NIHRIO at SemEval-2018 Task 3: A Simple and Accurate Neural Network Model for Irony Detection in Twitter

Thanh Vu[1], **Dat Quoc Nguyen**[2], **Xuan-Son Vu**[3], **Dai Quoc Nguyen**[4],
Michael Catt[1] and **Michael Trenell**[1]
[1]NIHRIO, Newcastle University, UK; [2]The University of Melbourne, Australia;
[3]Umeå University, Sweden; [4]Deakin University, Australia
thanh.vu@io.nihr.ac.uk; dqnguyen@unimelb.edu.au;
sonvx@cs.umu.se; dai.nguyen@deakin.edu.au;
{michael.catt, michael.trenell}@io.nihr.ac.uk

Abstract

This paper describes our NIHRIO system for SemEval-2018 Task 3 "Irony detection in English tweets." We propose to use a simple neural network architecture of Multilayer Perceptron with various types of input features including: lexical, syntactic, semantic and polarity features. Our system achieves very high performance in both subtasks of binary and multi-class irony detection in tweets. In particular, we rank third using the accuracy metric and fifth using the F_1 metric. Our code is available at: https://github.com/NIHRIO/IronyDetectionInTwitter.

1 Introduction

Mining Twitter data has increasingly been attracting much research attention in many NLP applications such as in sentiment analysis (Pak and Paroubek, 2010; Kouloumpis et al., 2011; Agarwal et al., 2011; Liu et al., 2012; Rosenthal et al., 2017; Cambria et al., 2018) and stock market prediction (Bollen et al., 2011; Vu et al., 2012; Bartov et al., 2015; Nofer and Hinz, 2015; Oliveira et al., 2017). Recently, Davidov et al. (2010) and Reyes et al. (2013) have shown that Twitter data includes a high volume of "*ironic*" tweets. For example, a user can use positive words in a Twitter message to her intended negative meaning (e.g., *"It is awesome to go to bed at 3 am #not"*). This especially results in a research challenge to assign correct sentiment labels for ironic tweets (Bosco et al., 2013; Ghosh et al., 2015; Farías et al., 2016; Nozza et al., 2017; Kannangara, 2018).

To handle that problem, much attention has been focused on automatic irony detection in Twitter (Davidov et al., 2010; Reyes et al., 2013; Barbieri and Saggion, 2014; Rajadesingan et al., 2015; Farías et al., 2016; Sulis et al., 2016; Karoui et al., 2017; Joshi et al., 2017; Huang et al., 2017; Ravi and Ravi, 2017). In this paper, we propose a neural network model for irony detection in tweets. Our model obtains the fifth best performances in both binary and multi-class irony detection subtasks in terms of F_1 score (Van Hee et al., 2018). Details of the two subtasks can be found in the task description paper (Van Hee et al., 2018). We briefly describe the subtasks as follows:

Subtask 1 (A): Ironic vs non-ironic This first subtask is a binary classification problem, in which we predict whether or not a tweet is *ironic*. For example, *"I just love when you test my patience!! #not"* is ironic, but *"Had no sleep and have got school now #not happy"* is non-ironic.

Subtask 2 (B): Different types of irony This second subtask is a multi-class classification problem, where we predict the correct label of a tweet from four classes: (1) *non-irony*, (2) *verbal irony by means of a polarity contrast*, (3) *other verbal irony* and (4) *situational irony*.

The remainder of this paper is organized as follows: We describe the ironic tweet dataset provided by the SemEval-2018 Task 3 in Section 2. We then describe our system in Section 3. The experimental results and conclusion are detailed in Section 4 and Section 5, respectively.

2 Dataset

The dataset consists of 4,618 tweets (2,222 ironic + 2,396 non-ironic) that are manually labelled by three students. Some pre-processing steps were applied to the dataset, such as the emoji icons in a tweet are replaced by a describing text using the Python emoji package.[1] Additionally, all the

[1] https://pypi.python.org/pypi/emoji/

Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018), pages 525–530
New Orleans, Louisiana, June 5–6, 2018. ©2018 Association for Computational Linguistics

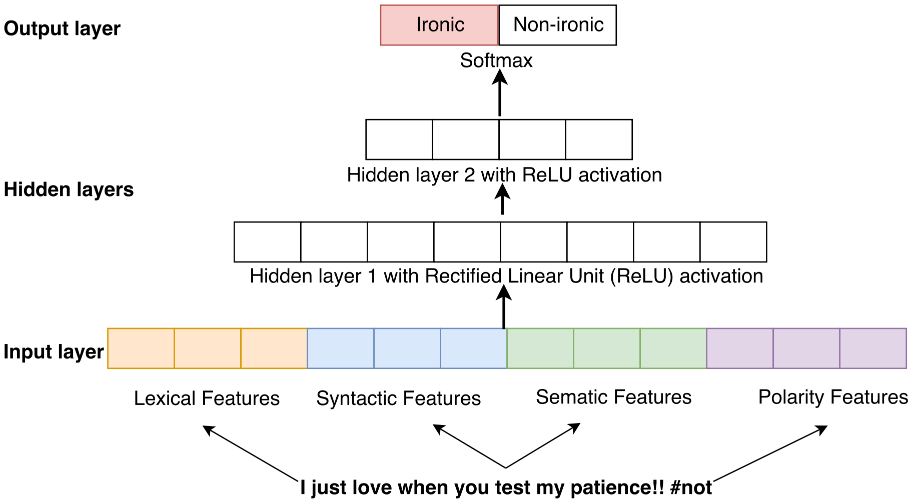
Output layer
Ironic
Non-ironic
Softmax
Hidden layers
Hidden layer 2 with ReLU activation
Hidden layer 1 with Rectified Linear Unit (ReLU) activation
Input layer
Lexical Features
Syntactic Features
Sematic Features
Polarity Features
I just love when you test my patience!! #not

Figure 1: Overview of our model architecture for irony detection in tweets.

Statistics	Training	Test
#samples	3,834	784
#non-irony	1,923	473
#irony	1,911	311
- *polarity contrast verbal*	1,390	164
- *other verbal*	316	85
- *situational*	205	62

Table 1: Basic statistics of the provided dataset.

ironic hashtags, such as #not, #sarcasm, #irony, in the dataset have been removed. This makes difficult to correctly predict the label of a tweet. For example, *"@coreybking thanks for the spoiler!!!! #not"* is an ironic tweet but without #not, it probably is a non-ironic tweet. The dataset is split into the training and test sets as detailed in Table 1.

Note that there is also an extended version of the training set, which contains the ironic hashtags. However, we only use the training set which does not contain the ironic hashtags to train our model as it is in line with the test set.

Our data pre-processing step: Tweet normalization is an important pre-processing step as there are around 15% of tweets containing 50% or more out-of-vocabulary tokens (Han and Baldwin, 2011). We normalize each tweet from the dataset using a lexicon-based approach proposed by Han et al. (2012), using a manually constructed normalization dictionary (e.g., "reeeaaalll" is normalized by "real'). We then replace all tagged users and urls by specific word tokens "<USER>" and "<URL>", respectively. It is because they are likely not correlated with the ironic labels.

3 Our modeling approach

We first describe our MLP-based model for ironic tweet detection in Section 3.1. We then present the features used in our model in Section 3.2.

3.1 Neural network model

We propose to use the Multilayer Perceptron (MLP) model (Hornik et al., 1989) to handle both the ironic tweet detection subtasks. Figure 1 presents an overview of our model architecture including an input layer, two hidden layers and a softmax output layer. Given a tweet, the input layer represents the tweet by a feature vector which concatenates lexical, syntactic, semantic and polarity feature representations. The two hidden layers with ReLU activation function take the input feature vector to select the most important features which are then fed into the softmax layer for ironic detection and classification.

3.2 Features

Table 2 shows the number of lexical, syntactic, semantic and polarity features used in our model.

Lexical features: Our lexical features include 1-, 2-, and 3-grams in both word and character levels. For each type of n-grams, we utilize only the top 1,000 n-grams based on the term frequency-inverse document frequency (tf-idf) values. That is, each n-gram appearing in a tweet becomes an entry in the feature vector with the corresponding feature value tf-idf. We also use the number of characters and the number of words as features.

Name	**# Features**
Lexical features	2,002
Syntactic features	45
Semantic features	700
Polarity features	12
Total	**2,759**

Table 2: Number of features used in our model

Cluster	**Word**	**Cluster**	**Word**
110000	wife	11001000	adorable
110000	sister	11001000	excellent
110000	boyfriend	11001000	interesting
110000	daughter	11001000	blessed
110000	mum	11001000	easy
110000	son	11001000	perfect
110000	dad	11001000	cool
110000	family	11001000	funny

Table 3: Example of clusters produced by the Brown clustering algorithm.

Syntactic features: We use the NLTK toolkit to tokenize and annotate part-of-speech tags (POS tags) for all tweets in the dataset. We then use all the POS tags with their corresponding tf-idf values as our syntactic features and feature values, respectively.

Semantic features: A major challenge when dealing with the tweet data is that the lexicon used in a tweet is informal and much different from tweet to tweet. The lexical and syntactic features seem not to well-capture that property. To handle this problem, we apply three approaches to compute tweet vector representations.

Firstly, we employ 300-dimensional pre-trained word embeddings from GloVe (Pennington et al., 2014) to compute a tweet embedding as the average of the embeddings of words in the tweet.

Secondly, we apply the latent semantic indexing (Papadimitriou et al., 1998) to capture the underlying semantics of the dataset. Here, each tweet is represented as a vector of 100 dimensions.

Thirdly, we also extract tweet representation by applying the Brown clustering algorithm (Brown et al., 1992; Liang, 2005)[2]—a hierarchical clustering algorithm which groups the words with similar meaning and syntactical function together. Applying the Brown clustering algorithm, we obtain a set of clusters, where each word belongs to only one cluster. For example in Table 3, words that indicate the members of a family (e.g., "mum", "dad") or positive sentiment (e.g., "interesting", "awesome") are grouped into the same cluster. We run the algorithm with different number of clustering settings (i.e., 80, 100, 120) to capture multiple semantic and syntactic aspects. For each clustering setting, we use the number of tweet words in each cluster as a feature. After that, for each tweet, we concatenate the features from all the clustering settings to form a cluster-based tweet embedding.

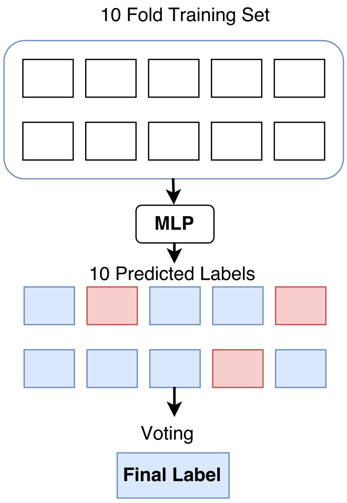

Figure 2: The training mechanism.

Polarity features: Motivated by the verbal irony by means of polarity contrast, such as *"I really love this year's summer; weeks and weeks of awful weather"*, we use the number of polarity signals appearing in a tweet as the polarity features. The signals include positive words (e.g., love), negative words (e.g., awful), positive emoji icon and negative emoji icon. We use the sentiment dictionaries provided by Hu and Liu (2004) to identify positive and negative words in a tweet. We further use boolean features that check whether or not a negation word is in a tweet (e.g., *not*, *n't*).

3.3 Implementation details

We use Tensorflow (Abadi et al., 2015) to implement our model. Model parameters are learned to minimize the the cross-entropy loss with L_2 regularization. Figure 2 shows our training mechanism. In particular, we follow a 10-fold cross-validation based voting strategy. First, we split the training set into 10 folds. Each time, we combine 9 folds to train a classification model and use the remaining fold to find the optimal hyperparameters.

[2] https://github.com/percyliang/brown-cluster

Name	1 (A)	2 (B)
Hidden layers	(800, 400)	(800, 300)
# epoch	100	100
early stop	30	30
Learning rate	10^{-4}	10^{-4}
l_2	10^{-5}	10^{-5}

Table 4: The optimal hyperparameter settings for subtasks 1 (A) and 2 (B).

Accuracy	Precision	Recall	$\mathbf{F_1}$
70.15_3	60.91	69.13	64.76_5

Table 5: The performance (in %) of our model on the test set for subtask 1 (binary classification). The subscripts denote our official ranking.

Table 4 shows optimal settings for each subtask.

In total, we have 10 classification models to produce 10 predicted labels for each test tweet. Then, we use the voting technique to return the final predicted label.

4 Experiments

4.1 Metrics

The metrics used to evaluate our model include *accuracy*, *precision*, *recall* and F_1. The accuracy is calculated using all classes in both tasks. The remainders are calculated using only the positive label in subtask 1 or per class label (i.e., macro-averaged) in subtask 2. Detail description of the metrics can be found in Van Hee et al. (2018).

4.2 Results for subtask 1

Table 5 shows our official results on the test set for subtask 1 with regards to the four metrics. By using a simple MLP neural network architecture, our system achieves a high performance which is ranked **third** and **fifth** out of forty-four teams using accuracy and F_1 metrics, respectively.

4.3 Results for subtask 2

Table 6 presents our results on the test set for subtask 2. Our system also achieves a high performance which is ranked **third** and **fifth** out of thirty-two teams using accuracy and F_1 metrics, respectively. We also show in Table 7 the performance of our system on different class labels. For ironic classes, our system achieves the best performance on the verbal irony by means of a polarity contrast with F_1 of 60.73%. Note that the performance on the situational class is not high. The reason is probably that the number of situational tweets in the training set is small (205/3,834), i.e. not enough to learn a good classifier.

Accuracy	Precision	Recall	$\mathbf{F_1}$
65.94_3	54.46	44.75	44.37_5

Table 6: The performance (in %) of our model on the test set for subtask 2 (multi-class classification).

Class	Precision	Recall	$\mathbf{F_1}$
Non-irony	72.97	79.92	76.29
Contrast verbal	53.21	70.73	60.73
Other verbal	48.78	23.53	31.75
Situational	42.86	4.84	8.70

Table 7: The performance (in %) of our model on the test set for each class label in subtask 2.

4.4 Discussions

Apart from the described MLP models, we have also tried other neural network models, such as Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) and Convolutional Neural Network (CNN) for relation classification (Kim, 2014). We found that LSTM achieves much higher performance than MLP does on the extended training set containing the ironic hashtags (about 92% vs 87% with 10-fold cross-validation using F_1 on subtask 1). However, without the ironic hashtags, the performance is lower than MLP's. We also employed popular machine learning techniques, such as SVM (Hearst et al., 1998), Logistic Regression (Harrell, 2001), Ridge Regression Classifier (Le Cessie and Van Houwelingen, 1992), but none of them produces as good results as MLP does. We have also implemented ensemble models, such as voting, bagging and stacking. We found that with 10-fold cross-validation based voting strategy, our MLP models produce the best irony detection and classification results.

5 Conclusion

We have presented our NIHRIO system for participating the Semeval-2018 Task 3 on "Irony detection in English tweets". We proposed to use Multilayer Perceptron to handle the task using various features including lexical features, syntactic features, semantic features and polarity features. Our system was ranked the fifth best performing one with regards to F_1 score in both the subtasks of binary and multi-class irony detection in tweets.

Acknowledgments

This research is supported by the National Institute for Health Research (NIHR) Innovation Observatory at Newcastle University, United Kingdom.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, and Rebecca Passonneau. 2011. Sentiment analysis of twitter data. In *Proceedings of the workshop on languages in social media*, pages 30–38.

Francesco Barbieri and Horacio Saggion. 2014. 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*, pages 56–64.

Eli Bartov, Lucile Faurel, and Partha Mohanram. 2015. Can twitter help predict firm-level earnings and stock returns? *The Accounting Review*.

Johan Bollen, Huina Mao, and Xiaojun Zeng. 2011. Twitter mood predicts the stock market. *Journal of computational science*, 2(1):1–8.

Cristina Bosco, Viviana Patti, and Andrea Bolioli. 2013. Developing corpora for sentiment analysis: The case of irony and senti-tut. *IEEE Intelligent Systems*, 28(2):55–63.

Peter F. Brown, Peter V. deSouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai. 1992. Class-based n-gram models of natural language. *Comput. Linguist.*, 18(4):467–479.

Erik Cambria, Soujanya Poria, Devamanyu Hazarika, and Kenneth Kwok. 2018. Senticnet 5: discovering conceptual primitives for sentiment analysis by means of context embeddings. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*.

Dmitry Davidov, Oren Tsur, and Ari Rappoport. 2010. Semi-supervised recognition of sarcastic sentences in twitter and amazon. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, pages 107–116.

Delia Irazú Hernańdez Farías, Viviana Patti, and Paolo Rosso. 2016. Irony detection in twitter: The role of affective content. *ACM Trans. Internet Technol.*, 16(3):19:1–19:24.

Aniruddha Ghosh, Guofu Li, Tony Veale, Paolo Rosso, Ekaterina Shutova, John Barnden, and Antonio Reyes. 2015. Semeval-2015 task 11: Sentiment analysis of figurative language in twitter. In *Proceedings of the 9th International Workshop on Semantic Evaluation*, pages 470–478.

Bo Han and Timothy Baldwin. 2011. Lexical normalisation of short text messages: Makn sens a #twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, pages 368–378.

Bo Han, Paul Cook, and Timothy Baldwin. 2012. Automatically constructing a normalisation dictionary for microblogs. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 421–432.

Frank E Harrell. 2001. Ordinal logistic regression. In *Regression modeling strategies*, pages 331–343.

Marti A. Hearst, Susan T Dumais, Edgar Osuna, John Platt, and Bernhard Scholkopf. 1998. Support vector machines. *IEEE Intelligent Systems and their applications*, 13(4):18–28.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

K. Hornik, M. Stinchcombe, and H. White. 1989. Multilayer feedforward networks are universal approximators. *Neural Netw.*, 2(5):359–366.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177.

Yu-Hsiang Huang, Hen-Hsen Huang, and Hsin-Hsi Chen. 2017. Irony detection with attentive recurrent neural networks. In *European Conference on Information Retrieval*, pages 534–540.

Aditya Joshi, Pushpak Bhattacharyya, and Mark J Carman. 2017. Automatic sarcasm detection: A survey. *ACM Computing Surveys*, 50(5):73.

Sandeepa Kannangara. 2018. Mining twitter for fine-grained political opinion polarity classification, ideology detection and sarcasm detection. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 751–752.

Jihen Karoui, Benamara Farah, Véronique Moriceau, Viviana Patti, Cristina Bosco, and Nathalie Aussenac-Gilles. 2017. Exploring the impact of pragmatic phenomena on irony detection in tweets: A multilingual corpus study. In *Proceedings of the*

15th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers), pages 262–272.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1746–1751.

Efthymios Kouloumpis, Theresa Wilson, and Johanna D Moore. 2011. Twitter sentiment analysis: The good the bad and the omg! *Proceedings of the 5th International Conference on Web and Social Media*, pages 538–541.

Saskia Le Cessie and Johannes C Van Houwelingen. 1992. Ridge estimators in logistic regression. *Applied statistics*, pages 191–201.

Percy Liang. 2005. *Semi-supervised learning for natural language*. Ph.D. thesis, Massachusetts Institute of Technology.

Kun-Lin Liu, Wu-Jun Li, and Minyi Guo. 2012. Emoticon smoothed language models for twitter sentiment analysis. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, pages 1678–1684.

Michael Nofer and Oliver Hinz. 2015. Using twitter to predict the stock market. *Business & Information Systems Engineering*, 57(4):229–242.

Debora Nozza, Elisabetta Fersini, and Enza Messina. 2017. A multi-view sentiment corpus. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 273–280.

Nuno Oliveira, Paulo Cortez, and Nelson Areal. 2017. The impact of microblogging data for stock market prediction: using twitter to predict returns, volatility, trading volume and survey sentiment indices. *Expert Systems with Applications*, 73:125–144.

Alexander Pak and Patrick Paroubek. 2010. Twitter as a corpus for sentiment analysis and opinion mining. In *Proceedings of the Seventh conference on International Language Resources and Evaluation*.

Christos H. Papadimitriou, Hisao Tamaki, Prabhakar Raghavan, and Santosh Vempala. 1998. Latent semantic indexing: A probabilistic analysis. In *Proceedings of the Seventeenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 159–168.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543.

Ashwin Rajadesingan, Reza Zafarani, and Huan Liu. 2015. Sarcasm detection on twitter: A behavioral modeling approach. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, pages 97–106.

Kumar Ravi and Vadlamani Ravi. 2017. A novel automatic satire and irony detection using ensembled feature selection and data mining. *Knowledge-Based Systems*, 120:15–33.

Antonio Reyes, Paolo Rosso, and Tony Veale. 2013. A multidimensional approach for detecting irony in twitter. *Lang. Resour. Eval.*, 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*, pages 502–518.

Emilio Sulis, Delia Irazú Hernández Farías, Paolo Rosso, Viviana Patti, and Giancarlo Ruffo. 2016. Figurative messages and affect in twitter: Differences between# irony,# sarcasm and# not. *Knowledge-Based Systems*, 108:132–143.

Cynthia Van Hee, Els Lefever, and Véronique Hoste. 2018. SemEval-2018 Task 3: Irony Detection in English Tweets. In *Proceedings of the 12th International Workshop on Semantic Evaluation*, page to appear.

Tien Thanh Vu, Shu Chang, Quang Thuy Ha, and Nigel Collier. 2012. An experiment in integrating sentiment features for tech stock prediction in twitter. In *The COLING Workshop on Information Extraction and Entity Analytics on Social Media Data*, pages 23–38.