UACh at MEX-A3T 2020: Detecting Aggressive Tweets by Incorporating Author and Message Context

Marco Casavantes[a], Roberto López[a] and Luis Carlos González[a]

[a]*Universidad Autónoma de Chihuahua. Facultad de Ingeniería. Chihuahua, Chih., Mexico*

Abstract

In this paper we describe our participation in the Aggressiveness Detection Track at the third edition of MEX-A3T. We evaluate two strategies for text classification, a traditional classifier (Logistic Regression) and a classifier based on transformers (BETO). We also study the inclusion of social media metadata features to try to get context from authors and text messages.

Keywords

Spanish text classification, Aggressiveness Detection, Metadata, Twitter

1. Introduction

Social media platforms are one of the most popular ways to communicate in the "Information Age", allowing their users to express and spread many kinds of ideas, from the cheerful to the not so positive side of "freedom of speech". These networks aren't immune to people that share offensive content, users that show malicious intent and are quick to reply with aggressive manners. Anonymity, ease of access and lack of punishment for the most part, encourages these individuals to express themselves offensively. The volume of messages that are sent daily on social media makes moderation a difficult task to be dealt with by conventional means, and as people increasingly communicate online, the need for high quality automated abusive language classifiers becomes much more profound[1]. One of the goals of the third edition of MEX-A3T [2] is to tackle this problem and further improve the research of this important Natural Language Processing (NLP) task, the detection of aggressive tweets in Mexican Spanish. The issue is that spotting offensive messages and hate speech is challenging because systems cannot rely on the text content [3, 4]; for this reason, in this work we evaluate a strategy to try to give context to short texts from social media by taking into account message and author metadata. Our hypothesis is that these additional attributes are expected to better distinguish between offensive and not-offensive messages and improve classification scores.

Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2020)

EMAIL: p271673@uach.mx (M. Casavantes); jrlopez@uach.mx (R. López); lcgonzalez@uach.mx (L.C. González)

ORCID: 0000-0001-6186-0192 (R. López); 0000-0003-1546-9752 (L.C. González)

Table 1
Metadata of Tweet object.

Attribute	Type
Retweet count	Integer
Favorite count	Integer
Date of creation	Date
Reply status	Boolean
Quote status	Boolean

Table 2
Metadata of User object.

Attribute	Type
Username	String
Verified	Boolean
Followers count	Integer
Friends count	Integer
Listed count	Integer
Favorites count	Integer
Statuses count	Integer
Default profile	Boolean
Default profile image	Boolean
Created_at	Date

2. Proposed Method

2.1. Data Pre-processing

1. We replaced the string of characters "&" with "&". This was necessary to get a closer representation to the text used in the original tweets.
2. We strip emojis from the tweets.
3. All words were made lowercase.

For LogisticRegression classifier, we tokenized the dataset using the TweetTokenizer utility from NLTK [5], for BETO's case we employed BertTokenizer [6].

2.2. Features

We conducted our research using the following features:
Lexical: We use word n-grams (n=1, 3) as features, this collection of terms is weighted with its term frequency (TF).
Metadata (MD): By using the Standard Twitter API platform [7] together with additional libraries in Python such as GetOldTweets3 [8] and Twython [9] it is possible to search for every tweet in the dataset; if a message is still available online, we are able to retrieve properties of the post as well as information of the author of the tweet (shown in tables 1 and 2).

2.3. Classifiers

Logistic Regression (LR): Considered part of the traditional approaches for most of the NLP tasks. This algorithm uses a linear regression equation that includes a function called "logistic/sigmoid function", this function produces an "S" shaped curve that is able to tell the probability of class assignment.
BETO: BERT model trained on a big Spanish corpus. BETO [10] is of size similar to a BERT-Base and was trained with the Whole Word Masking technique. BERT models [11] are a new method of pre-training language representations, currently obtaining state-of-the-art results on a wide array of NLP tasks.

Table 3
Data distribution for Spanish tweets corpus.

Class	Train set	Test set
Aggressive (1)	2110	N/A
Non-aggressive (0)	5222	N/A
Total	7332	3143

Table 4
Recovered tweets for MEX-A3T 2020 dataset.

Original Tweets	Recovered Tweets	Percentage Recovered
10,475	7,320	69.88%

Because of its computational affordability and flexibility at handling different types of inputs (including NULL values), XGBoost Classifier was selected as the blender of the information present in our proposed systems. The first step in our framework involves feeding the text part of the dataset to either LR or BETO, then the classifier returns class probabilities, and lastly these predictions are concatenated with either metadata or NaN values (for the tweets that we couldn't retrieve their metadata online) to form the input vector for XGBoost, which outputs the final decisions.

3. Experiments and Results

The datasets were provided by MEX-A3T Team. Table 3 shows the distribution of training and test partitions for Spanish tweets.

Our first task was to "extend" the dataset. This process is simple: for every message in the collection a query is made using the text to search for that tweet online. Due to the nature of the task at hand, some tweets couldn't be recovered, possibly due to deletion of posts or suspended accounts. Table 4 shows the amount of tweets that we were able to recover.

We trained two classification systems for this task, one with Logistic Regression and one using BETO, and we decided to submit a set of predictions for each system:

- **Run 1** consists of a XGBoost classifier fed with metadata features and Logistic Regression probabilities, trained with features from a Bag of Words of range=(1, 3) considering the term frequency of all the tokens that appear at least twice.
- **Run 2** is similar to Run 1, but in this case the XGBoost classifier takes metadata features and BETO probabilities as inputs.

To evaluate our experiments with the features discussed in section 2.2, we performed a 5-Fold Cross Validation on the train set for LR, and a single train-test split using BETO due to time constraints.

We performed all modeling regarding the creation of TF feature matrices, LR and XGBoost classifiers using scikit-learn[12], and for the BETO model, we used the implementation described

Table 5
Detailed classification with F1-scores in the validation stage.

Run	Added features	Classifier	F1 offensive	F1 macro
	None	LR	0.6931	0.7937
	None	LR + XGB	0.7147	0.8025
Run 1	User favorites count User statuses count Default profile Reply status	LR + XGB	**0.7195**	**0.8062**
	None	BETO	0.7566	0.8335
	None	BETO + XGB	0.7566	0.8312
Run 2	Tweet favorite count User listed count Default profile	BETO + XGB	**0.7618**	**0.8352**

Table 6
Top 5 scores of the Aggressiveness Identification Track.

Rank	Team Name	F1 offensive	F1 non-offensive	F1 macro
1	CIMAT-1	0.7998	0.9195	0.8596
2	CIMAT-2	0.7971	0.9205	0.8588
3	UPB-2	0.7969	0.9107	0.8538
4	**UACh-2**	**0.7720**	**0.9042**	**0.8381**
5	INGEOTEC	0.7468	0.8933	0.8200

in [10]. We decided to include or exclude metadata features through manual feature selection. For each run, a different combination of added features exhibited improved F1 scores, the attributes used along with F1 scores are shown in Table 5.

3.1. Results

From a total of 19 registered submissions, the BETO classification was ranked 4th place (above both Bi-GRU and BoW-SVM baselines), while the LR classification was ranked 8th. Table 6 lists the top five final rankings for the Aggressiveness Identification track for 2020 (our scores appear in bold).

3.2. Analysis

To breakdown our results, we started by addressing the performance of our proposal, regarding F1-score and contrasted against the rest of the competitors. Fig. 1 presents two box plots for the complete distribution of submitted results in terms of F1 offensive and F1 non-offensive. This analysis suggests that the outcome achieved by our proposal is competitive, been located within the first quartile for all submissions. The second part of our analysis focuses on reviewing what kind of class predictions were changed by adding metadata to our best ranked system. In the validation stage, metadata features were able to rectify 25 label assignments (including 17 true positives) and also make 27 new mistakes (with 24 false positives). Table 7 shows some

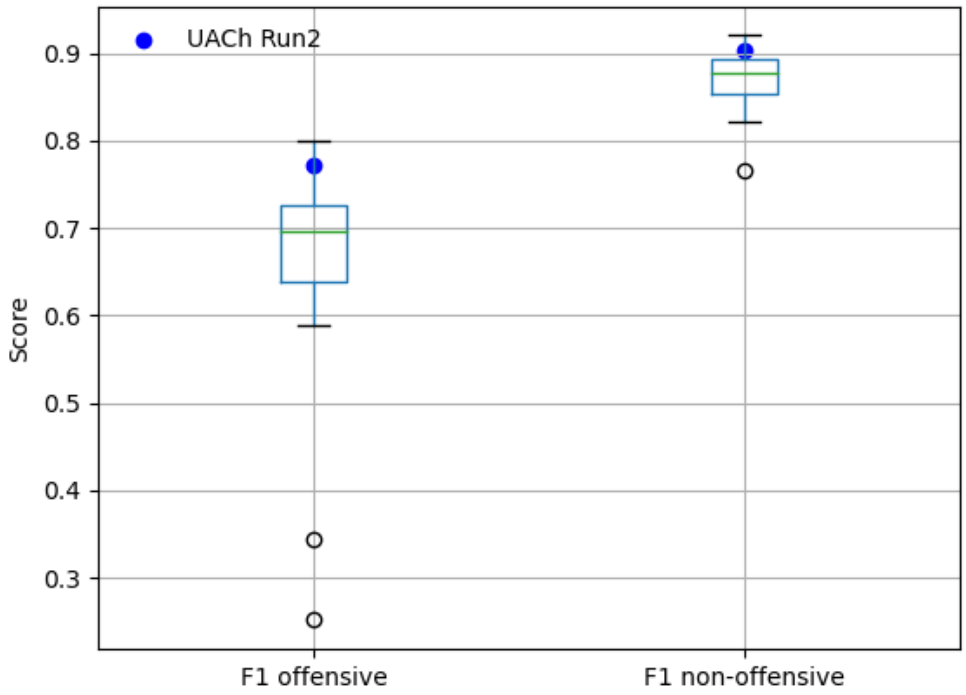

Figure 1: Box plots of the results for Aggressive Identification task.

examples of these results (trigger words appear in bold).

4. Conclusions and Future Work

In this paper, we describe our strategy to classify aggressive and non-aggressive tweets in Mexican Spanish. In our best performing system, we use a transformers based classifier, BETO, paired with the addition of metadata features through a decision-tree-based ensemble algorithm, XGBoost. Our proposal shows to be competitive for this specific task. We noticed that metadata can be helpful to detect subtle samples of aggressiveness but also classify tweets as offensive when swearing is present and not necessarily used to insult someone. However it is interesting to see how these additional attributes based on the behavior of users inside social media can be used to strengthen classification methods. We look forward to enhance our current framework, focusing on different levels of lexical and contextual analysis using state-of-the-art approaches and complementing them with metadata attributes.

Table 7
Examples of fixed and new mistakes by adding metadata features.

Text	Class	w/o MD	w/MD
“Me caes más **gorda** de lo que estás , imagínate."	1	0	1
“Una frentona, una narizona, una **gorda** y una dientona, UF SUPER COMBO."	1	0	1
“Una **gorda** es como un pozole, si esta bueno te chingas otro."	0	1	0
“Esa mujer de intelectual no tiene **ni madres!** Sólo copia lo que escribe!!!	0	0	1
“Me sorprende que la gente diga que soy **mamón** cuando soy a toda **pinche madre putos**."	0	0	1
“Las únicas **gordas prietas** que no me dan **asco** son las caguamas."	1	1	0

References

[1] C. Nobata, J. Tetreault, A. Thomas, Y. Mehdad, Y. Chang, Abusive language detection in online user content, in: Proceedings of the 25th International Conference on World Wide Web, WWW ’16, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 2016, pp. 145–153. URL: https://doi.org/10.1145/2872427.2883062. doi:10.1145/2872427.2883062.

[2] M. E. Aragón, H. Jarquín, M. Montes-y Gómez, H. J. Escalante, L. Villaseñor-Pineda, H. Gómez-Adorno, G. Bel-Enguix, J.-P. Posadas-Durán, Overview of MEX-A3T at IberLEF 2020: Fake news and Aggressiveness Analysis in Mexican Spanish, in: Notebook Papers of 2nd SEPLN Workshop on Iberian Languages Evaluation Forum (IberLEF), Malaga, Spain, September, 2020.

[3] S. Modha, T. Mandl, P. Majumder, D. Patel, Overview of the HASOC track at FIRE 2019: Hate speech and offensive content identification in indo-european languages, CEUR Workshop Proceedings 2517 (2019) 167–190. doi:10.1145/3368567.3368584.

[4] M. Casavantes, R. López, L. González, M. Montes-y Gómez, UACH-INAOE at HASOC 2019: detecting aggressive tweets by incorporating authors’ traits as descriptors, in: Proceedings of the 11th annual meeting of the Forum for Information Retrieval Evaluation, 2019.

[5] nltk.tokenize package — NLTK 3.5 documentation, 2020. URL: https://www.nltk.org/api/nltk.tokenize.html.

[6] BERT — transformers 2.11.0 documentation, 2020. URL: https://huggingface.co/transformers/model_doc/bert.html.

[7] Tweets – Twitter Developers, 2020. URL: https://developer.twitter.com/en/products/tweets.

[8] GetOldTweets3 · PyPI, 2018. URL: https://pypi.org/project/GetOldTweets3/.

[9] R. McGrath, Twython — Twython 3.6.0 documentation, 2013. URL: https://twython.readthedocs.io/en/latest/.

[10] J. Cañete, G. Chaperon, R. Fuentes, J. Pérez, Spanish pre-trained bert model and evaluation data, in: to appear in PML4DC at ICLR 2020, 2020.

[11] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional

transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).
[12] F. Pedregosa, 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, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.