Irony Detector at SemEval-2018 Task 3: Irony Detection in English Tweets using Word Graph

Usman Ahmed, Lubna Zafar, Faiza Qayyum and Muhammad Arshad Islam
Parallel Computing Network,
Department of Computer Science,
Capital University of Science and Technology,
Islamabad, Pakistan.
usmanahmed189@gmail.com,
lubbnaa@gmail.com,
faizaqayyum@cust.edu.pk,
arshad.islam@cust.edu.pk.

Abstract

This paper describes the Irony detection system that participates in SemEval-2018 Task 3: Irony detection in English tweets. The system participated in the subtasks A and B. This paper discusses the results of our system in the development, evaluation and post evaluation. Each class in the dataset is represented as directed unweighted graphs. Then, the comparison is carried out with each class graph which results in a vector. This vector is used as features by machine learning algorithm. The model is evaluated on a hold on strategy. The organizers randomly split 80% (3,833 instances) training set (provided to the participant in training their system) and testing set 20% (958 instances). The test set is reserved to evaluate the performance of participants systems. During the evaluation, our system ranked 23 in the Coda Lab result of the subtask A (binary class problem). The binary class system achieves accuracy 0.6135, precision 0.5091, recall 0.7170 and F measure 0.5955. The subtask B (multi-class problem) system is ranked 22 in Coda Lab results. The multiclass model achieves the accuracy 0.4158, precision 0.4055, recall 0.3526 and f measure 0.3101.

1 Introduction

Social media are deemed as a diverse web-based network that serves as an online platform to communicate and disseminate information or ideas among individuals and fraternities. Since its advent, people all around the globe harness it as a major source to express their opinions or emotions, however, an expeditious increase in its usage has been reported in the last decade (Kelly et al., 2016; Perrin, 2015). Among the multifarious range of social media platforms, Twitter is the most popular one. It is basically a microblogging site diffuses information pertaining to what is happening around the world, and what are the current top-interest areas among the wider population (Rosenthal et al., 2017). According to a recent survey, 6000 tweets per second are sent by 320 million active monthly users, thus 500 million tweets per day (Statistics, 2014). This poses a challenge for the scientific community to accurately discern the sentiment of a tweet out of this plethora. Since certain aspects associated with sentiment analysis are quite arduous yet feasible to ascertain (such as negative, positive, a neutral aspect of the opinion) than irony.

Irony detection has its implications in sentiment analysis (Reyes et al., 2009), opinion mining (Sarmento et al., 2009) and advertising (Kreuz, 2001). For the past few years, irony-aware sentiment analysis has attained significant computational treatment due to the prevalence of irony on the web content (Farías et al., 2016). It is a broad concept, which has an association with multiple disciplines such as psychology, linguistics, etc. The irony is to efficaciously delineate a contrary aspect of the utterance (Grice, 1975). Irony cannot be detected with the simple scrutiny of words expressed in a statement, whereas, an aspect of irony is implicitly connected with the utterance. Furthermore, it could be deemed as a stance that has been expressed in an ironic or sarcastic environment (Grice, 1975; Alba-Juez and Attardo, 2014). Detection of this implicit aspect poses a strenuous computational challenge over the scientific community in terms of initiating effective models in this regard. In the stream of irony detection, the first-ever computer model was proposed by (Utsumi, 1996). Subsequently, various other models have been presented that have specifically addressed the irony detection among tweets by using different features such as, cue-words or user-generated tags (i.e., Hashtags) etc (Van Hee, 2017; Hernández-Farías et al., 2015; Reyes et al., 2013).

Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018), pages 581–586
New Orleans, Louisiana, June 5–6, 2018.

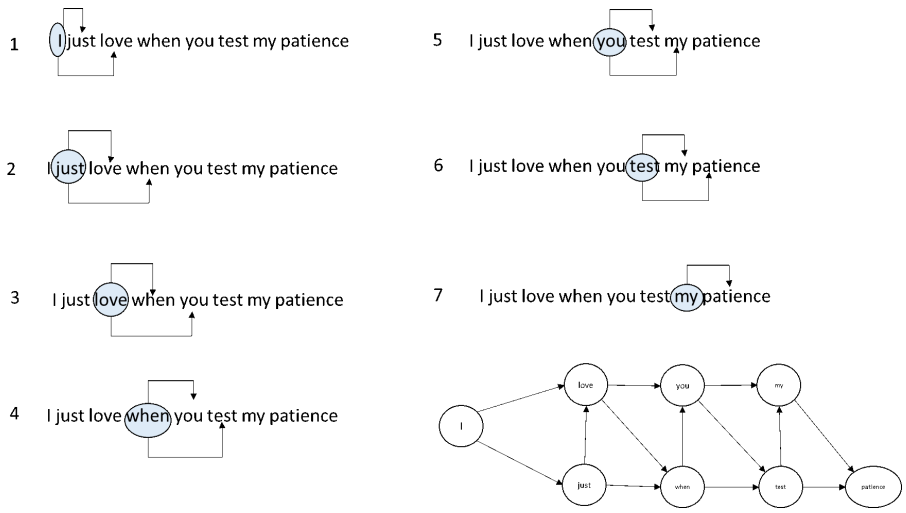

Figure 1: Graph Construction with vicinity size 2 illustrates how the vicinity size move toward the end of the tweet; in this example the frame is the two following words and for each word some edges and nodes are added to the graph.

Though, there does not exist any optimal model that could be considered as a baseline for irony detection. This paper presents a model to automatically detect sarcasm or irony from the plethora of tweets. The proposed model is used in the two subtasks. The first module assigns the binary value against tweets (i.e., 1 indicates that tweet in ironic and 0 indicates that a tweet is non-ironic). The second module performs multi-class classification: (i) verbal irony realized through a polarity contrast, ii) verbal irony without such a polarity contrast (i.e., Other verbal ironies), iii) descriptions of situational irony and iv) non-irony. For classification, data set is comprised of 4792 samples, taken from GitHub link provided by the SemEval 2018 organizers.

2 Task Overview

In SemEval-2018 (Cynthia Van Hee, 2018), task 3 contains two subtasks for the detection of Irony in English tweets. In the first task, the system has to determine whether a tweet is ironic or non- ironic, making it a binary classification problem. The second task is the multiclass classification problem where the ironic and non-ironic task is further divided into four categories as mentioned below:

1. verbal irony realized through a polarity contrast
2. verbal irony without such a polarity contrast (i.e., other verbal irony)
3. descriptions of situational irony
4. Non-irony

Systems are evaluated using standard evaluation metrics, including accuracy, precision, recall and F1-score.

3 Proposed Model

The proposed model is inspired by the previous work (Giannakopoulos et al., 2008; Maas et al., 2011), however, we used some additional features as well as a word graph similarity score. Each tweet is represented as directed unweighted word graph and the edge between each word is created based on the vicinity window size explained in 1. Each class in the dataset is represented as directed unweighted graphs. Then, the comparison is carried out with each class graph which results in a vector. This vector is used as features by machine learning algorithm. The graph is constructed based on a class assignment and then we measure the similarity of a tweet with each class graph. The similarity between two graphs (tweet graph and class graph) can be measured in multiple ways, but in this research, we used the containment similarity (non-normalized value), maximum common subgraph similarity and its variant compare graph in terms of similarity.

3.1 Graph Construction

The tweet contained a set of words. Theses word will be used to construct the word graph based on their vicinity. Each word in the tweet is represented by the labelled node. The nodes within

Class Name	Number of Coloumn
Verbal irony by means of a polarity contrast	1728
Other types of verbal irony	267
Situational Irony	401
Non-Ironic	604

Table 1: Data set Description

window size are joined by an edge. The sequence of the words is preserved by using directed edges. The size of the vicinity window can affect the accuracy of the method. In this research, we used a vicinity size of 2, as seen in 1

The graph similarity between the graph of a tweet and the graph of the irony class can define the degree of irony in the tweet. For the purposes of our study, we used the containment similarity (non-normalized value), maximum common subgraph similarity and its variant compare graph.

3.2 Dataset

The dataset is provided on the GitHub source. This corpus is constructed of 3,000 English language tweets. These tweets are searched by using hashtags #irony, #sarcasm and #not. The data were collected from the period of five months (1st December 2014 to 1st April 2015) and represent 2,676 unique users. All tweets were manually annotated using the scheme of Van el al (Van Hee et al., 2016). The organizer used the services of three students in linguistics as well as English language speakers to annotate the entire corpus. The (Stenetorp et al., 2012) tool was used as the annotation tool. The percentage agreement score (kappa scores 0.72) is also calculated for the annotation. The number of instances for each class is mentioned in Table 1.

As seen in Table, 2396 instances are ironic (1,728 + 267 + 401) while 604 are non-ironic. The organizer balances the class data by using background corpus. After balancing the total data set contain 4,792 tweets that contain 2,396 ironic and 2,396 non-ironic tweets. The SemEval-2018 competition used the hold on the strategy to check the effectiveness of each participated system. The organizers randomly split 80% (3,833 instances) training set (provided to the participant in training their system) and testing set 20% (958 instances). The test set is reserved to evaluate the performance of participants systems.

3.3 Feature Engineering

3.3.1 Containment Similarity

The containment similarity measure has been used to calculate, graph similarity (Aisopos et al., 2012). In this research, we used bigram nodes. The measure expresses the common edges between two graphs by the number of edges of the smaller graph.

$$CS(G_T, G_S) = \frac{\sum_{e=G_T} \mu(e, G_S)}{min(|G_T|, |G_S|)} \quad (1)$$

Where GT (target graph) is the word graph of a tweet, Gs (source graph) is the word graph of an irony classes. The graph size can be the number of nodes or edges that are contained. e is an edge of a word graph.

3.3.2 Maximum Common Sub graph

The maximum common sub graph similarity is based on the size of the graph. We used the three variations of the metric are described in the equation 2, 3 and 4

$$MCSNS = \frac{MCSN(|G_T|, |G_S|)}{min(|G_T|, |G_S|)} \quad (2)$$

Maximum Common Sub graph Node Similarity (MCSNS): where MCSNS (GT (target graph) — Gs (source graph)) is the total number of nodes that are contained in the MCS of that graphs..

$$MCSUES = \frac{MCSUE(|G_T|, |G_S|)}{min(|G_T|, |G_S|)} \quad (3)$$

Maximum Common Sub graph Edge Similarity (MCSNS): where MCSUE (GT (target graph) — Gs (source graph)) is the total number of the edges contained in the MCS regardless the direction of them.

$$MCSDES = \frac{MCSDE(|G_T|, |G_S|)}{min(|G_T|, |G_S|)} \quad (4)$$

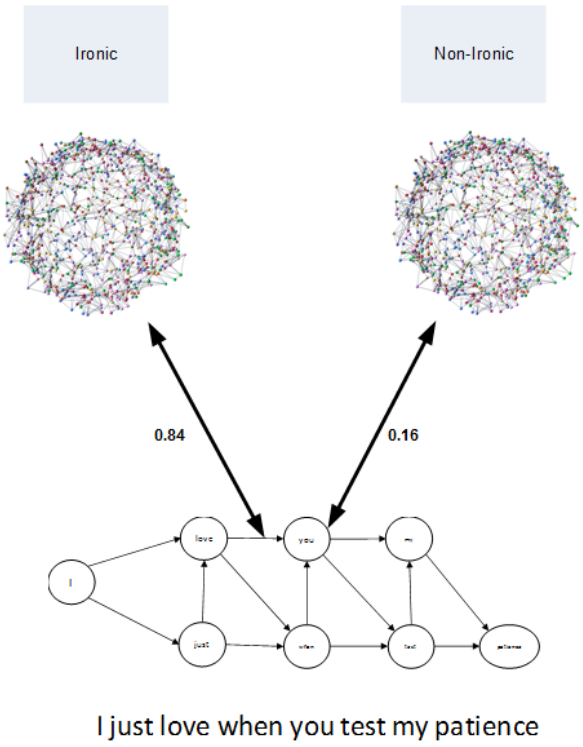

Figure 2: Graph Similarity Feature Extraction for one measure. The graph of a tweet used to compare with training data class graphs, in order to produce two numbers (depending upon the numbers of classes). These numbers will be used as a feature vector. The feature vector is provided to trained model to predict the class of the new tweet.

Maximum Common Sub graph Directed Edge Similarity (MCSNS): where MCSDES (GT (target graph) — Gs (source graph)) is the number of the edges contained in the MCS and have the same direction in the graphs.

3.3.3 Tweet Polarity and Latent Dirichlet Allocation

We used the SenticNet library to calculate the sentence polarity score as well as subjectivity score. Moreover, we also perform latent Dirichlet Allocation on the corpus and then used the trained model to calculate similarity helinger distance for each class (Blei et al., 2003; Beran, 1977).

3.4 Model Selection

In this paper, we used Tree-based Pipeline Optimization Tool (TPOT) that designs and optimizes the machine learning pipelines by using an evolutionary algorithm (Olson et al., 2016). The labelled data are provided for TPOT classification. Both TPOT classes return hyper tune model for both types of data (binary and Multiclass problem). After, data analysis, it was observed that the number of classes in the multiclass dataset is a

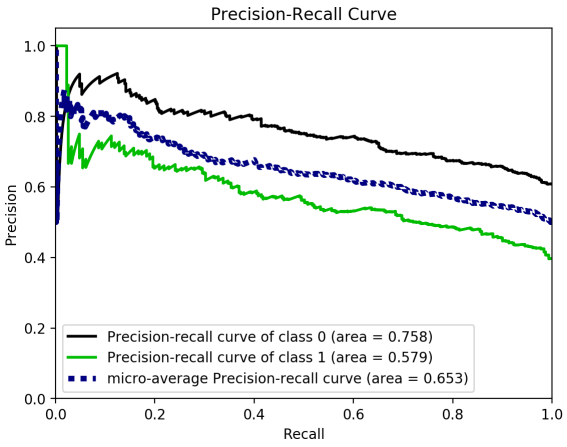

Figure 3: Precision Recall Curve of Binary Class problem

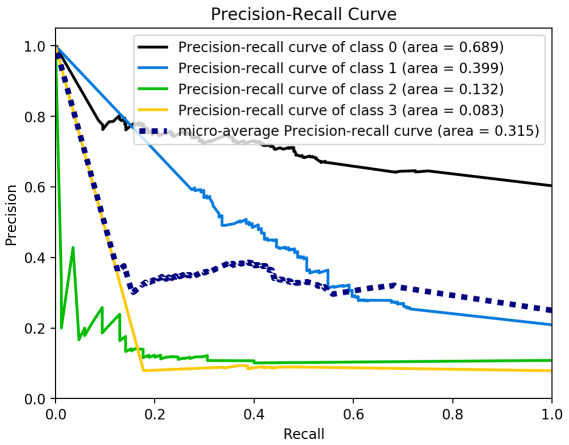

Figure 4: Precision Recall Curve of Multi Class problem

significant imbalance, which gives rise to the class imbalance problem. In order to handle this problem, we used SMOTE (Cummins et al., 2017) a Python toolbox to tackle the curse of imbalanced data. For binary classification problem, TPOT gives extreme gradient boosting classifier tune parameters. For the multiclass problem, TPOT gives stacking of extreme gradient boosting classifiers, extra trees classifier and random forest classifier.

4 Results Evaluation

For experimentation, we used efficient tool sklearn (Machine Learning Library) to train machine models mentioned above (Pedregosa et al., 2011). For both model hold on strategy was adopted. Training data contain 80% (3,833 instances) and testing sets 20% (958 instances). Our system ranked 23 in the Coda Lab result of the binary classification problem. The binary class system achieves accuracy 0.6135, precision 0.5091, recall 0.7170 and F measure 0.5955. After the release of the gold set, the model is again tuned by using TPOT library and result are evaluated as seen in Figure 3. Our system ranked 22 in the Coda Lab result of the multi-class problem. The multiclass model achieves the accuracy 0.4158, precision 0.4055, recall 0.3526 and f measure 0.3101. After the release of the gold set model was retrained and evaluated. The result of the multiclass problem is shown in Figure 4

5 Conclusion and Analysis

An innovative citation classification technique is proposed that combines the well-described structure of graphs with classification algorithm. The word graphs can seize the collection of the words that are contained in a tweet. The tweet word graph is generated and then by using several graph similarity techniques is applied to the dataset. These graph similarity metrics output is represented as a feature vector by the classification algorithm. It is concluded that word graph with different vicinity window is a good source of information to classify irony in the tweet. The model can be improved by using a large dataset. The proposed method can be enhanced by using a different graph similarity metric as features. The word graph construction method with different vicinity window size might improve results.

References

Fotis Aisopos, George Papadakis, Konstantinos Tserpes, and Theodora Varvarigou. 2012. Content vs. context for sentiment analysis: a comparative analysis over microblogs. In *Proceedings of the 23rd ACM conference on Hypertext and social media*, pages 187–196. ACM.

Laura Alba-Juez and Salvatore Attardo. 2014. The evaluative palette of verbal irony. *Evaluation in context*, 242:93.

Rudolf Beran. 1977. Minimum hellinger distance estimates for parametric models. *The annals of Statistics*, pages 445–463.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Chris Cummins, Pavlos Petoumenos, Zheng Wang, and Hugh Leather. 2017. Synthesizing benchmarks for predictive modeling. In *Code Generation and Optimization (CGO), 2017 IEEE/ACM International Symposium on*, pages 86–99. IEEE.

Veronique Hoste Cynthia Van Hee, Els Lefever. 2018. Semeval-2018 task 3: Irony detection in english tweets. in proceedings of the 12th international workshop on semantic evaluation.

Delia Irazú Hernańdez Farías, Viviana Patti, and Paolo Rosso. 2016. Irony detection in twitter: The role of affective content. *ACM Transactions on Internet Technology (TOIT)*, 16(3):19.

George Giannakopoulos, Vangelis Karkaletsis, George Vouros, and Panagiotis Stamatopoulos. 2008. Summarization system evaluation revisited: N-gram graphs. *ACM Transactions on Speech and Language Processing (TSLP)*, 5(3):5.

H Paul Grice. 1975. Logic and conversationin p. cole and j. morgan (eds.) syntax and semantics volume 3: Speech acts.

Irazú Hernández-Farías, José-Miguel Benedí, and Paolo Rosso. 2015. Applying basic features from sentiment analysis for automatic irony detection. In *Iberian Conference on Pattern Recognition and Image Analysis*, pages 337–344. Springer.

Brendan S Kelly, Ciaran E Redmond, Gregory J Nason, Gerard M Healy, Niall A Horgan, and Eric J Heffernan. 2016. The use of twitter by radiology journals: an analysis of twitter activity and impact factor. *Journal of the American College of Radiology*, 13(11):1391–1396.

R Kreuz. 2001. Using figurative language to increase advertising effectiveness. In *Office of naval research military personnel research science workshop. Memphis, TN*.

Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*, pages 142–150. Association for Computational Linguistics.

Randal S Olson, Nathan Bartley, Ryan J Urbanowicz, and Jason H Moore. 2016. Evaluation of a tree-based pipeline optimization tool for automating data science. In *Proceedings of the Genetic and Evolutionary Computation Conference 2016*, pages 485–492. ACM.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.

Andrew Perrin. 2015. Social media usage: 2005-2015.

Antonio Reyes, Paolo Rosso, and Davide Buscaldi. 2009. Humor in the blogosphere: First clues for a verbal humor taxonomy. *Journal of Intelligent Systems*, 18(4):311–332.

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

Sara Rosenthal, Noura Farra, and Preslav Nakov. 2017. Semeval-2017 task 4: Sentiment analysis in twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 502–518.

Luís Sarmento, Paula Carvalho, Mário J Silva, and Eugénio De Oliveira. 2009. Automatic creation of a reference corpus for political opinion mining in user-generated content. In *Proceedings of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion*, pages 29–36. ACM.

Twitter Usage Statistics. 2014. Internet live stats.

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. Brat: a web-based tool for nlp-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107. Association for Computational Linguistics.

Akira Utsumi. 1996. A unified theory of irony and its computational formalization. In *Proceedings of the 16th conference on Computational linguistics-Volume 2*, pages 962–967. Association for Computational Linguistics.

Cynthia Van Hee. 2017. *Can machines sense irony?: exploring automatic irony detection on social media*. Ph.D. thesis, Ghent University.

Cynthia Van Hee, Els Lefever, and Véronique Hoste. 2016. Exploring the realization of irony in twitter data. In *LREC*.