricky. The irony is also a challenge for the sentiment analysis problems. Faras, Delia et. al. gives a irony detection on tweets data. The author considers it as a binary classification problem where mostly tweets specified with distinct hashtags. They propose an affective method for extraction of features based on an extensive range of lexical resources accessible for English. The author uses twitter data for corpora creation using two approach- self-tagging and crowdsourcing. They use irony detection model with emotion information for a complete model named emoIDM [6]. Similarly, de Freitas et al. also uses tweets data for irony detection, the author describes a set of patterns that may help to suggest ironic/ sarcastic statements. In this paper, the author has analyzed tweets under the scope of the domain "Fim do Mundo" in the experiment section [7]. On the other hand, Buschmeier et al. treats this problem as supervised learning and evaluate the different classier. The author uses the revies dataset for classification, whether the review is irony or non-irony [3].

Moving ahead to the direction of sentiment analysis in Arabic language irony detection, many authors proposed their methods with different methodologies. Heikal, Maha et al. proposed a deep learning method for sentiment analysis. The Arabic language is more complicated than any other language because of the complex structure and inclusion of various dialects in language. The author used ensemble techniques, which include Convolution neural network and Long short term memory to predict the sentiment of Arabic tweets [11]. Jihen Karouia et al. describe the sentiment analysis in Arabic tweets as a binary classification technique. They define tweet as a vector composed of four groups of features: surface, sentiment, shifter and internal context features, and applies different types of machine learning classifiers [12]. El-Masri et al. performed sentiment analysis in Arabic tweets using a combination of features; those features are preprocessing on Arabic tweets using stemming and retweets, n-gram features, lexicon-based method and machine learning-based methods [5].

Most researcher also focuses on emotion detection. Agrawal et al. proposed a methodology for sarcasm detection called automated word embedding for sarcasm, which using relevant information from word representation [1]. Duppada, Venkatesh, et al. presents a task paper for SemEval-2018 Affect in Tweets (English) sub-tasks. This task is mainly focused on the ordinal classification and regression sub-tasks for valence and emotion. The author uses four different model domain adaptation and creates an ensemble to give the final prediction [4].

3. Experimental Setup

Earlier work on ironic content identification depended on feature engineering. The handcrafted feature extraction required knowledge about language, which is a tedious task for a person who is new for the language. Our classification method solely depends upon emotional features and tf-idf features.

3.1 Dataset Statistics

In this task, the task organiser has provided a balanced dataset of 4024 posts for the training, out of which 2091 are ironic and remaining as non-ironic posts, and 1006 are the testing dataset, as mentioned in Table 1 [8].

Data	ironic	non-ironic	Total
Training	2091	1933	4024
Testing	-	-	1006

Table 1. Dataset statistics

https://cloud.google.com/translate/docs/

3.2 Data Preprocessing

In the preprocessing part, all posts were cleaned using the "Tweet Preprocessing" library with default settings. We got the clean post after removing the Hashtag, URL's, Mentions, Emoji's and Smileys for getting the stable emotion features from Deepmoji, and the get the preprocessed post after ignoring the Arabic stop words (available in NLTK) from the tokenised sentences. The tf-idf features aid to design a language-independent model. Whereas lemmatisation, stemming and grammatical features are language-dependent. The emotions and tf-idf feature have been extracted through cleaned and preprocessed posts, respectively.

3.3 Feature Extraction

For the classification task, our system combines two features: Emotion-based feature and Frequency-based feature. We have assumed that the meaning of posts will be captured through emotions and syntactic features through the frequency-based method. Emotion features extraction has been done though the translation of clean posts into English via Google translation API. The translated posts feed to deepmoji (using in default setting), which gives the emotion feature vector. The preprocessed post has used tf-idf vectorization to represent the syntax features.

3.4 Classier

Our classier model includes the robust classier: Multimodel Naive Bayes (MNB), Support Vector Machine (SVM), Logistic Regression (LR), with Stochastic Gradient Descent (SGD). We stored all tracks performance of the individual classifiers. All classier takes input in the form of concatenated features (emotion feature and tf-idf feature) matrix and output as a binary label for the categorical result. The training has done using cross-validation, in which 0.8% was used as training and remaining for validation, randomly. Each classier has a different score, as classifiers have different specialties. Therefore, we combined them in order of Multimodel Naive Bayes, Support Vector Machine, and Logistic Regression to get an ensembled result. We make a voting system based on majority rule with the classifiers. The complete processing flow has shown in Figure 1.

4. Experiment and Result

We trained different models on the training set and tested them on the validation set, using scikit-learn. The initial experimental results were based on tf-idf features as a baseline, later on, improves by concatenation of emotion features. The training of logistic regression with stochastic gradient descent with hinge loss till 50 maximum iterations and regularised by L2. Remaining classifiers applied with the default setting. The result of each classier on validation data are shown in Table 2. After using the voting system, the result of validation and test data has shown in Table 3.

Techniques	**MNB**			**LR**			**SVM**		
Labels	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
ironic	0.68	0.87	0.76	0.7	0.86	0.77	0.72	0.82	0.76
non -ironic	0.84	0.64	0.72	0.84	0.67	0.74	0.81	0.81	0.76
weighted average	0.77	0.75	0.74	0.77	0.76	0.76	0.77	0.76	0.76
Accuracy	0.745			0.755			0.761		

Table 2. Classier result on validation dataset at Precision, Recall, F-score and Accuracy in %

5. Conclusion

Our majority voted classier for Arabic tweet identification has achieved 0.75%, 0.807% F-score on cross-validation and testing

https://pypi.org/project/tweet-preprocessor/
https://www.nltk.org/
https://github.com/bfelbo/DeepMoji
https://scikit-learn.org/stable/

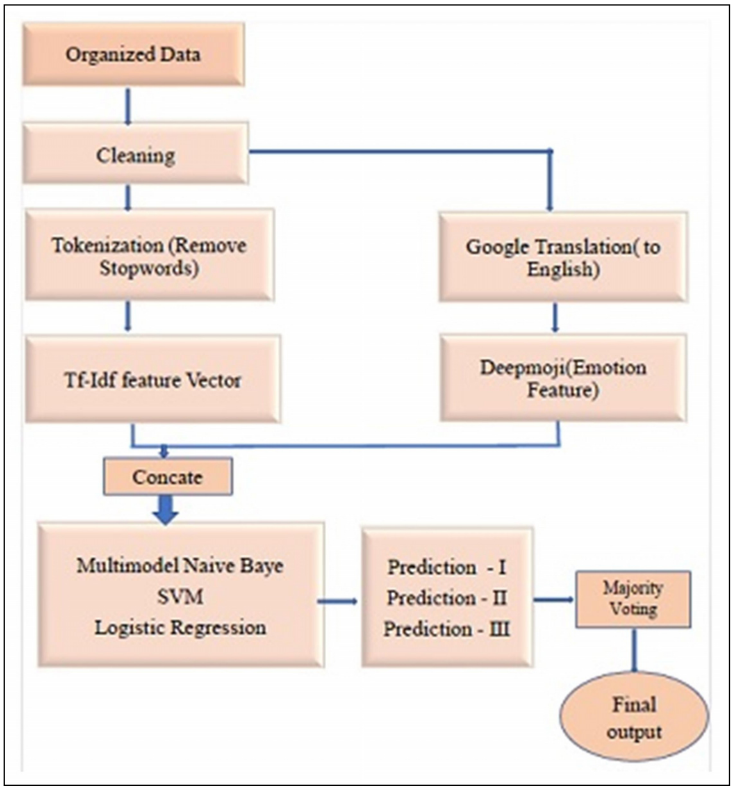

Figure 1. Voted based system for Arabic irony identification

Labels	Precision	Recall	F1
Validation Data	0.77	0.76	0.75
Testing Data	-	-	0.807

Table 3. Voted classifier result on validation and testing dataset in %

data, respectively. The F-score difference between our system and the top-ranking system is 0.037%. The data is a binary labeled, which handled through the concatenation of tf-idf and emotion based feature. That features are generated using the generative (Multimodel Naive Bayes) and discriminative (Logistic Regression, Support Vector Machine) algorithms. The voted classier is based on high majority rule. As future work, It would be helpful to use deep learning-based techniques with pre-trained subword level embeddings.

References

[1] Agrawal, A., An, A. (2018). Affective representations for sarcasm detection. *In*: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval. p. 1029-1032. ACM.

[2] Al-Ayyoub, M., Khamaiseh, A. A., Jararweh, Y., Al-Kabi, M. N. (2019). A comprehensive survey of arabic sentiment analysis. *Information Processing & Management*, 56 (2) 320-342.

[3] Buschmeier, K., Cimiano, P., Klinger, R. (2014). An impact analysis of features in a classification approach to irony detection in product reviews. *In*: Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis. p. 42-49.

[4] Duppada, V., Jain, R., Hiray, S. (2018). Seernet at semeval-2018 task 1: Domain adaptation for affect in tweets. arXiv preprint

arXiv:1804.06137.

[5] El-Masri, M., Altrabsheh, N., Mansour, H., Ramsay, A. (2017). A web-based tool for arabic sentiment analysis. *Procedia Computer Science*, 117, 38-45.

[6] Far–as, D. I. H., Patti, V., Rosso, P. (2016). Irony detection in twitter: The role of affective content. ACM Transactions on Internet Technology (TOIT), 16 (3), 19.

[7] de Freitas, L. A., Vanin, A. A., Hogetop, D. N., Bochernitsan, M. N., Vieira, R. (2014). Pathways for irony detection in tweets. *In*: Proceedings of the 29th Annual ACM Symposium on Applied Computing. p. 628-633. ACM.

[8] Ghanem, B., Karoui, J., Benamara, F., Moriceau, V., Rosso, P. (2019). Idat@re2019: Overview of the track on irony detection in arabic tweets. *In*: FIRE (Working Notes) (2019)

[9] Ghosh, A., Veale, T. (2016). Fracking sarcasm using neural network. *In*: Proceedings of the 7th workshop on computational approaches to subjectivity, sentiment and social media analysis. p. 161-169.

[10] Gonz–alez-Ib–anez, R., Muresan, S., Wacholder, N. (2011). Identifying sarcasm in twitter: a closer look. *In*: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers- 2. p. 581-586. *Association for Computational Linguistics.*

[11] Heikal, M., Torki, M., El-Makky, N. (2018). Sentiment analysis of arabic tweets using deep learning. *Procedia Computer Science* 142, 114-122.

[12] Karoui, J., Zitoune, F. B., Moriceau, V. (2017). Soukhria: Towards an irony detection system for arabic in social media. *Procedia Computer Science* 117, 161-168.

[13] Liu, B. (2012). Sentiment analysis and opinion mining. Synthesis lectures on human language technologies 5 (1) 1-167.

[14] Maynard, D., Greenwood, M. A. (2014). Who cares about sarcastic tweets? investigating the impact of sarcasm on sentiment analysis. *In*: LREC 2014 Proceedings. ELRA.

[15] Rosso, P., Rangel, F., Far–as, I.H., Cagnina, L., Zaghouani, W., Char, A. (2018). A survey on author profiling, deception, and irony detection for the arabic language. *Language and Linguistics Compass*, 12 (4), e12275.

[16] Van Hee, C., Lefever, E., Hoste, V. (2018). Semeval-2018 task 3: Irony detection in en- glish tweets. *In*: Proceedings of The 12th International Workshop on Semantic Evaluation. p. 39-50.

[17] Zhang, S., Zhang, X., Chan, J., Rosso, P. (2019). Irony detection via sentiment-based transfer learning. *Information Processing & Management,* 56 (5) 1633-1644.