TeamUFPR at IDPT 2021: Equalizing a Strategy Using Machine Learning for Two Types of Data in Detecting Irony

Tiago Heinrich[1] [0000−0002−8017−1293], Fabrício Ceschin[1] [0000−0001−6853−8083], and Felipe Marchi[2] [0000−0002−7711−3498]

[1] Federal University of Paraná – Curitiba, Brazil
{theinrich,fjoceschin}@inf.ufpr.br
[2] Santa Catarina State University – Joinville, Brazil
felipe.ramos@edu.udesc.br

Abstract. This paper describes the participation of the TeamUFPR at the Task on Irony Detection in Portuguese (IDPT 2021), framed within the Iberian Languages Evaluation Forum (IberLEF 2021). The task consists of creating a methodology for irony detection in Portuguese using two datasets, one of them containing news texts obtained from different sources and the second being tweets collected on twitter. Our proposal focused mainly on using only one approach for both datasets, three tests were submitted using different strategies to identify the impact of the models considering the type of data. We evaluate a total of ten machine learning algorithms, with four feature selection strategies that explore a variety of parameters. Three strategy's were used in IDPT 2021, focusing in undersampling and lemmatization. Overall, the result was relatively pleasant with the best results being found by Multilayer Perceptron and Random Forest, and we were able to demonstrate a new approach to identifying irony in messages.

Keywords: Sentimental Analysis · Natural Language Processing · Machine Learning.

1 Introduction

Sentiment analysis focuses on extracting sentiment from texts found in sources such as news, social networks, or e-mails, to classify as positive or negative [1], or a more specific classification task, such as irony detection. Applications using *Natural Language Processing* (NLP) have been popularizing in recent years, with the widespread of solutions in both academia and industry.

The representation of texts or phrases (such as tweets or documents) by techniques that aim to analyze and explore new models of representation are

IberLEF 2021, September 2021, Málaga, Spain.

known as NLP [6]. This type of approach considers a language evaluation, with the objective of making an algorithm that understands this information in the most similar way to the understanding of a human being.

Over the years new strategies were developed with the focus in using *machine learning* (ML), that could take advantage of computational power. Unsupervised algorithms began to have more popularity in recent years, achieving adequate results to label a large amount of data.

The IDPT 2021 is the first IberLEF task turned to irony detection in Portuguese. The competition uses two sets of data crawled from the web, one presenting news and other, tweets for the competitors [4]. The IDPT is one of the tasks offers in the IberLEF 2021, in the section of humor and irony.

Our proposal focuses on exploring machine learning techniques for the learning phase, and recurring strategies for the preprocessing phase. Overall, our team explores a total of nine strategies in the preprocessing step, four on the feature extraction step, and ten algorithms in the learning step. The final approach consists in evaluating the average of the execution of 10-fold cross validation of each combination, looking for the strategy that best fits the two sets of data proposed for the competitors. The source code is available in `https://github.com/h31nr1ch/TeamUFPR-IDPT2021`.

The experience base of our team is varied, consisting of knowledge in machine learning facing security applications and protein structure prediction with metaheuristics. Our main motivation in participating in the competition is to use concepts already known and adapting them to sentiment analysis using NLP.

The reminder of this paper is structured as follows: Section 2 describes the IDPT 2021 task. Section 3 presents the methodology used. Section 4 explains our evaluation and algorithm choices. Section 5 presents the related work; and Section 6 concludes the paper.

2 Task description

The IberLEF 2021 has a focus on irony detection in the Portuguese language [4]. The task aims to identify the presence of irony in two sets of data (News and Tweets). The proposed dataset for competitors are show in Table 1. The train set consist of 15.2k tweets and 18.4k news, which must be used to classify 600 messages (found in the Test set) half representing each class.

Table 1: IDPT 2021 Datasets.

Dataset	**Train**	**Test**
Tweets	15,212	300
News	18,494	300

The competitors must provide an `id` and a `label`, that will be used to check the efficiency of their strategy. After that, the results will be presented by the following metrics: Bacc, Accuracy, F1, Precision, and Recall. Each team was allowed to submit three runs for each data set, making a total of six runs.

3 Methodology

In this section, we describe our methodology for the competition. Dividing the section in three groups, Section 3.1 preprocessing stage; Section 3.2 feature extraction process; and Section 3.3 our machine learning methodology.

3.1 Preprocessing stage

The first step consists of the preprocessing stage, focusing on cleaning up undesirable and irrelevant patterns. In total, nine preprocessing strategies were used, which are: (1) removal of all accented characters; (2) fix encoding found in some texts that were not utf-8; (3) remove tags from users or entities; (4) remove punctuation from the text; (5) remove special characters; (6) remove duplicate spacing; (7) change texts to lowercase; (8) remove numbers; and (9) removal of stop words (in this case we used nltk list of stop words [7]).

These nine steps are responsible for eliminating features that can be problematic for the feature extraction algorithms and later can harm the machine learning algorithms. After all this process, two additional preprocessing cases were defined, one representation using stemming (using the spaCy [10]) and one without, the focus here was on optimizing the results for the set of tweets. Our focus in the tweet set is due to the wide variation of phrases, along with the high number of slang used.

3.2 Feature extraction

For the feature extraction strategies, we considered and evaluated four methods. In the end, only two were selected. The algorithms used were: `CountVectorizer` (Token Counts); `TfidfVectorizer` (TF-IDF); `HashingVectorizer` (Hashing Trick); and `Word2Vec`, all of them from `scikit-learn` [9]. The choice was due to familiarity and past experiences with these extractors, given that they are widely used in the literature of NLP applications and we wanted to test their performance in irony detection tasks.

The feature extraction step will be responsible for converting the textual information into a format that is understandable for machine learning algorithms. Each of the four algorithms focuses on creating different types of outputs, according to their feature extraction strategy.

The number of features tested for `CountVectorizer`, `TfidfVectorizer`, and `HashingVectorizer` was their default parameters (found in scikit-learn documentation [9]), 10k, 20k, 30k, 40k, 50, 100k, and 200k. The feature dimension extracted from `Word2Vec` was 50, 100, 250, and 500.

3.3 Machine learning

With the train dataset, a total of ten algorithms were tested: (1) `Random Forest` (RF); (2) `Multilayer Perceptron` (MLP); (3) `sgd`; (4) `linearSVC`; (5) `svc`; (6) `decisionTree`; (7) `perceptron`; (8) `k-nearest neighbors` (KNN); (9) `multinomialNB`; and (10) `gaussianNB`.

For each algorithm we run tests considering the preprocessing and feature extraction stage, using each configuration presented. After all these steps, we checked if lemmatization could help detect irony by comparing classifiers trained with and without it.

At the end of all runs, the algorithms that presented the best results for the news dataset and tweets dataset were `Multilayer Perceptron` and `Random Forest`, respectively. Taking into account the set of tests we ran, in this section we focused only in presenting the best scenarios of our approach. The complete set of results are available on GitHub along with the source code.

4 Evaluation

In this section, we will discuss the evaluation of our methodology using only the train dataset. After this process we will present the strategy used in the IDPT 2021 task (Section 4.1). Our objective here is to choose the algorithm with the best average value for both classes (News and Tweets), also checking which settings are used in the respective data with the best result.

Figure 1, 2, 3, and 4 present the evaluation of the news dataset. These tests split the training dataset in 50/50 for train/test, and use the `Multilayer Perceptron`. The approach using `Word2Vec` didn't present an acceptable result (Fig. 2 and 4) in comparison with the other three approach that have achieved results above 96% (Fig. 1 and 3). We believe that it happens due to the use of a small dataset to train the `Word2Vec` model, which requires a lot of data to achieve better results. The `TfidfVectorizer` was the feature selection that presented the best results continuously, consequently it was the method chosen for the news set. The difference between using of lemmatization was quite small, but we decided not to use it considering that it presented the best average result.

Figure 5, 6, 7 and 8 present the evaluation of the tweets dataset. These tests also split the training dataset in 50/50 for train/test, and use the `Random Forest` classifier. `Word2Vec` present the worst results overall in this scenario, even with the use of lemmatization that helped the news set in this same scenario. The results presented in 5 and 7 highlights the best performance when not using and when using lemmatization with `HashingVectorizer`. Taking that into account, it was decided not to apply lemmatization again, which indicates that the inflected forms of a word might help to detect irony.

Table 2 show the size difference between both classes, found in the news and tweet datasets. Because of the high class imbalance, a strategy considering undersampling was considered, with the goal of balancing the dataset and avoid problems with the algorithms.

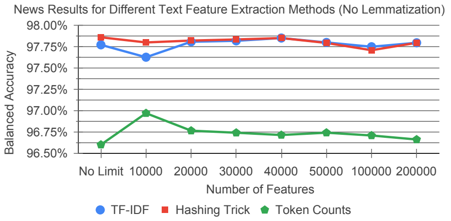

Fig. 1: News Results for Different Text Feature Extraction Methods (No Lemmatization).

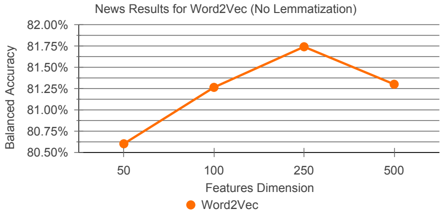

Fig. 2: News Results for Word2Vec (No Lemmatization).

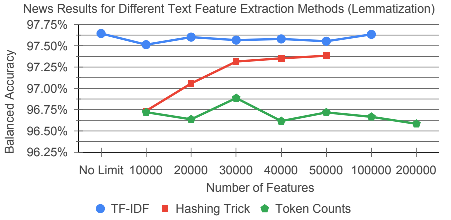

Fig. 3: News Results for Different Text Feature Extraction Methods (Lemmatization).

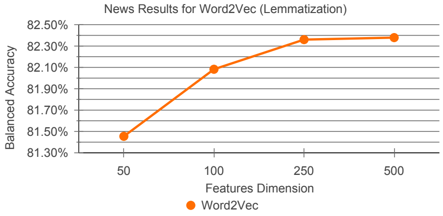

Fig. 4: News Results for Word2Vec (Lemmatization).

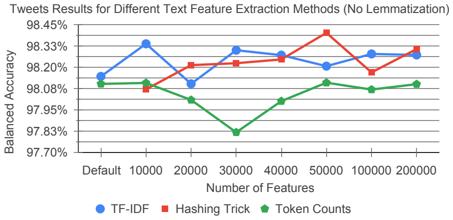

Fig. 5: Tweets Results for Different Text Feature Extraction Methods (No Lemmatization).

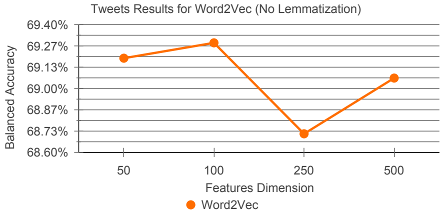

Fig. 6: Tweets Results for Word2Vec (No Lemmatization).

After all this process of evaluation, `HashingVectorizer` was choose for the tweets set with 50k for maximum features; and `TfidfVectorizer` was used for the news set with maximum features of 40k. Now for the unbalanced classes, two execution using undersampling were choose.

4.1 TeamUFPR strategy used in IDPT 2021

The IDPT 2021 task, allow teams to submit three runs. As we already had knowledge of the best algorithms with the results of Section 4, we focused on

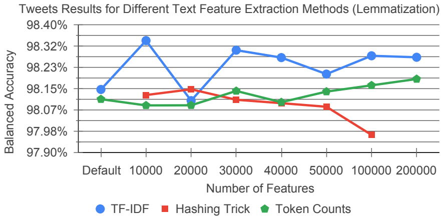

Fig. 7: Tweets Results for Different Text Feature Extraction Methods (Lemmatization).

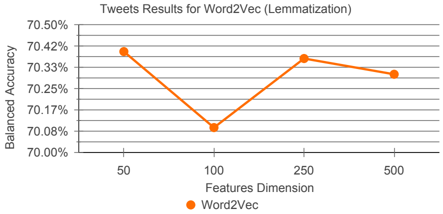

Fig. 8: Tweets Results for Word2Vec (Lemmatization).

Table 2: Data distribution inside each dataset.

Dataset	Irony (1)	Non irony (0)
Tweet	12,736	2,476
News	7,222	11,272

a strategy to deal with unbalanced classes. Since the dataset presents a high variation between the number of samples in each class.

The following configuration was define:

1. No undersampling strategy was used, the data consist of only the preprocessing stage and feature extraction;
2. Random undersampling was used to approximate the size of both classes; and
3. Random undersampling was used and a threshold of 0.9 was defined for minority class. This strategy had the objective of diversifying the options for the tweet set.

The random undersampling and the use of a threshold presented an impact in ours tests using just the train dataset. And we expected that the test dataset would be similar to the training dataset (as confirmed by the final results).

4.2 IDPT 2021 Results

Table 3 presents the results for the TeamUFPR, with the `dataset`, the `rank` that represents the overall position of the given run, `run` that represents the three approach's defined in section 3.3, and the metrics used to evaluate the teams.

Considering both datasets, our approach achieves the most stable results with little variation considering the results of the other teams, despite of being very different from the results we obtained in the train/test phase (which indicates that the datasets used to evaluate the solutions are very different from them). The strategy to treat the unbalanced classes ended up affecting the result. But for future experiments, we can point out that using lemmatization do not help to detect irony.

Table 3: IDPT 2021 results for TeamUFPR.

Dataset	Rank	Run	Bacc	Accuracy	F1	Precision	Recall
Tweets	5	1	0.50	0.41	0.58	0.41	1.0
	11	2	0.49	0.41	0.57	0.40	0.99
	17	3	0.42	0.38	0.46	0.36	0.64
News	5	1	0.83	0.82	0.78	0.71	0.87
	6	2	0.81	0.81	0.77	0.72	0.81
	10	3	0.78	0.79	0.73	0.72	0.74

5 Related Works

Four articles were used to guide our methodology. [5] presents a task perform in SemEval 2017, that had the goal of detecting sarcasm in sentences. The sentimental classification was made by a two-level classification system. The first phase used three strategies for the preprocessing of the data. The second phase, focus in identify key factors as affection, cognition, and sociolinguistics of the sentences.

[2] was a task in the HaSpeeDe 2018, that consists of the detection of hate speech in Italian social media. Three tasks were tenders to nine teams, that aim to find the best strategy for identifying hateful speeches. The document lists the general approaches used by each team and their results.

Focusing on irony detection [3] presents two tasks for identifying irony in sentences and the identification of types of irony. The competition received an overall of seventeen submissions, which were evaluated by their results, approaches, algorithms, and features.

Irony detection in Spanish is the focus in [8], which presents the first task for identifying irony in short messages IroSvA. Three tasks were defined for the irony detection, one case focusing on the identification in Spain tweets, another case with a focus on Mexican tweets, and the last focusing on Cuban news. A detailed set of strategies used by the competitors is presented, along with metrics to help the comparison of results.

6 Conclusion

In this paper we describe the participation of the TeamUFPR at the IDPT 2021 Task on Irony Detection in Portuguese. The task consisted in creating a methodology for irony detection in Portuguese using two datasets, one of them containing news texts obtained from different sources and the second being tweets collected on twitter. Our proposal focused mainly on using only one approach for both datasets, three tests were submitted using different strategies to identify the impact of the models considering the type of data.

Overall, we identified that TF-IDF was the best feature extraction option for the news dataset, and `HashingVectorizer` was the best option for the tweets dataset. The classifiers that presented the best results for the news dataset and tweets dataset were `Multilayer Perceptron` and `Random Forest`, respectively. Also, using random undersampling or ensemble classifiers (with and without the use of a threshold on classifier output) did not helped us to improve our classification results, which indicates that future work should focus on different strategies to fix the imbalanced data problem in irony detection, mainly in the tweets dataset. Finally, we also concluded that lemmatization is a step that should not be performed in detecting irony, indicating that the inflected forms of a word might help to detect it.

For future works, we believe that creating new feature extraction methods (such as BERT) and classifiers that consider imbalanced data, without using word lemmatization, are key to improve classification performance of irony detection.

Acknowledgments This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES). The authors also thank the UFPR Computer Science department.

References

1. Boiy, E., Moens, M.F.: A machine learning approach to sentiment analysis in multilingual web texts. Information retrieval **12**(5), 526–558 (2009)
2. Bosco, C., Felice, D., Poletto, F., Sanguinetti, M., Maurizio, T.: Overview of the evalita 2018 hate speech detection task. In: EVALITA 2018-Sixth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. vol. 2263, pp. 1–9. CEUR (2018)
3. Cignarella, A.T., Frenda, S., Basile, V., Bosco, C., Patti, V., Rosso, P., et al.: Overview of the evalita 2018 task on irony detection in italian tweets (ironita). In: Sixth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian (EVALITA 2018). vol. 2263, pp. 1–6. CEUR-WS (2018)
4. Correa, U.B., dos Santos, L.P., Coelho, L., de Freitas, L.A.: Overview of the IDPT Task on Irony Detection in Portuguese at IberLEF 2021. Procesamiento del Lenguaje Natural, vol. 67, (2021)
5. Gupta, R.K., Yang, Y.: Crystalnest at semeval-2017 task 4: Using sarcasm detection for enhancing sentiment classification and quantification. In: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017). pp. 626–633 (2017)
6. Liddy, E.D.: Natural language processing (2001)
7. nltk: Natural language toolkit v3.6.2 (may 2021), `http://www.nltk.org/`
8. Ortega-Bueno, R., Rangel, F., Hernández Farıas, D., 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. vol. 2421, pp. 229–256 (2019)
9. 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)

10. spaCy: spacy v3.0 (may 2021), `https://spacy.io/`