PiLN IDPT 2021: Irony Detection in Portuguese Texts with Superficial Features and Embeddings

Rafael T. Anchiêta[1], Francisco A. Ricarte Neto[1], Jeziel C. Marinho[2], Kauan V. do Nascimento[2], and Raimundo S. Moura[2]

[1] Federal Institute of Piauí - IFPI
{rta, farn}@ifpi.edu.br
[2] Computer Department
Federal University of Piauí - UFPI
Teresina, Brazil
{jezielcm,kauanvazc,rsm}@ufpi.edu.br

Abstract. This paper describes the participation of the PiLN team in the IberLEF 2021 shared task on Irony Detection in Portuguese. The goal of this task is to create a system to detect irony in tweets and news texts. To deal with this topic, we develop a superficial feature-based strategy and an embeddings approach. Moreover, we adopt a back-translation method as data-augmentation to make the tweets *Corpus* less unbalanced. We evaluate our approaches within several machine learning-based classifiers and take the first and second places in the tweets *Corpus* category and seventh and thirteenth places in the news *Corpus*.

Keywords: Irony detection · Portuguese Tweets and News · Superficial features and embeddings.

1 Introduction

According to Oxford dictionary [14], irony may be viewed as a funny or strange aspect of a very different situation from what is expected, using words that say the opposite of what really mean, often as a joke, and with a tone of voice that shows this. In some circumstances, such as dialogue, other resources may assist in elaborating irony, such as a facial gesture, tone of voice, or movements with the arms. When dealing only with texts, the task of irony detection becomes even more challenging. Due to its nature, irony has important implications for Natural Language Processing (NLP) task, which aims to understand and produce human languages [15].

When dealing with textual content on the internet, it is possible to observe the use of ironies as a mechanism to reinforce an utterance or efficiently state

IberLEF 2021, September 2021, Málaga, Spain.

a point of view. Therefore, automatic irony detection may be useful for several tasks, such as detecting online harassment, identifying author profiling, opinion mining, and others [10]. Also, it may improve the results of the sentiment analysis classification task [11].

In this paper, we describe our strategies to deal with automatic irony detection in tweets and news texts in the IberLEF 2021 shared task on Irony Detection in Portuguese [2].

2 Irony Detection task in Portuguese

The organizers of this shared task released two *Corpora* with ironic texts: tweets and news. In the training set, the first *Corpus* has $15,212$ tweets, while the second one has $18,494$ news. Table 1 presents the number of ironic and non-ironic texts for each *Corpus*. As we can see, the *Corpora* are unbalanced. The *Corpus* of news is unbalanced regarding the ironic label, whereas the *Corpus* of tweets is unbalanced concerning the non-ironic label.

Table 1. Number of ironic and non-ironic texts.

Corpus	**Ironic**	**Non-ironic**
Tweets	$12,736$ (84%)	$2,476$ (16%)
News	$7,222$ (40%)	$11,272$ (60%)

From these *Corpora*, the goal of this task was defined as a binary classification problem where the participants were asked to classify a text (tweets and news) as ironic or non-ironic.

In what follows, we detail our strategy to handle ironic texts.

3 Developed strategies

In order to make the *Corpus* of tweets less unbalanced, we used the *Corpus* of the SemEval 2018 Task 3 (Irony Detection in English Tweets) [15]. That *Corpus* has $4,390$ tweets, being $2,147$ ironic and $2,243$ non-ironic. To translate tweets from English to Portuguese, we adopted a back-translation approach [12], as depicted in Figure 1.

From this figure, we translated the original tweets from the SemEval *Corpus* to Portuguese, using the machine translation model provided by the Google Translate API[3]. We used that model because it achieved good results in the semantic parsing [13], textual entailment inference [4], and paraphrase detection [1] tasks. In sequence, we translate the Portuguese tweets back to English, because

[3] `https://cloud.google.com/translate/`

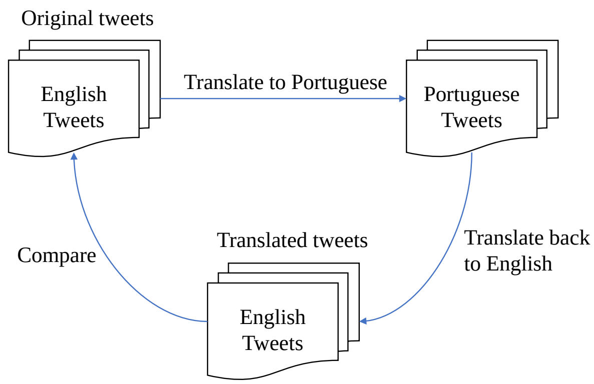

Fig. 1. Back-translation strategy.

there are no Portuguese reference tweets to evaluate the quality of the translations. In this way, we may measure the quality of the translations, comparing the original tweets with the back-translated tweets. To evaluate the quality of the translations, we computed the harmonic mean between the ROUGE[4] [7] and BLEU [8] metrics, as in Equation 1.

$$F_1 = 2 \times \frac{rouge \times bleu}{rouge + bleu} \tag{1}$$

Based on that strategy, we obtained f-score values as shown in Table 2, when comparing the original tweets with the back-translated tweets. From this table, 25%, 50%, and 75% refer to the first, second, and third quartile, respectively.

Table 2. Statistics obtained from the back-translation strategy.

Statistic	**Value**
Mean	0.70
Std	0.13
Min	0.00
Max	0.94
25%	0.62
50%	0.71
75%	0.79

Taking into account the values of Table 2, we got the non-ironic tweets with f-scores greater than 0.79, which equals 680 tweets corresponding to the best translations, to make the *Corpus* of tweets less unbalanced. Our new *Corpus* of tweets has $12,736$ ironic tweets and $3,156$ non-ironic tweets.

[4] We used the F-score of the ROUGE-L.

With the *Corpus* of news and the new *Corpus* of tweets, we adopted a method based on superficial features and an embedding approach to extract features of *Corpora*, train classifiers to predict tweets or news as ironic or non-ironic. In subsection 3.1, we detail our superficial feature method, and in subsection 3.2, we describe the embedding approach.

3.1 Superficial features

In this approach, we tried several linguistic features to identify ironic texts, such as: number of named entities, presence/absence of some symbols, expressions, number of emojis, frequent words, among others. Table 3 presents the analyzed features in the Tweets and News datasets.

Table 3. Analyzed features to identify ironic texts.

Feature name	**Description**
#tok	Number of tokens formed only by letters
#NE	Number of Named Entities in text
%misspelling	Percentage of wrong words
#misspelling	Number of wrong words
%uppercase	Percentage of words written in uppercase
has_quotes	Use of quotation marks in text
has_SQNexpr	Use of expressions: 'sqn', 'SQN', 'só que não', 'SÓ QUE NÃO', 'so que não', 'SO QUE NÃO', 'so que ñ', 'SO QUE Ñ', 'Só que não'
has_marks	Excessive use of question and exclamation marks
#emojis	Number of emojis in text
#hashtag	Number of hashtags in text
#smiley	Number of smiley face symbols
has_laughter	Use of laugh expressions in text: 'kkk', 'hahaha', 'hehehe', 'rsrsrs'
#affectiveAdj	Number of affective adjectives in text
freqWords	10 most frequent words in each class: ironic and non_ironic, excluding repeated words

From this feature list, we experimented several machine learning algorithms and observed that the most relevant feature is the $freqWords$. Thus, we adopted a very simple strategy to detect an ironic text. We computed the Term Frequency-Inverse Document Frequency (TF-IDF) to extract features from the Tweets and News *Corpora*, and we fed them into the Linear Support Vector Machine (SVM), using the Stochastic Gradient Descent (SGD) technique to train the classifier. Moreover, we applied a greedy search method to find the best parameters for the TF-IDF and the classifier. Table 4 presents the found out parameters for tweets and news. From this table, number of features is the vector size, max. df refers to the document frequency, and max. iterations is the number of iterations of the training algorithm.

Table 4. Best parameters for the TF-IDF and the classifier

	Twitter	News
Parameter	**Value**	**Value**
Number of features	26,202	64,194
Max. df	0.75	0.50
Max. iterations	100	50

3.2 Embeddings approach

According to Le and Mikolov [6], Paragraph Vector is an unsupervised method that learns a fixed length of features from variable forms of texts e.g., sentence, paragraph, document. The authors mention two approaches, Distributed Memory (DM) and Distributed Bag of Words (DBOW). The first one uses word vectors with a paragraph vector representation for the task of predicting the next word in a sentence. In DM, the paragraph vector representation acts like memory, granting context information to the prediction task. Unlike DM, DBOW uses only paragraph information. This method randomly samples words in the paragraph vector representation and uses this information to predict the next word in the text.

To extract embedding features from the *Corpora*, we trained the Distributed Bag of Words Paragraph Vector model [6] for each *Corpus* with the parameters, as shown in Table 5. We choose the DBOW approach based on better results obtained from initial experiments with two Paragraph Vector methods. To train the model, we used the Gensim library [9].

Table 5. Parameter settings of the Paragraph Vector model.

Parameter	Value	Parameter	Value
vector_size	300	dm	0
window	4	sample	1e−4
min_count	1	workers	5
epochs	300	alpha	0.025

From this table, *vector_size* is the dimensionality of vectors, i.e., each text has 300-dimensional embeddings, *dm* is the training algorithm, where the 0 value refers to the Distributed Bag of Words version of Paragraph Vector. We set 0.025 as initial learning rate on parameter *alpha*, 300 *epochs*, which is the number of iterations over the corpus, and choose to ignore words with lower frequency than 1 on parameter *min_count*. The parameter *window* represents the distance between the current and predicted word, defined by size 4 and for parameters *sample* and *workers*, we set empirical values based on initial experiments.

In the next section, we present our conducted experiments and obtained results.

4 Experiments and Results

To assess the systems for this shared task, the organizers made available test sets for the news and tweets *Corpora*, where each test set has 300 texts. The systems were ranked by the balanced accuracy metric (BACC) [3]. Each team could submit three runs for each *Corpus*. The shared task had seven teams and fifteen submissions. Out of 15 submissions, our strategies ranked $1st$ (superficial features + SVM) and $2nd$ (embeddings + MLP) for the tweets *Corpus*, and $7th$ (superficial features + SVM) and $13th$ (embeddings + MLP) for the news *Corpus*. Table 6 presents the obtained results by our methods.

Table 6. Results obtained by our strategies.

Approach	Classifier	*Corpus*	BACC	Accuracy	F-score	Precision	Recall
Superficial features	SVM	Tweets	0.52	0.47	0.55	0.42	0.80
Embeddings	MLP		0.51	0.46	0.55	0.41	0.80
Superficial features	SVM	News	0.80	0.77	0.76	0.64	0.93
Embeddings	MLP		0.71	0.74	0.63	0.69	0.58

In general, one can see that the superficial features approaches outperformed the embeddings strategy. The features have discriminated better an ironic text from non-ironic text than embeddings. Moreover, we believe that the back-translation method helped to produce the best results for the tweets *Corpus*.

The methods for the news *Corpus* achieved better scores than the tweets *Corpus*. We believe that it is due to two reasons: the news *Corpus* is larger than the tweets *Corpus* and the text style of the tweets since they are usually short, unstructured and seldom obey grammar and punctuation rules. Hence, generally, tweets texts are more challenging than news texts.

For the news *Corpus*, the embedding method seems to be unable to generalize well, likely because of the news length. This result suggests the necessity to investigate a more robust architecture. We believe that a transformer architecture, as BERT [5], may help to produce better results for news text.

It is important to highlight that simple approaches, as TF-IDF and embeddings, produce good results for the irony detection task. Our trained models are available at `https://github.com/rafaelanchieta/PiLN/`.

5 Conclusion

In this paper, we presented our strategies to IberLEF 2021 shared task on Irony Detection in Portuguese. We developed a method based on superficial features and an embedding-based approach. More than that, we used a back-translation strategy as data-augmentation to make the tweets *Corpus* less unbalanced. Our superficial feature approach ranked 1*st* and 7*th* and the embedding method ranked 2*nd* and 13*th* for the tweets and news *Corpora*, respectively. These results suggest that the data-augmentation strategy help to obtain the best results in the tweets *Corpus* and that it is necessary to investigate more robust approaches to the news *Corpus*.

References

1. Anchiêta, R.T., Sousa, R.F.d., Pardo, T.A.: Modeling the Paraphrase Detection Task over a Heterogeneous Graph Network with Data Augmentation. Information **11**(9), 422 (2020)
2. Brisolara Corrêa, U., Pereira dos Santos, L., Coelho, L., A. de Freitas, L.: Overview of the idpt task on irony detection in portuguese at iberlef 2021. Procesamiento del Lenguaje Natural **67** (2021)
3. Brodersen, K.H., Ong, C.S., Stephan, K.E., Buhmann, J.M.: The Balanced Accuracy and Its Posterior Distribution. In: Proceedings of the 20th International Conference on Pattern Recognition. pp. 3121–3124. IEEE, Istanbul, Turkey (Aug 2010)
4. Cabezudo, M.A.S., Inácio, M., Rodrigues, A.C., Casanova, E., de Sousa, R.F.: Natural language inference for portuguese using bert and multilingual information. In: Proceedings of the 14th International Conference on Computational Processing of the Portuguese Language. pp. 346–356. Springer International Publishing, Évora, Portugal (Mar 2020)
5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019)
6. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: Proceedings of the 31st International Conference on Machine Learning. pp. 1188–1196. JMLR, Beijing, China (Jun 2014)
7. Lin, C.Y.: ROUGE: A package for automatic evaluation of summaries. In: Text Summarization Branches Out. pp. 74–81. Association for Computational Linguistics, Barcelona, Spain (Jul 2004)
8. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics. pp. 311–318. Association for Computational Linguistics, Philadelphia, Pennsylvania, USA (Jul 2002)
9. Řehůřek, R., Sojka, P.: Software Framework for Topic Modelling with Large Corpora. In: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks. pp. 45–50. ELRA, Valletta, Malta (May 2010)
10. Reyes, A., Rosso, P., Buscaldi, D.: Humor in the blogosphere: First clues for a verbal humor taxonomy. Journal of Intelligent Systems **18**(4), 311 (2009)
11. Rosenthal, S., Ritter, A., Nakov, P., Stoyanov, V.: SemEval-2014 task 9: Sentiment analysis in Twitter. In: Proceedings of the 8th International Workshop on Semantic

Evaluation. pp. 73–80. Association for Computational Linguistics, Dublin, Ireland (Aug 2014)
12. Sennrich, R., Haddow, B., Birch, A.: Improving neural machine translation models with monolingual data. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 86–96. Association for Computational Linguistics, Berlin, Germany (Aug 2016)
13. Sobrevilla Cabezudo, M.A., Mille, S., Pardo, T.: Back-translation as strategy to tackle the lack of corpus in natural language generation from semantic representations. In: Proceedings of the 2nd Workshop on Multilingual Surface Realisation (MSR 2019). pp. 94–103. Association for Computational Linguistics, Hong Kong, China (Nov 2019)
14. Stevenson, A.: Oxford dictionary of English. Oxford University Press, USA (2010)
15. Van Hee, C., Lefever, E., Hoste, V.: SemEval-2018 task 3: Irony detection in English tweets. In: Proceedings of The 12th International Workshop on Semantic Evaluation. pp. 39–50. Association for Computational Linguistics, New Orleans, Louisiana (Jun 2018)