

UO-CERPAMID at IroSvA: Impostor Method Adaptation for Irony Detection

IroSvA@IberLEF 2019

Daniel Castro¹ and Llinet Benavides²

¹ Center for Pattern Recognition and Data Mining, Universidad de Oriente, Cuba

daniel.castro@cerpamid.co.cu

² Universidad de Oriente, Cuba

llinet@uo.edu.cu

Abstract. Irony in text allows expressing implicit negative opinions, using a figurative and humorous language. IroSvA is a challenging task proposed this year that allows the evaluation of Irony Detection algorithms in the analysis of short texts for three Spanish language variants (Cuban, Spain and Mexican). Our proposal focuses on the study of three different representations of textual information and similarity measures using a weighted combination of these representations. We use an adaptation of the impostors method for classifying texts in Ironic or Non ironic. We consider the non-ironic texts of the training dataset as the list of impostors. The results achieved are encouraging and the best were obtained for the cuban variant.

Keywords: Irony detection · Impostor method · Text similarity

1 Introduction

Irony is a fundamental rhetorical device. It is a uniquely human mode of communication, curious in that the speaker says something other than what he or she intends [16]. Irony is an active part in the speech of users on the web, when it comes to expressing their opinions (blogs, forums, social networks, specialized sites). Hence the importance of its computational detection for the analysis of data by companies with access to data generated by these or entities with an interest in sentiment analysis, among others. The detection of irony is defined as “a set of characteristics and techniques that allow you to decide whether a text is ironic or not” [12]. Another definition is “Irony detection is an interesting machine learning problem, because, in contrast to most text classifications tasks, it requires a semantics that cannot be inferred directly from word counts over documents alone” [16].

As such, modeling irony has a large potential for applications in various research

Copyright © 2019 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0). IberLEF 2019, 24 September 2019, Bilbao, Spain.

areas, including text mining, author profiling, detecting online harassment and, perhaps one of the most investigated applications at present, automatic sentiment analysis. [4].

Therefore, detecting irony involves the most challenging areas of natural language processing. Research in user-generated content is a very challenging task. Despite the social media texts convey an invaluable source of information, they are difficult to process because they are noisy, informal, with little context, and plenty of grammatical mistakes [3].

In the task of detecting irony, although in its beginnings, the English language [14,1,3] was the most studied. There are already works in other languages such as Dutch [8], Catalan [9], Arabic [7] and still few are considered in Spanish [12,6]. SemEval 2018 Shared Task[5] on “irony detection in tweets” and Ironita 2018 [2] were the first tasks in irony detection. In SemEval 2018 “The systems that were submitted represent a variety of neural-network-based approaches (i.e. CNNs, RNNs and (bi-)LSTMs) exploiting word and character embedding as well as handcrafted features. Other popular classification algorithms include Support Vector Machines, Maximum Entropy, Random Forest, and Naive Bayes. While most approaches were based on one algorithm, some participants experimented with ensemble learners (e.g. SVM + LR, CNN + bi-LSTM, stacked LSTMs), implemented a voting system or built a cascaded architecture (for Task B) that first distinguished ironic from nonironic tweets and subsequently differentiated between the fine-grained irony categories.”[5].

IroSvA (Irony Detection in Spanish Variants) is the first shared task fully dedicated to identify the presence of irony in short messages (tweets and news comments) written in Spanish. [10].

The task will be structured into three subtasks, each one of these, for predicting whether messages are ironic or not, in one of the three Spanish variants. The three subtasks aim to the same goal: participants should determine whether a message is ironic or not according to specified context.

Recent advances in detection of irony have shown that the supervised classification methodology with a great extent of feature engineering produces satisfactory indicators for irony or sarcasm. This methodology has been tested in short texts, such as product reviews, news commentaries and tweets [6]. Supervised Classification focused in determine the class of an object based on a set of known objects grouped by class.

2 Proposal presented

Our approach is based on the representation of the text using three different vectors of features. For classifying a new text in the Ironic or Non-ironic class, we used the proposed General Impostor Method (GIM) of [15], but with a simplified variation of the Impostor Method (IM) presented by [15]. The similarity between texts is defined by a weighted similarity considering the three vector representation. In the next sections we explain the representation, the weighted similarity and the GIM. The IM uses a set of Non-ironic documents related with

the Ironic sample and analyze the similarity of an unknown document with the set and the Ironic sample. The unknown document would be Ironic if it is more similar to the Ironic one than a randomly subset of the Non-ironic documents.

2.1 Document representation

A text (document) is modeled as an object with three vector representations, but using some features in common. The idea behind this type of representation is to adjust what representation could impact for different genres of text.

One representation, and denoted by R_1 , is based on the classical Bag of Word (BoW) vector and we build this with the tokens extracted from a Natural Language tokenizer or with lemmas extracted using a Natural Language lemmatizer. The second representation, R_2 , is based only on the frequency of Punctuation Sign, considering all punctuation extracted by a Part of Speech Tagger. For the third representation, R_3 , are computed different stylistic features and the frequency of them. Some of the stylistic features are: entire capitalized words (QUE BUENO), character flooding (ooooohhhhhh), repetition of closed exclamation sign (!!!!!!!), etc.

2.2 Similarity measures

The similarity measures to compare two documents needs to consider all the representations proposed, but also to be flexible, so that we can use only some of them. For that reason we use the following measure.

$$\beta(D_1, D_2) = \alpha * \beta^1(D_{1R_1}, D_{2R_1}) + \gamma * \beta^2(D_{1R_2}, D_{2R_2}) + \delta * \beta^3(D_{1R_3}, D_{2R_3}) \quad (1)$$

For the β^* similarity function we implemented different similarities proposed in the literature, for example, cosine, jaccard, dice, tanimoto or distance like euclidean or minmax. In the evaluation phase we tested with all of them and used the one that allows us to obtain the best result for each representation.

The parameters α, γ, δ , let us give importance to the representation, and if one of the parameters is 0, then, not considering that representation. $\alpha + \gamma + \delta = 1$.

2.3 Impostor method

We used the proposed General Impostor Method (GIM), but not considering in the IM the step of randomly choose a subset of features from the representation of text.

The set of impostors S corresponds to the set of non-ironic documents provided in the training dataset. D_1 is the document to be classified and D_2 is an ironic document. Δ^* is a parameter that needs to be adjusted using the training dataset and allows us to determine that D_1 is ironic if it has more similarity with ironic D_2 than a percentage of non-ironic impostors.

Impostors Method (IM)

Input: (D_1, D_2) A pair of documents. S : a set of impostors documents

Output: (ironic) or (non-ironic)

1. $Score = 0$
2. **Repeat** k times
 - a. Randomly choose n impostors from $S : I_1, \dots, I_n$
 - b. $Score+ = \frac{1}{k}$ **if** $\beta(D_1, D_2) * \beta(D_2, D_1) > \beta(D_1, I_i) * \beta(D_2, I_i)$,
for each $i \in \{1 \dots n\}$
3. **return** (ironic) **if** $Score > \Delta^*$; **else** (non-ironic)

The set of Non-ironic texts are considered the impostors texts and for the classification of the evaluation dataset we used all the Non-ironic texts provided in the training dataset. All the parameters were optimized and the best value used for the evaluation phase and resumed in the Evaluation section. Next we present the pseudo-code of IM and GIM.

General Impostors Method (GIM)

Input: (D) A document to be classified. $Y : (D_1, \dots, D_n)$ ironic documents

Output: (ironic) or (non-ironic)

1. **For each** pair of documents (D, D_i)
 - a. Run original IM to obtain a similarity binary score $S(D, D_i)$
2. $Score =$ Average over similarity scores $[S(D, D_1), \dots, S(D, D_n)]$
3. **return** (ironic) **if** $Score > \theta^*$; **else** (non-ironic)

3 Evaluation results

The evaluation was executed for the three variants of Spanish language presented by the task, and for each of them we needed to optimize the parameters of the GIM and the weighted similarity measure. For that purpose we run a 10 cross fold validation over the training dataset provided. The range of the parameters evaluated was varied and the best parameter chosen is illustrated in the Table 1(parameters of UO-run2).

Table 1. Parameters optimization for the three Spanish variants over the training dataset

parameters	ES	MX	CU
K	5.0	5.0	5.0
S	5.0	5.0	5.0
Δ	0.6	0.6	0.7
θ	0.5	0.5	0.5
α	0.2	0.8	0.2
γ	0.8	0.2	0.0
δ	0.0	0.0	0.8

The structure and data distribution for the training and test datasets were presented by [10], and also the metrics used for evaluation. The similarity between documents was calculated considering the euclidean distance function for each of the three representation, because it gets the best results in the cross validation evaluation over the rest of the comparison functions implemented.

3.1 Discussion

In the Table 2, we present in row 1 and row 2 the results achieved by the two runs sent to the task, and all the baselines exposed by the organizers, and the highest value obtained by[13]. The main difference of UO-run1 and UO-run2 is that in run1 the BoW representation takes as features the token extracted by the Freeling[11] NLP tokenizer and in run2 the BoW representation considers as features the lemma extracted by the FreeLing lematizer. Also the parameters Δ and θ were 0.5 for UO-run1.

Table 2. F1-macro for the three Spanish variants test corpus

approach	ES	MX	CU	AVG
UO-run1	0.5110	0.4890	0.4996	0.4999
UO-run2	0.5445	0.5353	0.5930	0.5576
LDSE	0.6795	0.6608	0.6335	0.6579
W2V	0.6823	0.6271	0.6033	0.6376
Word nGrams	0.6696	0.6196	0.5684	0.6192
Majority	0.4000	0.4000	0.4000	0.4000

Our best result was achieved for UO-run2, and for the cuban variant, also similar to those of two of the baselines. It is important to notice that with lemma representation the results were always better than the representation based only on lexical tokens, because in the first one we reduced the lexical variety of words referring to the same lemma.

4 Conclusions and Future Work

In general, based on the results achieved in the cross validation from the training dataset of the Spain and Mexican dataset (spanish tweets), the R_3 representation gets the worse results and this is based on the stylistic variety between ironic tweets, and also the similarity in stylistic features between ironic and non-ironic tweets.

As future directions, we will introduce representations based on Word nGrams and feature selection methods based on the importance of features by class.

References

1. Barbieri, F., Saggion, H.: Modelling irony in twitter: Feature analysis and evaluation. In: Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26-31, 2014. pp. 4258–4264 (2014), <http://www.lrec-conf.org/proceedings/lrec2014/summaries/231.html>
2. Cignarella, A.T., Frenda, S., Basile, V., Bosco, C., Patti, V., Rosso, P.: Overview of the EVALITA 2018 task on irony detection in italian tweets (ironita). In: Proceedings of the Sixth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2018) co-located with the Fifth Italian Conference on Computational Linguistics (CLiC-it 2018), Turin, Italy, December 12-13, 2018. (2018), <http://ceur-ws.org/Vol-2263/paper005.pdf>
3. Fariás, D.I.H.: Irony and Sarcasm Detection in Twitter: The Role of Affective Content. phdthesis, Universidad Politécnica de Valencia (2017)
4. Hee, C.V.: Exploring automatic irony detection on social media. phdthesis, Universidad de Gent (2017)
5. Hee, C.V., Lefever, E., Hoste, V.: Semeval-2018 task 3: Irony detection in english tweets. In: Proceedings of The 12th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2018, New Orleans, Louisiana, USA, June 5-6, 2018. pp. 39–50 (2018), <https://aclanthology.info/papers/S18-1005/s18-1005>
6. Jasso, G., Meza-Ruíz, I.V.: Character and word baselines systems for irony detection in spanish short texts. *Procesamiento del Lenguaje Natural* **56**, 41–48 (2016), <http://journal.sepln.org/sepln/ojs/ojs/index.php/pln/article/view/5285>
7. Karoui, J., Zitoune, F.B., Moriceau, V.: SOUKHRIA: towards an irony detection system for arabic in social media. In: Third International Conference On Arabic Computational Linguistics, ACLING 2017, November 5-6, 2017, Dubai, United Arab Emirates. pp. 161–168 (2017). <https://doi.org/10.1016/j.procs.2017.10.105>, <https://doi.org/10.1016/j.procs.2017.10.105>
8. Liebrecht, C., Kunneman, F., van den Bosch, A.: The perfect solution for detecting sarcasm in tweets #not. In: Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, WASSA@NAACL-HLT 2013, 14 June 2013, Atlanta, Georgia, USA. pp. 29–37 (2013), <http://aclweb.org/anthology/W/W13/W13-1605.pdf>
9. Muñoz, J.R.: TwIrony: Identificación de la ironía en Tweets en Catalán. candthesis, Universitat Pompeu Fabra (2015)
10. Ortega-Bueno, R., Rangel, F., Hernández Fariás, D.I., Rosso, P., Montes-y-Gómez, M., Medina Pagola, J.E.: Overview of the Task on Irony Detection in Spanish Variants. In: Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2019), co-located with 34th Conference of the Spanish Society for Natural Language Processing (SEPLN 2019). CEUR-WS.org (2019)
11. Padró, L., Stanilovsky, E.: Freeling 3.0: Towards wider multilinguality. In: Proceedings of the Eighth International Conference on Language Resources and Evaluation, LREC 2012, Istanbul, Turkey, May 23-25, 2012. pp. 2473–2479 (2012), <http://www.lrec-conf.org/proceedings/lrec2012/summaries/430.html>
12. Pinto, M.: Modelo de Detección Automática de Ironía en Textos en Español. mathesis (2017)
13. Rangel, F., Rosso, P., Franco-Salvador, M.: A low dimensionality representation for language variety identification. In: 17th International Conference on Intelligent Text Processing and Computational Linguistics, CICLing’16. Springer-Verlag, LNCS(9624), pp. 156-169 (2018)

14. Reyes, A., Rosso, P., Veale, T.: A multidimensional approach for detecting irony in twitter. *Language Resources and Evaluation* **47**(1), 239–268 (2013). <https://doi.org/10.1007/s10579-012-9196-x>, <https://doi.org/10.1007/s10579-012-9196-x>
15. Seidman, S.: Authorship verification using the impostors method notebook for PAN at CLEF 2013. In: *Working Notes for CLEF 2013 Conference*, Valencia, Spain, September 23-26, 2013. (2013), <http://ceur-ws.org/Vol-1179/CLEF2013wn-PAN-Seidman2013.pdf>
16. Wallace, B.C.: Computational irony: A survey and new perspectives. *Artif. Intell. Rev.* **43**(4), 467–483 (2015). <https://doi.org/10.1007/s10462-012-9392-5>, <https://doi.org/10.1007/s10462-012-9392-5>