VRAIN at IroSva 2019: Exploring Classical and Transfer Learning approaches to Tweet Irony Detection

Javier Iranzo-Sánchez[0000−0002−4035−3295] and
Ramon Ruiz-Dolz[0000−0002−3059−8520]

Valencian Research Institute for Artificial Intelligence (VRAIN)
Camino de Vera s/n. 46022 Valencia (Spain)
jairsan@upv.es, raruidol@dsic.upv.es

Abstract. This paper describes VRAIN's participation at *IroSvA 2019: Irony Detection in Spanish Variants* task of the Iberian Languagues Evaluation Forum (IberLEF 2019). We describe the entire pre-processing, feature extraction, model selection and hyperparameter optimization carried out for our submissions to the shared task. A central part of our work is to provide an in-depth comparison of the performance of different classical machine learning techniques, as well as some recent transfer learning proposals for Natuaral Language Processing (NLP) classification problems.

Keywords: Natural Language Processing · Irony Detection · Transfer Learning

1 Introduction

From the linguistic point of view, irony is a very interesting property of language. As defined in [7], irony is the ability of expressing some specific meaning by the use of terms and words that, by its own, have the completely opposite meaning. On the other hand, from a computational viewpoint, irony can be seen as an important headache when performing natural language analysis tasks. For example, in sentiment analysis, some of the most important features to determine the text polarity are inferred from the words appearing in the text (e.g. negation, n-grams, POS tagging, etc.) [8]. With an ironic text, some of these features should be smoothed in order to perform correctly the sentiment analysis. Irony can be a problem when performing sentiment analysis from a text, this issue has been directly observed in past International Workshop on Semantic Evaluation (SemEval[1]) editions. This paper describes VRAIN's participation at *IroSvA 2019: Irony Detection in Spanish Variants* task of the Iberian Languages Evaluation Forum (IberLEF 2019). In this task we must identify ironic texts written in three Spanish variations (from Spain, Mexico and Cuba). For

[1] http://alt.qcri.org/semeval2019/

Spain and Mexico subtasks, we must detect ironic tweets and for Cuba subtask we are supposed to detect ironic comments from a news website.

We worked in each substask in isolation (only the Spanish tweets from Mexico were used for the Mexico subtask and so on), but used the same approach and pipeline in all 3 subtasks. Model selection and hyper-parameter optimisation were individually carried out for each subtask.

2 Feature Extraction

The text was tokenized using NLTK's [2] TweetTokenizer. Additionally, we experimented with substituting all occurrences of hashtags, url, user mentions and numbers by a generic topic for each category, but we finally decided against it since it decreased model performance.

Each tweet was represented by a vector of counts of word n-grams. Using counts directly instead of tf-idf performed better in our exploratory experiments. The dataset contains additional information apart from the tweets themselves. Specifically, we are given the corresponding topic for each of the tweets. We have tried two ways of leveraging this information. In the first approach, which we have called *global-model*, only one model is trained for each subtask, and a one-hot vector encoding the topic is append to every sample. Therefore, in this approach, we have a single model per sub-task, trained with data from all the topics.

In the second approach, which we have called *topic-model*, we train one model per topic. Thus, at training time, we train each of the individual models using only data from one topic, and at inference time, for each of the tweets, we use the predictions of the model that has been trained using the data of the tweet's topic. The results of both approaches are compared and evaluated in Section 4.

3 System Description

We will now describe the different models we tried for irony detection. In order to select appropriate values for the hyperparameters of each model, we carry out 5-fold Cross Validation, and select the configurations that obtained higher F1 (macro-averaged). Unless otherwise noted, methods are implemented using the sklearn toolkit [10].

3.1 Classification approaches

- **Naive Bayes**: The Naive Bayes approach is a well-known technique for tackling many classification problems. A Multinomial distribution is used to model $P(x_i|c)$.
- **Support Vector Machines**: Support Vector Machines [4] are Maximum Margin Classifiers that have been shown to obtain good results in a variety of tasks. We use a linear kernel, that has been shown to out-perform other non-linear kernels in text classification problems [12].

– **Gradient Tree Boosting**: Gradient Tree Boosting is a boosting technique that consists in an ensemble of tree models built in a sequential way from a set of weak learners. We have used the implementation available in the XGBoost toolkit [3].
– **Linear Models (fastText)** : fasttext [6] is a toolkit implementing a set of linear architectures for text classification. The model based on the CBOW architecture [9], has a word embedding matrix used to look up a representation of each word in the text. The embeddings are summed and averaged into a fixed-sized vector, which is then fed into a softmax classifier. Additionally, we have also trained a version using pre-trained word-embeddings, using a publicly available dataset of 200-dimensional word embeddings trained on Spanish Tweets [1].
– **BERT**: BERT [5] is a pre-training methodology for Transformer models [11]. BERT models are pre-trained on massive amounts of unsupervised text data, and can then be used in a transfer-learning approach for other downstream tasks. For this task, we have used the pre-trained *BERT-Base Multilingual Cased* model, and fine-tuned it on the IroSvA data for 10 epochs.

4 Experimental Evaluation

The results obtained by the different models are shown in Table 1.

Table 1. Model performance measured with 5-fold CV (macro F1)

	Subtask		
Model	*es*	*mx*	*cu*
Naive Bayes (Topic model)	0.67	0.51	0.50
Naive Bayes	0.63	0.52	0.57
fastText	0.63	0.62	0.61
fastText(Tweeter pre-trained)	0.63	0.62	0.61
BERT	0.61	0.50	0.57
SVM (Topic model)	0.70	0.60	0.58
SVM	0.70	0.60	0.66
Gradient Tree Boosting (Topic)	0.52	0.50	0.45
Gradient Tree Boosting	0.69	0.60	0.66
Ensemble (SMV + Gradient Boosting)	0.71	0.65	0.66

We can see a number of interesting results from the Table. First, except for a single case (Naive Bayes for the *es* variety), *topic models* obtain similar or worse results than their *global* counterparts. Most likely due to the reduced number of training data, the fine-grained approach of individually modelling each topic seems counterproductive.

In term of the transfer learning approaches we tried, we have not been able to leverage the knowledge obtained from the pre-trained tasks. The fastText model using pre-trained embeddings does not improve the results of the base fassText model, and the BERT model obtains results similar to the Naive Bayes model.

Overall, the best results are obtained by the SVM and Gradient Boosting models. In order to further improve the results, we have constructed an Ensemble of the SMV and Gradient Boosting models, whose predictions are the average of the individual models predictions. This obtains additional improvements in the *es* and *mx* variants, and was the model submitted to the competition. Table 2 shows the performance of our model compared to the competition baselines.

Table 2. System comparison between our submission and the competition baselines (macro F1)

	Subtask			
Model	*es*	*mx*	*cu*	Average
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
VRAIN	0.6842	0.6476	0.5204	0.6174

The results obtained by our model present significant variations depending on the task. In the case of the *es* task, our model outperforms all baselines, and in the *mx*, our system comes in second place behind the LDSE baseline. However, in the case of the *cu* task, our model is only able to beat the majority baseline. We do not know the reasons for the significant performance drop in the *cu* task between our internal experiments and the competition results.

5 Conclusions

This paper has described VRAIN's submission to IroSvA 2019. The different experiments have shown that, under the current conditions, classical models have an edge over some of the recent transfer-learning techniques that we tested. We believe that the limiting factor is the lack of sufficient training data for the finetuning step.

Our submission, based on an Ensemble of SVM and Gradient Tree Bosting models, obtains competitive results in all of the subtasks. This has been achieved using non-task-specific bag of n-gram features. It is expected that these results could be further improved with specific features for irony detection.

Acknowledgements The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 761758 (X5gon) and from the Valencian Government project PROMETEO/2018/002.

References

1. Word embeddings trained with word2vec on 200 million spanish tweets using 200 dimensions, http://new.spinningbytes.com/resources/wordembeddings/
2. Bird, S.: NLTK: the natural language toolkit. In: ACL 2006, 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, Sydney, Australia, 17-21 July 2006 (2006), http://aclweb.org/anthology/P06-4018
3. Chen, T., Guestrin, C.: Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016. pp. 785–794 (2016). https://doi.org/10.1145/2939672.2939785, https://doi.org/10.1145/2939672.2939785
4. Cortes, C., Vapnik, V.: Support-vector networks. Machine Learning **20**(3), 273–297 (1995). https://doi.org/10.1007/BF00994018, https://doi.org/10.1007/BF00994018
5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
6. Grave, E., Mikolov, T., Joulin, A., Bojanowski, P.: Bag of tricks for efficient text classification. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 2: Short Papers. pp. 427–431 (2017), https://aclanthology.info/papers/E17-2068/e17-2068
7. Grice, H.P., et al.: Logic and conversation
8. Liu, B.: Sentiment analysis and opinion mining. Synthesis lectures on human language technologies **5**(1), 1–167 (2012)
9. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: 1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings (2013), http://arxiv.org/abs/1301.3781
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), http://dl.acm.org/citation.cfm?id=2078195
11. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA. pp. 6000–6010 (2017), http://papers.nips.cc/paper/7181-attention-is-all-you-need
12. Yang, Y., Liu, X.: A re-examination of text categorization methods. In: SIGIR '99: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 15-19, 1999, Berkeley, CA, USA. pp. 42–49 (1999). https://doi.org/10.1145/312624.312647, https://doi.org/10.1145/312624.312647