LDR at SemEval-2018 Task 3: A Low Dimensional Text Representation for Irony Detection

Bilal Ghanem
Universitat Politecnica de Valencia,
Spain
bigha@doctor.upv.es

Francisco Rangel
Universitat Politecnica de Valencia,
Spain
francisco.rangel@autoritas.es

Paolo Rosso
Universitat Politecnica de Valencia,
Spain
prosso@dsic.upv.es

Abstract

In this paper we describe our participation in the SemEval-2018 task 3 Shared Task on Irony Detection. We have approached the task with our low dimensionality representation method (LDR), which exploits low dimensional features extracted from text on the basis of the occurrence probability of the words depending on each class. Our intuition is that words in ironic texts have different probability of occurrence than in non-ironic ones. Our approach obtained acceptable results in both subtasks A and B. We have performed an error analysis that shows the difference on correct and incorrect classified tweets.

1 Introduction

With the existence of online social networks, a huge amount of information rapidly pervades, which attracting the attention of researchers to investigate the linguistic phenomenon that appears. One of these complex phenomenon is irony, where the speaker uses words that mean the opposite of the literal meaning and what others really think, especially in order to be funny[1]. Moreover, irony can be considered as a strategy intended to criticise or to praise (Hernández-Farías et al., 2015). The detection of irony recently is quite a hot research topic due to its importance for efficient sentiment analysis (Ghosh et al., 2015). Also, another figurative language device noticed recently is sarcasm, where the writer intend to offend someone rather than creating a humor situation. In many research works, irony and sarcasm are often viewed as the same language device, or they considered irony as an umbrella term that covers also sarcasm (Wang, 2013). Several approaches have been proposed to detect irony, where most of them have turned the problem into a binary classification task using a set of features. (Carvalho et al., 2009) proposed one of the first works on irony detection. They worked on the identification of a set of patterns to identify ironic sentences. The adopted features were the use of punctuation marks and emoticons. (Reyes et al., 2013) proposed a model that employed four types of conceptual features: signatures, unexpectedness, style and emotional scenarios. (Barbieri and Saggion, 2014) proposed a model using lexical features, such as frequency of rare and common terms, synonyms, adjectives, emoticons, punctuation marks, positive and negative terms. Their results showed that the most important features are structure, frequency and synonyms for detecting irony in multiple datasets. (Karoui et al., 2015) presented a model to detect irony using a vector composed of six main groups of features: surface features (such as punctuation marks), sentiment (positive and negative words), sentiment shifter (positive and negative words in the scope of an intensifier), shifter (presence a negation word or reporting speech verbs), opposition (sentiment opposition or contrast between a subjective and an objective proposition) and internal contextual (the presence of personal pronouns). (Reyes et al., 2012) also studied the effect of multiple features to distinguish ironic and non-ironic tweets messages. The adopted features include quantifiers of sentence complexity, morphosyntactic and semantic ambiguity, polarity, unexpectedness, emotional activation, imagery, and pleasantness of words. (Wallace et al., 2015) presented a way to approach verbal irony classification by exploits contextual features, specifically by combining noun phrases and sentiment extracted

[1] As defined in the Merriam Webster Dictionary, http://www.merriam-webster.com/dictionary/irony; accessed on Feb. 2018.

Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018), pages 531–536
New Orleans, Louisiana, June 5–6, 2018.

from comments using a dataset of comments collected from reddit site which is social news aggregation. Most of these features did well in detecting ironic sentences, where in general they relied on different types of high level features that use lexical resources, sentiment analysis methods or common terms occurrence. Based on these previous works, we investigate the using of low dimensional features extracted from a given text. LDR uses terms weights to represent the probability that each Twitter message belongs to a specific class (e.g. ironic vs. non-ironic). Our intuition is that words usage in ironic texts has different probability of occurrence than in non-ironic ones, and LDR is good at capturing these differences. LDR does not need external resources, hand-crafted features, and drastically reduces the dimensionality of the representation. This allows the method to deal with big data problems. In this paper, we present the participation of LDR in the SemEval-2018 task 3 on Irony Detection (Van Hee et al., 2018). The rest of the paper is structured as follows. In Section 2, the LDR method is presented. In Section 3, we discuss the results we have achieved in both tasks and further analyse the error in Section 4. For example, to know whether the terms usage changes depending on ironic and non-ironic tweets. Finally, we draw some conclusions in Section 5 together with future work proposals.

2 Low Dimensional Representation Description

We proposed the low dimensional representation (LDR) in (Rangel et al., 2017a). LDR has been used in multiple author profiling tasks (Rangel et al., 2017b; Litvinova et al., 2017), and especially in language variety identification (Franco-Salvador et al., 2015; Fabra et al., 2015). The key aspect of the LDR is the use of weights to represent the probability of each term belonging to each one of the different language varieties. In our approach we built a vector of features from a matrix of terms weights. Starting from a set of training documents, with using Tf-Idf weighting scheme, we built a matrix of terms weights where each row represents Tf-Idf terms weights of a document (specifically this row of term weights represents a unique class C of its document), and each column corresponds to a specific term. Therefore, we obtained another matrix where each term weight was built using the ratio between the weights of the documents belonging to a concrete language variety C and the total distribution of weights for that term t over other documents, where each column in the matrix represents a term distribution overall documents.

AVG	The average weight of a document is calculated as the sum of weights W(t,c) of its terms divided by the total number of vocabulary terms of the document.
STD	The standard deviation of the weight of a document is calculated as the root square of the sum of all the weights W(t,c) minus the average.
MIN	The minimum weight of a document is the lowest term weight W(t,c) found in the document.
MAX	The maximum weight of a document is the highest term weight W(t,c) found in the document.
PROB	The overall weight of a document is the sum of weights W(t,c) of the terms of the document divided by the total number of terms of the document.
PROP	The proportion between the number of vocabulary terms of the document and the total number of terms of the document.

Table 1: LDR features for each classification class

Then, a vector of features was built where each feature was obtained in a different way, Table 1 describes the used features in LDR. In this paper, we used LDR in order to investigate its performance in irony detection classification task. LDR was proposed for different application where its discriminative statistical features proved to be efficient for classification purposes. Therefore, in this task, we investigated the LDR efficiency in irony detection, where the implicit meaning of a sentence is required to identify the correct class type.

3 Experiments and Results

During the experiments, LDR was tested using different classifiers. In the following sections we will illustrate the experiments that we carried out. For the evaluation, we used both accuracy and macro-average F-score. Moreover, the error anal-

Dataset	# of tweets	# Positive	# Negative
SemEval-2018	3,834	1,911	1,923
(Karoui et al., 2017)	540	540	-
(Ptáček et al., 2014)	67,779	18,889	48890

Table 2: Number of positive and negative tweets in the used datasets - Task A.

ysis we did allowed us to further understand the behavior of LDR in irony detection.

3.1 Data

We trained our model using the provided task 3 dataset with other two datasets collected previously by other researchers. The task A training subset consists of 3,834 tweets, where 1,911 tweets labeled as irony and 1,923 as not. While in task B, the same number of tweets was used with different type of subcategories. The tweets distribution over the categories was as follows: 1,390 irony tweets with a polarity contrast (labeled as 1), 316 as situational irony tweets (labeled as 2), 205 irony tweets without a polarity contrast (labeled as 3), and finally 1,923 non-irony tweets (labeled as 0). The second dataset that we used was provided by (Karoui et al., 2017) by collecting tweets using the Twitter API. The authors searched a set of keywords for different topics, such as politics, sport, artists, locations, Arab Spring, environment, racism, health, social media. These topics have been discussed in the French and American media during a specific period. We used the English part of the dataset which consists of 540 tweets. The last dataset was created by (Ptáček et al., 2014). The dataset approximately consists of 67,800 tweets of sarcasm linguistic phenomenon. As in the previous dataset, they used the Twitter API to collect the tweets but looking for the "sarcasm" hashtag. Table 2 summarises the statistical numbers of the datasets. We have conducted several experiments by combining the described datasets.

3.2 Task A

We have tested LDR with different classifiers using the Weka toolkit with standard parameters and conducted 10-fold cross-validation to find out the classifier that achieves the best results. LDR has achieved 64% of accuracy and 65% of F-score with the DecisionStump classifier. To improve the results, we adopted the Majority Vote (MV) algorithm using the results that were generated by the other two datasets with the training subset, each combined with the training part of task A subset in the 10-fold cross-validation. To note, none of the datasets improves the results more than using the task A subset independently. In spite of that, combining with Karoui *et al.* dataset achieved its highest results with Multilayer Perceptron classifier, and with Ptacek *et al.* dataset with REPTree classifier. By applying MV, we improved by 0.6% in accuracy the previous results obtained only with the task A training subset. Accordingly, we submitted two different runs, constrained and not. In the constrained one, we used the DecisionStump classifier with the task A training subset independently, while in the unconstrained, we used MV combining the three described datasets.

3.3 Task B

In this task, we experimented several runs that are similar to task A experiments. Since we did not find such a dataset with the used four subcategories, no other subset involved in this task experiments. Therefore, 10-cross validation technique was used without involving any other dataset. For the accuracy, the MultiClassUpdateable classifier achieved the highest result with 60.68%, while for the macro average F-score BayesNet achieved 38%. Accordingly, we adopted the classifiers that have the highest results in F-score to apply MV where BayesNet, NaiveBayes and NaiveBayesUpdateable classifiers are involved. By applying MV, we got a value 39% of macro average F-score. Finally, when the test subset for each task was released, we submitted our runs for each task. For the task A, both constrained and unconstrained runs were submitted while for task B, only the constrained run was submitted. Upon that, LDR attained the results that are in Table 2.

Tasks	Run Type	Accuracy	F-score
Task A	Constrained	0.56	0.43
	Unconstrained	-	0.43
Task B	Constrained	0.46	0.23

Table 3: LDR classification results of Task A and B using Accuracy and Macro average F-score.

The classification results for both runs in task A achieved the same score in terms of F-score mea-

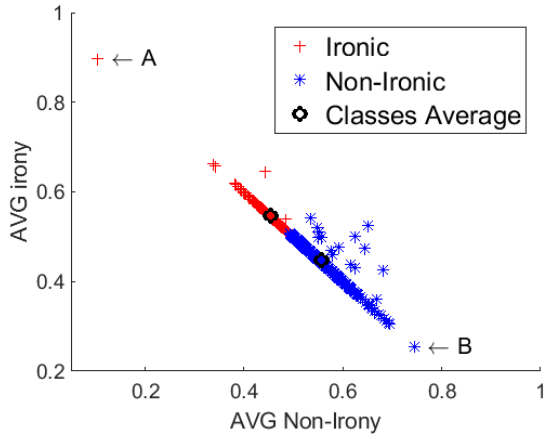

Figure 1: The distribution of **correctly classified** cases in term of AVG feature.

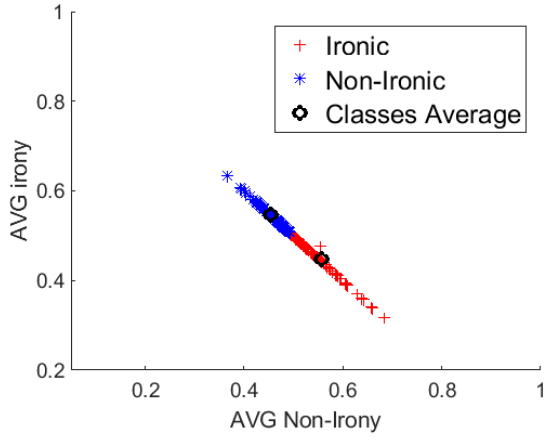

Figure 2: The distribution of **incorrectly classified** cases in term of AVG feature.

sure with some tiny progress to the unconstrained run. In the task B, LDR achieved a lower result comparing to task A. We believe that the reason is due to the number of training records for some classes in the training subset is small comparing to the other classes (unbalanced subset), where our model on these classes has a high bias (underfitting). Another possibility could be regarding to the LDR features, where maybe some of them could not be suitable for such task.

4 Error Analysis

We aim from analyzing the error of LDR to better understand the weak classification results that were obtained, especially in task B. We started with investigating the ability of LDR features to discriminate the data. So, we applied the GainRatio algorithm for features selection under the Weka toolkit to evaluate and to find out how much the LDR features are relevant to the irony training subset. The GainRatio result shows that LDR features were weak in discriminating the data, where the highest ranked attribute is the AVG feature for class 1 (ironic) with a value of 10%, followed by the AVG feature of class 0 (non-ironic) with 9%, where the rest of features are lower.

In the Fig. 1 and 2, we plotted the distribution of AVG feature of both correctly classified and incorrectly classified cases in test subset of task A, to show how they are distributed. In both figures, we can figure out that there is an overlapping between ironic and the non-ironic classes. To deduce which of them is more overlapped, we calculated the Euclidean distance (Ed) between the average points (the black circles) of both classes in each figure. As a result, the Ed in the Fig.1 of the correctly classified cases is 0.143, whilst in the second figure is 0.102. Therefore, the overlapping between both classes in the Fig.1 is lower than in the second figure, which clearly shows that the AVG feature is a good feature to infer both ironic and non-ironic classes.

As we discussed before, the LDR features are built based on a weighting scheme. Therefore, the larger is the training subset used, the more classification accuracy our model produces. To infer this fact, we investigated manually the AVG weights of two cases from the Fig. 1, to show how the weights are differentiated when: 1) a correctly classified ironic and non-ironic cases are far from each other in the figure, 2) and when they are near. Therefore, in the first case, we selected the cases A and B where they are far from each other. The tweets of A and B points in the figure are:

Ironic (A)

```
Yay jury duty #sarcasm
```

We found that the term ”Yay” was mentioned frequently in the ironic cases, where the writer used it as a figurative term to represent an irony situation. Meanwhile, this term was presented rarely in the non-ironic tweets. Therefore, this term made a distinctive situation for this tweet. The other two terms ”jury & duty” were not used in the process of weights building of the tweets, since we excluded terms that rarely appeared in the corpus.

Classes	Classified as				Correct
	0	1	2	3	
0	270	108	59	36	57%
1	36	104	19	5	64%
2	36	16	24	9	29%
3	32	9	17	4	6%

Table 4: Confusion matrix of the 4-class classification.

Non-Ironic (B)

```
Extended cut "NICHA"
https://t.co/qdzpcuRqc1
#trailer #spoof #film #dramatic
#action #film2014 #youtube
#fightscenes
```

Similarly in this tweet, the terms have a high probability of occurrence in the non-ironic tweets while a very small probability in the ironic tweets. For the second case, we took other two tweets from the overlapping area between the ironic and non-ironic tweets. The tweets' terms weights are very similar to each other where the terms in both sentences were mentioned. This is what makes the tweets AVG features near to each in the figure.

For the task B, we built a confusion matrix to clarify the predictions of test subset cases, as in Table 3. We can conclude that: in the classes that have a low number of training records (2 and 3), our model was highly confused in detecting their original labels, where a very small number of cases was classified correctly with a detection ratio of 29% and 6% for class 2 and 3 sequentially. Moreover, the highest confusion occurs from class 0 to class 1 where 108 cases were classified incorrectly, where the lowest was from class 1 to class 3. In general, both classes 1 and 2 were correctly classified better than 2 and 3 classes. In our view, our model did not fit the training subset for the classes 2 and 3.

5 Conclusion

In this paper, we have evaluated LDR on the irony detection task to investigate its performance on irony detection. Using LDR the classification task accomplished with accepted results without using any semantic, sentiment or contextual features. Despite the fact that LDR previously showed its competitive results on language variety identification and author profiling tasks and outperformed traditional state-of-the-art representations, from the results of this shared task we can conclude that the low dimensionality features do not perform as good as other language dependent features do, and from our point of view, they are not suitable to infer of such language phenomenon as irony. As future work we will continue studying how LDR will perform on other language applications.

Acknowledgement

This research work was done in the framework of the SomEMBED TIN2015-71147-C2-1-P MINECO research project.

References

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.

Paula Carvalho, Luís Sarmento, Mário J Silva, and Eugénio De Oliveira. 2009. Clues for detecting irony in user-generated contents: oh...!! it's so easy;-. In *Proceedings of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion*, pages 53–56. ACM.

Raül Boluda Fabra, Francisco Rangel, and Paolo Rosso. 2015. Nlel upv autoritas participation at discrimination between similar languages (dsl) 2015 shared task. In *Proceedings of the Joint Workshop on Language Technology for Closely Related Languages, Varieties and Dialects*, pages 52–58.

Marc Franco-Salvador, Francisco Rangel, Paolo Rosso, Mariona Taulé, and M Antònia Martít. 2015. Language variety identification using distributed representations of words and documents. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 28–40. Springer.

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 (SemEval 2015)*, pages 470–478.

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.

Jihen Karoui, Farah Benamara, Véronique Moriceau, Nathalie Aussenac-Gilles, and Lamia Hadrich Belguith. 2015. Towards a contextual pragmatic model to detect irony in tweets. In *53rd Annual Meeting of the Association for Computational Linguistics (ACL 2015)*, pages PP–644.

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*, volume 1, pages 262–272.

Tatiana Litvinova, Francisco Rangel, Paolo Rosso, Pavel Seredin, and Olga Litvinova. 2017. Overview of the rusprofiling pan at fire track on cross-genre gender identification in russian. *Notebook Papers of FIRE*, pages 8–10.

Tomáš Ptáček, Ivan Habernal, and Jun Hong. 2014. Sarcasm detection on czech and english twitter. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 213–223.

Francisco Rangel, Marc Franco-Salvador, and Paolo Rosso. 2017a. A low dimensionality representation for language variety identification. *arXiv preprint arXiv:1705.10754*.

Francisco Rangel, Paolo Rosso, Martin Potthast, and Benno Stein. 2017b. Overview of the 5th author profiling task at pan 2017: Gender and language variety identification in twitter. In *CLEF 2017 Labs and Workshops, Notebook Papers. CEUR Workshop Proceedings*, volume 1866.

Antonio Reyes, Paolo Rosso, and Davide Buscaldi. 2012. From humor recognition to irony detection: The figurative language of social media. *Data & Knowledge Engineering*, 74:1–12.

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

Cynthia Van Hee, Els Lefever, and Vronique Hoste. 2018. Semeval-2018 task 3: Irony detection in english tweets. In *Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018)*.

Byron C Wallace, Eugene Charniak, et al. 2015. Sparse, contextually informed models for irony detection: Exploiting user communities, entities and sentiment. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 1035–1044.

Po-Ya Angela Wang. 2013. # irony or# sarcasma quantitative and qualitative study based on twitter. In *Proceedings of the 27th Pacific Asia Conference on Language, Information, and Computation (PACLIC 27)*, pages 349–356.