A Contextualized Word Representation Approach for Irony Detection

Lizeth García[1], Daniela Moctezuma[2], and Víctor Muñiz[1]

[1] Centro de Investigación en Matemáticas A.C., Monterrey, NL. México.
http://www.cimat.mx/es/Monterrey
[2] Centro de Investigación en Ciencias de Información Geoespacial A.C., Aguascalientes, México.
http://www.centrogeo.org.mx

Abstract. IroSvA (Irony Detection in Spanish variants) is a contest dedicated to identify the presence of irony in a given context in short messages written in Spanish, specifically tweets and news comments. In this case, it is necessary to consider that irony may be expressed in a different way according to the Spanish variant, which makes this task more complex to tackle it. Taking into account that irony detection is a very important task in many applications, we proposed a system based on a distributed representation of the texts, using the ELMo approach. We did not use any handcrafted features, lexicons or external datasets as prior information.

1 Introduction

Irony is a sophisticated way of communication, characterized by the speaker saying something different, generally the opposite, than what he or she means to. In the case of texts, this becomes more complicated due to lack of gestures such as facial expressions or variations in the voice or tones. The automatic detection of irony has been widely studied, mainly in English, although there are a variety of works for other languages. Different approaches have been proposed to identify irony in texts. In general terms, we can identify those models based on bag of features (lexical and/or semantic) and those based on Neural Network Language Models, which includes Deep Leaning and Recurrent Neural Networks, among many others architectures.

The initial approaches proposed to detect irony were mainly based on rules or lexical features. Most of the attempts were made using classification models which relied on textual cues such as lexical indicators like punctuation symbols, interjections, quotation marks [3], emotional scenarios and style features as textual sequences, such as character n-grams, skip-grams, and polarity s-grams [12], among others. However, these methods cannot utilize contextual information

from texts. Due to this, models based on semantic features have been developed to tackle the task of irony detection. In [1], semantic information has been incorporated into the feature representation like synonyms and rare words, while [6] uses word embeddings to capture context incongruity through the semantic similarity/discordance.

In recent years, the use of deep learning architectures for tracking irony in Twitter showed a significant improvement over traditional methods. A few representative works in this direction are based on LSTM architecture. In [2], the authors designs and ensembles two independent models, based on BiLSTM, which operate at the word and character level, in order to capture both the semantic and syntactic information in tweets. The proposal in [5] adopted the Siamese architecture to detect incongruity between different sections of a sentence, usual in the irony.

In this paper, we describe our system submitted to IroSvA shared task dedicated to identifying the presence of irony in short messages [9]. We propose a word-level representation in order to exploit the semantic information of each text, by using Contextualized Word Vectors. Specifically, we use the Deep Contextualized Word Representations ELMo (Embeddings from Language Models) from [11], which use bidirectional language model (biLM) to learn both word (e.g., syntax and semantics) and linguistic context (i.e., to model polysemy). For this purpose, we employ pre-trained ELMo vectors to predict whether a tweet is ironic or not for each Spanish variant.

It is worthwhile to say that, in order to tackle this contest, we explore different models based on different features extracted from texts, i.e., lexical, semantic and combination of both. Lexical features were similar to those described before, and for the semantic ones, we explored architectures for word and document embeddings based on the proposals of [8] and [7]. After an extensive set of experiments, and using different classification methods, we choose the model with the best results, and is the one we report in this paper.

In the followings sections, all the details of data and the system proposed are explained.

2 Dataset preparation and resources

The dataset used in this work is provided by the contest organizers. This corpus consist of $9,000$ tweets about different topics, $3,000$ for each Spanish variant from Cuba, Mexico, and Spain. Approximately, 80% of the corpus is used for training purposes, while the remaining 20% is used for test. The detailed statistics of the train data in each subtask are shown in Table 1. In this Table it can be see the number of topics in each Spanish variant, as well as the number of samples with and without irony.

Table 1. Description of dataset for training

Variant	Topics	No Irony	Irony
Mexico	10	1600	800
Spain	10	1600	800
Cuba	9	1600	800

2.1 Preprocessing

Preprocessing steps are essential to any text classification task. In this work we apply several standard preprocessing techniques. That is, for each tweet, we made several processes in the following order:

- Tokenization. We use a word representation in order to exploit semantic and linguistic information of each word in order to predict irony in text.
- In the case of Mexico and Spain variants, @user from the tweets are replaced with a unique special tag.
- We omitted URLs, emails and numbers.
- Hashtags are processed removing the special symbol # and keeping the text.
- Stop words are not removed.
- As final step, we convert all characters to lowercase.

2.2 Word Embeddings

We use the ELMo pre-trained word embeddings provided by [4], which were trained with a corpus of 20 million-words randomly sampled from the raw text released by the CoNLL 2018 shared task (wikidump + common crawl) for Spanish language. One of the main characteristics of ELMo (different from other word embeddings approaches), is that we can obtain multiple embeddings for each word depending on the context it was used. Higher-level layers capture context-dependent aspects of word embeddings while lower-level layers capture model aspects of syntax. In our case, we only use the top layer (layer number 0) for each word, and then we use the average as a function to group the words within a sentence.

For training purposes, the train dataset was split into 5 folds for each subtask. Our experimentation is performed using several classifiers: Support Vector Machines, Random Forests, and Logistic Regression, for which we used the scikit-learn implementation [10]. As we consider a semantic approach, we implemented a Baseline based on a distributed representation using doc2vec to obtain the embeddings for each tweet, and Random Forests as the classifier.

3 Experiments and Results

For evaluation purposes, we use the macro F1 average metric to select the classifier among the options mentioned before and compare the performance of our

model against this baseline. Table 2 shows the results for the 5-Fold Cross Validation approach for each classifier, the best performance in each subtask is obtained with Logistic Regression. In comparison with the baseline, there is a slight improvement when we use of ELMo vectors for the variants of Cuba and Mexico, however, in the case of Spain, the performance of the ELMO model does not exceed the baseline. It is important to note that Random Forest has a good performance when we use the embeddings obtained with doc2vec but not with the embeddings obtained with ELMO.

Table 2. Average CV F1-score for baseline and our model.

Model	Classifier	Cuba	Mexico	Spain
doc2vec	Random Forests	0.5795	0.5946	0.6718
ELMo	Logistic Regression	0.6351	0.5951	0.6542
ELMo	Support Vector Machines	0.6308	0.5930	0.6511
ELMo	Random Forests	0.4949	0.4632	0.4823

For each sub tasks, our model is compared with the official baseline. The best result we obtained, as it is shown in Table 3, is 0.6396 for the Cuba dataset. In the case of Mexico, the W2V and Word nGrams models perform similarly to our model with respect to F1 metric. The worst result we attained is in the Spain dataset. The results obtained in the test dataset are consistent with those obtained in the experiments.

Table 3. Baseline comparison and performance of our model for each Spanish variant F1-score. The first four rows are the official baselines.

Model	Spain	Mexico	Cuba	AVG
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	0.4	0.4	0.4	0.4
LabGeoCi	0.6251	0.6121	0.6396	0.6256

4 Conclusions

Detection of irony has been widely studied in recent years. However, works focused on the detection of irony for Spanish texts are still scarce, and there are no references to the complexity of this task for Spanish. It is necessary to take into account that IroSvA task also seeks to study the way irony changes in different variants of the language. We developed a system based on contextualized word representation, which seeks to establish the relationship of a word within its context, and thus, to identify irony.

Our proposal achieved an average F1 score of 0.6256, which is below the semantic baseline of W2V. For the Cuban texts, we obtained an F1 score superior to all the baseline, which we believe is due to the difference between the source of information of the Cuban texts compared to those of Mexico and Spain. The texts extracted from Twitter (the case of Mexico and Spain) usually have more noise than those texts that come from news comment due to the reduction of words by the limit of characters per tweet and by the use of resources such as images, emojis and references to another tweets to complement the context of the message.

Our model had a lower performance in texts that come from Twitter, which we believe is due to the fact that the semantic representations we used, based on words, does not represent the way people write on social networks, where errors, misspellings, abbreviations or repetition of letters to emphasize a word are very common.

As future work, we want to prepare and use a corpus based on Twitter texts in Spanish in order to train word embedding methods. We believe that the performance reported in this paper can be improved by using the embedding trained in this way. Also, we plan to explore character-level embedding models that can be tested in combination with lexical features for further improvement of the results.

References

1. Barbieri, F., Ronzano, F., Saggion, H.: Italian irony detection in twitter: a first approach. In: The First Italian Conference on Computational Linguistics CLiC-it. p. 28 (2014)
2. Baziotis, C., Athanasiou, N., Papalampidi, P., Kolovou, A., Paraskevopoulos, G., Ellinas, N., Potamianos, A.: Ntua-slp at semeval-2018 task 3: Tracking ironic tweets using ensembles of word and character level attentive rnns. arXiv preprint arXiv:1804.06659 (2018)
3. Carvalho, P., Sarmento, L., Silva, M.J., De Oliveira, E.: 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. pp. 53–56. ACM (2009)
4. Che, W., Liu, Y., Wang, Y., Zheng, B., Liu, T.: Towards better UD parsing: Deep contextualized word embeddings, ensemble, and treebank concatenation. In: Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. pp. 55–64. Association for Computational Linguistics, Brussels, Belgium (October 2018)
5. Ghosh, A., Veale, T.: Ironymagnet at semeval-2018 task 3: A siamese network for irony detection in social media. In: Proceedings of The 12th International Workshop on Semantic Evaluation. pp. 570–575 (2018)
6. Joshi, A., Tripathi, V., Patel, K., Bhattacharyya, P., Carman, M.: Are word embedding-based features useful for sarcasm detection? arXiv preprint arXiv:1610.00883 (2016)
7. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: International conference on machine learning. pp. 1188–1196 (2014)

8. Mikolov, T., Chen, K., Corrado, G.S., Dean, J.: Efficient estimation of word representations in vector space (2013)
9. Ortega-Bueno, R., Rangel, F., Hernández Farías, 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)
10. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
11. Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. arXiv preprint arXiv:1802.05365 (2018)
12. Reyes, A., Rosso, P., Veale, T.: A multidimensional approach for detecting irony in twitter. Language resources and evaluation **47**(1), 239–268 (2013)