TASS 2018: Workshop on Semantic Analysis at SEPLN, septiembre 2018, págs. 29-35

Atalaya at TASS 2018: Sentiment Analysis with Tweet Embeddings and Data Augmentation

Atalaya en TASS 2018: Análisis de Sentimiento con Embeddings de Tweets y Aumentación de Datos

Franco M. Luque[1], **Juan Manuel Pérez**[2]
[1]Universidad Nacional de Córdoba & CONICET
[2]Universidad de Buenos Aires & CONICET
francolq@famaf.unc.edu.ar, jmperez@dc.uba.ar

Resumen: El workshop TASS 2018 propone diferentes desafíos de análisis semántico del Español. Este trabajo presenta nuestra participación con el equipo Atalaya en la tarea de clasificación de polaridad de tweets. Seguimos técnicas estándar de preprocesamiento, representación y clasificación, y también exploramos algunas ideas novedosas. En particular, para obtener embeddings de tweets entrenamos word embeddings con información de subpalabras, y usamos un esquema de pesaje para promediarlos. Para lidiar con problemas de sobreajuste causados por la escasez de datos de entrenamiento, probamos una estrategia de aumentación de datos basada en traducción automática bidireccional. Experimentos con clasificadores lineales y modelos neuronales muestran resultados competitivos para las diferentes subtareas propuestas en el desafío.
Palabras clave: Análisis de Sentimiento, Clasificación de Polaridad, Embeddings, Aumentación de Datos, Modelos Lineales, Redes Neuronales

Abstract: TASS 2018 workshop proposes different challenges on semantic analysis in Spanish. This work presents our participation as team Atalaya in the task of polarity classification of tweets. We followed standard techniques in preprocessing, representation and classification, and also explored some novel ideas. In particular, to obtain tweet embeddings we trained subword-aware word embeddings and use a weighted scheme to average them. To deal with overfitting problems caused by training data scarcity, we tried a data augmentation strategy based on two-way machine translation. Experiments with linear classifiers and neural models show competitive results for the different subtasks proposed in the challenge.
Keywords: Sentiment Analysis, Polarity Classification, Embeddings, Data Augmentation, Linear Models, Neural Networks

1 Introduction

The TASS workshop presents every year different challenges related to sentiment analysis in Spanish. One of the main tasks is polarity classification of tweets and tweet aspects. In particular, task 1 of TASS 2018 (Martínez-Cámara et al., 2018) proposes polarity classification on tweet datasets from three different Spanish speaking countries: Spain (ES), Costa Rica (CR) and Perú (PE). This article describes our participation in TASS 2018 task 1 with team Atalaya. We present polarity classification systems using standard techniques and propose improvements based on an iterative experimental development process. We tried different approaches for tweet preprocessing, vector representation and polarity classification models. Standard preprocessing techniques, including text simplification, stopword filtering, lemmatization and negation handling were used. Tweets were represented with bag-of-words, bag-of-characters, tweet embeddings and combinations of these. As classification models, we considered linear classifiers and neural networks.

We used *fastText* subword-aware word vectors using tweet datasets specifically pre-

ISSN 1613-0073

pared for the task. Tweet vectors were computed from word vectors using a weighted averaging scheme, with weights inversely proportional to word frequency.

To cope with scarcity of training data, we experimented with a data augmentation trick based on translation of training data to other languages and back to Spanish.

Embedding weighting and data augmentation represent novel approaches in the context of TASS. In experiments, both ideas showed improvements in prediction quality for some configurations.

The rest of the paper is as follows: Next section describes the main techniques and resources we tried; section 3 presents the experimental development of the systems, describing explored configurations and final models selection; and section 4 summarizes our final results for the competition, and addresses conclusions and future work.

2 Techniques and Resources

This section describes the main techniques and resources we used to define the basic components to build our systems.

2.1 Preprocessing

Preprocessing is crucial in NLP applications, specially when working with noisy user-generated data.

We divided preprocessing into a two-stage process: First, we defined basic tweet preprocessing, using well-known standard and general purpose techniques; then, we defined sentiment-oriented preprocessing, using techniques that try to emphasize semantic information.

Basic tweet preprocessing includes:

- Tokenization using NLTK tweet tokenizer (Bird and Loper, 2004).
- Replacement of handles with token `'@USER'`, URLs with `'URL'`, and e-mails with `'user@mail.com'`.
- Replacement of four or more repeated letters with three letters.

Sentiment-oriented preprocessing includes the following additional steps:

- Lowercasing.
- Removal of stopwords, using NLTK Spanish stopword list.
- Removal of numbers.
- Lemmatization using TreeTagger (Schmid, 1995).
- Simple negation handling: We find negation words and add the prefix `'NOT_'` to the following tokens. Up to three tokens are negated, or less if a non-word token is found. (Das et al., 2001; Pang, Lee, and Vaithyanathan, 2002)
- Removal of punctuation.
- Removal of consecutive repetitions of handles and URLs.

No treatment was performed to hashtags, emojis, interjections and onomatopeias. Moreover, no spelling correction nor any other additional normalization was applied.

2.2 Bags of Words and Characters

The simplest approach we considered to build tweet representations was the bag-of-words encoding. A bag-of-words (BOW) builds feature vectors for each token seen in training data. For a particular tweet, its BOW vector contains the number of ocurrences of each token in the tweet. Resulting vectors are high-dimensional and sparse. Variations of BOWs include counting not only single tokens but also n-grams of tokens, binarizing counts, and limiting the number of features.

Character usage in tweets may also hold useful information for sentiment analysis. Character n-grams —such as presence and repetition of uppercase letters, emoticons and exclamation marks— may indicate strong presence of sentiment of some kind, where others may indicate a more formal writing style, and therefore an absence of sentiment.

To capture this information, we considered a bag-of-characters (BOC) representation that encodes counts of character n-grams for some values of n. These vectors are computed from original texts of tweets, with no preprocessing at all. BOCs have the same variants and parameters as BOWs.

2.3 Word Embeddings

Word embeddings are low-dimensional dense vector representations of words (Mikolov et al., 2013). These representations encode syntactical and semantical relations of words, useful for NLP tasks, and they can be learned in an unsupervised fashion using large quantities of plain text, providing high vocabulary coverage. When precomputed embeddings

are used as features in supervised tasks, they provide robust information for words that are rare or unseen in training data. This is particularly useful when training data is scarce, as in this competition.

Recent work on embeddings introduced the usage of subword information to compute word vectors. Informative representations for out-of-vocabulary (OOV) words can be obtained from subword embeddings. OOV words are an important issue when working with highly noisy data such as user generated data in social networks. Here, the need for text normalization in preprocessing can be alleviated with subword-based embeddings.

In our work, we used fastText subword-based embeddings library (Bojanowski et al., 2016). Instead of using pretrained vectors, we decided to train our own embeddings on Twitter data.

To address the multilingual character of the challenge, we first collected a database of ∼90 million tweets from various Spanish-speaking countries, including the ones concerning the challenge. Then, we prepared two versions of the data, one using only basic preprocessing, and the other one using sentiment oriented preprocessing (only excepting lemmatization). For these two datasets, we trained skipgram embeddings using different parameter configurations, including the number of dimensions, size of word and subword n-grams and size of context window.

2.4 Tweet Embeddings

There are a number of ways of using word embeddings for sentiment analysis on tweets: approaches go from simple averaging of vectors for each word in the tweet, to the use of more complex architectures such as CNNs or RNNs. In this work, we used averaging to compute a single tweet embedding of same dimensionality as the original word embeddings. We followed two simple approaches: plain averaging and weighted averaging. For weighted averaging, we used a scheme that resembles Smooth Inverse Frequency (SIF) Arora, Liang, and Ma (2017), inspired by TF-IDF reweighting. Each word w is weighted with $\frac{a}{a+p(w)}$, where $p(w)$ is the word unigram probability, and a is a smoothing hyper-parameter. Big values of a means more smoothing towards plain averaging.

We also considered two options that affect tweet embeddings: binarization, which ignores token repetitions in tweets; and normalization, which scales resulting tweet vectors to have unit norm.

2.5 Data Augmentation

As the amount of training instances was small, we paid special attention to model regularization. A technique used to address this is data augmentation, which consists of creating new synthetic instances out of real ones by applying label-preserving transformations. This overfitting-reduction strategy is widely used in Computer Vision (Krizhevsky, Sutskever, and Hinton, 2012; Simard, Steinkraus, and Platt, 2003) and Speech Recognition (Jaitly and Hinton, 2013; Ko et al., 2015). For instance, images can be zoomed, cropped, rotated, etc., while keeping the objects in it still recognizable.

Data augmentation in NLP is a more subtle problem: there are no straightforward invariant-transformations such as in Computer Vision. A common technique (Zhang, Zhao, and LeCun, 2015) is to replace words with synonyms using a thesaurus.

In this work we adopted a novel technique successfully used in a recent Kaggle NLP competition.[1] The technique consists of translating the texts to a different language, and then translating them back to the original one. This process results in tweets that vary lexically and syntactically, while mostly keeping its meaning.

The tool selected to do this work was Google Translate, and the languages used as intermediates were English, French, Portuguese and Arabic. We discarded other options (e.g. Mandarin Chinese) as they greatly altered the meaning of tweets. Table 1 displays examples of tweets and the resulting artificial instances.

3 Systems Development

This section describes the polarity classification systems we developed using the tools introduced in the previous section.

We worked on two type of classifiers: linear classifiers and neural networks. In both cases, we tried to do some kind of model selection, at times using development as the optimization target, and at other times using cross-validation on the combination of train and development.

[1] https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/discussion/52557

Original Tweet	Data-augmented tweets
Gracias por la información. Parece que el olor ha cesado. Ayer pasó lo mismo sobre la misma hora	- Gracias por la información. Parece que el olor se ha detenido. Ayer sucedió lo mismo al mismo tiempo - Gracias por la información. Parece que el olor se ha detenido. Ayer, lo mismo ocurrió al mismo tiempo
Muy buenas amigos! Como podemos contactar con ustedes	- ¡Muy buenos amigos! ¿Cómo podemos ponernos en contacto con usted? - Muy buenos amigos! ¿Cómo podemos contactarlo?
La verdad es que tiene buena pinta. Investigaré, gracias	- La verdad es que parece bueno. Voy a investigar, gracias - La verdad es que se ve bien. Voy a investigar, gracias - El hecho es que se ven bien. Lo comprobaré, gracias

Table 1: Data augmentation examples. Left column shows original tweets, right column shows results of two-way translations for several intermediate languages.

Next subsections describe the experimental development and the best configurations we found for both types of system

3.1 Linear Classifiers

We first built a classifying pipeline using simple linear classifying models —such as logistic regressions and SVMs— that were implemented with scikit-learn (Pedregosa et al., 2011). Next, we describe the model selection process, done almost entirely using the InterTASS ES corpus.

As input features, we combined the three representations described in the previous section: bag-of-words, bag-of-characters and tweet embeddings.

For the bag of words and characters, early experiments showed a clear advantage of binary values over counts, together with TF-IDF re-weighting. First choices for n-gram ranges were (1, 2) for words and (1, 3) for characters.

For the embeddings, sentiment-oriented word vectors showed an advantage over basic vectors. We tried embeddings of dimensions 50, 100, 200 and 300. Best results were found with 50 dimensions, and there were no statistically significant differences.

To compute tweet embeddings, we tried basic averaging (as provided by *fastText*) and the weighted averaging scheme described in section 2.4. We experimented with smoothing values $a = 10^n$ for $n \in \{-3, \ldots, 3\}$ resulting in a significant advantage of using $a = 0.1$. Here, binarization and normalization as described in section 2.4 showed better results.

For the classifier, we tried logistic regressions (LRs) and linear-kernel SVMs. To alleviate the class imbalance problem, training items were weighted according to the inverse of the class frequency. Both LR and linear SVM hyper-parameters were selected targeting the optimization of accuracy and Macro-F1 over InterTASS ES development set. In particular, the best regularization parameters found were $C = 1.0$ for LRs, and $C = 0.05$ for SVMs. Logistic regressions were selected over SVMs as they performed consistently better in all experiments.

Model	BOW	BOC	M-F1	Acc.
LR	(1, 2)	(1, 3)	0.496	0.634
LR+DA	(1, 2)	(1, 3)	0.490	0.615
LR	(1, 5)	(1, 6)	0.493	0.634
LR+DA	(1, 5)	(1, 6)	**0.529**	**0.648**

Table 2: Experiments with logistic regressions (LR), showing the interaction of training data augmentation (DA) with n-gram size ranges for bags of words and characters (BOW and BOC, resp.). Results are on InterTASS ES development set.

When adding augmented data, first results showed a significant degradation in accuracy. However, an exploration of parameter values showed that it allowed an improvement in performance when increasing the range of n-gram sizes considered for BOWs and BOCs. Best results were found with up to 5-grams for words, and up to 6-grams for characters. Tab. 2 shows how data augmentation combined with bigger n-gram ranges improved results.

Most previous parameter selection was reviewed after data augmentation, confirming selected values. We also tried adding training data from General TASS corpus, to find that this was harmful for our models. With the optimal models found in this process we submitted final results for the Spanish (ES) monolingual task.

For Costa Rica (CR) and Perú (PE) monolingual tasks, same values than for ES were used for most parameters. Only weighted averaging, data augmentation and n-gram ranges were explored. In CR data, weighting improved results, with the peak at

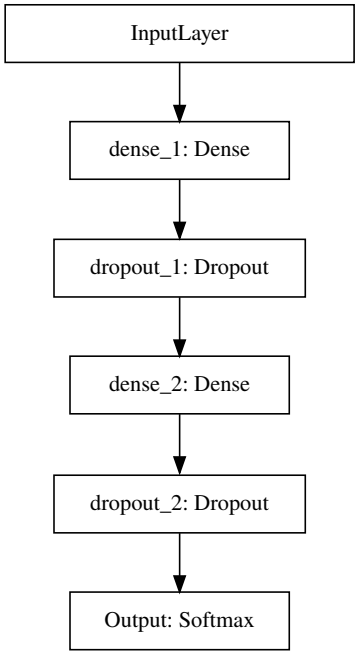

Figure 1: Architecture of the MLP.

$a = 0.5$. Data augmentation also was good, with the best results using up to 4-grams for words and 6-grams for characters. In PE data, neither weighting nor data augmentation were helpful. Best results were found using up to 2-gram for words and 5-grams for characters.

3.2 Multilayer Perceptron

In the second set of experiments we used multilayer perceptrons (MLP) neural networks. MLPs performed well in previous editions of the challenge (Díaz-Galiano et al., 2018).

Fig. 1 displays the chosen architecture, consisting of two hidden layers and a softmax output. ReLU units were used as activation functions in the hidden layers. To avoid overfitting, we tried dropout (Srivastava et al., 2014) and early stopping.

To find the best configurations, we performed random search (Bergstra and Bengio, 2012) using 5-fold cross-validation over the InterTASS ES training and development datasets. The explored configurations and hyperparameters were:

- BOW features: No BOW features at all, top-50 or top-150.
- Tweet embeddings: Basic or weighted averaging.
- Hidden layers: Different number of neurons and keep-probabilities[2].

Task	Model	M-F1	Acc.
Mono ES	MLP	0.476	0.544
	LR	0.468	0.599
Mono CR	MLP	0.451	0.562
	LR	0.475	0.582
Mono PE	MLP	0.437	0.520
	LR	0.462	0.451
Cross Lingual ES		0.441	0.485
Cross Lingual PE	MLP	0.438	0.523
Cross Lingual CR		0.453	0.565

Table 3: Submitted results for each subtask.

Results of this search showed that bag-of-words features and embedding weighting did not improve performance. Regarding the MLP architecture, we selected 256 as the size of the first layer and 128 for the second, and keep-probabilities of 0.25 and 0.55 respectively. This configuration was used in all subtasks.

Data augmentation in combination with MLPs showed mixed results. For the monolingual ES subtask, using synthetic data resulted in a Macro-F1 gain while for monolingual PE it degraded the results.

All monolingual models were trained using the respective train sections of InterTASS datasets. General TASS was not used as it not showed improvements. For cross-lingual tasks, models for each language were trained using the datasets for the two other languages.

We used Keras (Chollet and others, 2015) to implement the model and scikit-learn (Pedregosa et al., 2011) to perform the cross-validation.

4 Conclusions and Future Work

We presented our participation on TASS 2018 task 1 as team Atalaya. We explored standard approaches as well as some simple but original recent ideas such as data augmentation and word embedding weighting. Table 3 displays results for each subtask. Our systems ranked among the first three in all the subtasks.

Experiments show that competitive results can be achieved without having to resort to complex neural architectures such as CNNs, RNNs, LSTMs, etc. Even simple lo-

[2]Probability of keeping the value of a neuron when training with dropout.

gistic regressions were able to rank among the top performing systems.

Future work includes further exploration on data augmentation, tweet embedding techniques, and sentiment-oriented word embeddings. We also aim at improving preprocessing and adopting modern neural classifying models.

References

Arora, S., Y. Liang, and T. Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings.

Bergstra, J. and Y. Bengio. 2012. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb):281–305.

Bird, S. and E. Loper. 2004. Nltk: the natural language toolkit. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, page 31. Association for Computational Linguistics.

Bojanowski, P., E. Grave, A. Joulin, and T. Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Chollet, F. et al. 2015. Keras.

Das, S. R., M. Y. Chen, T. V. Agarwal, C. Brooks, Y. shee Chan, D. Gibson, D. Leinweber, A. Martinez-jerez, P. Raghubir, S. Rajagopalan, A. Ranade, M. Rubinstein, and P. Tufano. 2001. Yahoo! for amazon: Sentiment extraction from small talk on the web. In *8th Asia Pacific Finance Association Annual Conference*.

Díaz-Galiano, M. C., E. Martínez-Cámara, M. Ángel García Cumbreras, M. G. Vega, and J. V. Román. 2018. The democratization of deep learning in tass 2017. *Procesamiento del Lenguaje Natural*, 60(0):37–44.

Jaitly, N. and G. E. Hinton. 2013. Vocal tract length perturbation (vtlp) improves speech recognition. In *Proc. ICML Workshop on Deep Learning for Audio, Speech and Language*, volume 117.

Ko, T., V. Peddinti, D. Povey, and S. Khudanpur. 2015. Audio augmentation for speech recognition. In *Sixteenth Annual Conference of the International Speech Communication Association*.

Krizhevsky, A., I. Sutskever, and G. E. Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Martínez-Cámara, E., Y. Almeida Cruz, M. C. Díaz-Galiano, S. Estévez Velarde, M. A. García-Cumbreras, M. García-Vega, Y. Gutiérrez Vázquez, A. Montejo Ráez, A. Montoyo Guijarro, R. Muñoz Guillena, A. Piad Morffis, and J. Villena-Román. 2018. Overview of TASS 2018: Opinions, health and emotions. In E. Martínez-Cámara, Y. Almeida Cruz, M. C. Díaz-Galiano, S. Estévez Velarde, M. A. García-Cumbreras, M. García-Vega, Y. Gutiérrez Vázquez, A. Montejo Ráez, A. Montoyo Guijarro, R. Muñoz Guillena, A. Piad Morffis, and J. Villena-Román, editors, *Proceedings of TASS 2018: Workshop on Semantic Analysis at SEPLN (TASS 2018)*, volume 2172 of *CEUR Workshop Proceedings*, Sevilla, Spain, September. CEUR-WS.

Mikolov, T., I. Sutskever, K. Chen, G. Corrado, and J. Dean. 2013. Distributed representations of words and phrases and their compositionality. *CoRR*, abs/1310.4546.

Pang, B., L. Lee, and S. Vaithyanathan. 2002. Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, pages 79–86. Association for Computational Linguistics, July.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Schmid, H. 1995. Improvements in part-of-speech tagging with an application to german. In *In Proceedings of the ACL SIGDAT-Workshop*, pages 47–50.

Simard, P. Y., D. Steinkraus, and J. C. Platt. 2003. Best practices for convolutional

neural networks applied to visual document analysis. In *null*, page 958. IEEE.

Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Zhang, X., J. Zhao, and Y. LeCun. 2015. Character-level convolutional networks for text classification. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*. Curran Associates, Inc., pages 649–657.