ITCG's Participation at MEX-A3T 2020: Aggressive Identification and Fake News Detection Based on Textual Features for Mexican Spanish

Diego Zaizar-Gutiérrez[a], Daniel Fajardo-Delgado[a] and Miguel Á. Álvarez-Carmona[b,c]

[a] *Tecnológico Nacional de México / Campus Ciudad Guzmán, Mexico*
[b] *Centro de Investigación Científica y de Educación Superior de Ensenada (CICESE), Mexico*
[c] *Consejo Nacional de Ciencia y Tecnología (CONACYT), Mexico*

Abstract

This paper explains our approach to Aggressiveness Identification and Fake News Classification in the 2020 MEX-A3T shared task. The tasks propose a binary classification for both tasks (aggressive and non-aggressive or fake news and non-fake news). We approached the problem using simple basic methods of features selection and terms weighing. We trained with a set of machine learning algorithms. Our best run for aggressiveness identification achieved an accuracy of 0.81, where the best result obtained 0.88. On the other hand, for the aggressiveness identification, our accuracy result was 0.78, where the best result was 0.85.

Keywords

Aggressiveness Identification, Fake News Classification, Natural Language Processing

1. Introduction

Nowadays, technology has a significant role in which people communicate with each other, giving rise to new services such as social networks. Social networks present several challenges to maintain communication channels open to the free sharing of ideas. For example, it is effortless for some people sharing aggressive speeches affecting the experience of other consumers or people interested in being part of the communities and their conversations.

The number of messages sent daily makes the moderation of communication channels challenging to deal with by conventional means, and as people increasingly communicate online, the need for automated abusive language classifiers becomes much more profound [1].

Another inconvenience of the free ideas traffic on social networks is sharing news without an entity that regulates the veracity of the information. In [2], *fake news* is defined as fabricated information that mimics news media content in form but not in organizational process or intent.

Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2020)

EMAIL: diego15460155@itcg.edu.mx (D. Zaizar-Gutiérrez); dfajardo@itcg.edu.mx (D. Fajardo-Delgado); malvarezc@cicese.mx (M.Á. Álvarez-Carmona)
ORCID: 0000-0002-4486-2003 (D. Zaizar-Gutiérrez); 0000-0001-8215-5927 (D. Fajardo-Delgado); 0000-0003-4421-5575 (M.Á. Álvarez-Carmona)

CEUR Workshop Proceedings (CEUR-WS.org)

The problem is that fake news overlaps with other information environments, such as misinformation (false or misleading information), disinformation (false information that is purposely spread to deceive people), and real news [2]. This causes that discerning between real and false news can be a difficult task. For this reason, it is essential to design and develops methods capable of classifying among that news.

One of the goals of the third edition of MEX-A3T [3] is to tackle these problems and further improve the research of this critical NLP task, the detection of aggressive tweets and fake news in Mexican Spanish. In this work, we evaluate strategies proposed before, such as binary, TF, TF-IDF weighted representations and different preprocessing approaches as measuring the stop words and stemming importance to improve the features to observe the efficacy of these simple textual representations.

2. State of the art

The MEX-A3T is an evaluation forum with natural processing tasks. MEX-A3T 2020 is the third edition since the first edition carried on in 2018 [3].

The 2018 edition of the MEX-A3T shared represented the first attempt for organizing an evaluation forum for the analysis of social media content in Mexican Spanish. A variety of methods were proposed by participants, comprising content-based (bag of words, word n-grams, term vectors, dictionary words) and stylistic-based features (frequencies, punctuation, POS, Twitter-specific elements, slang words) as well as approaches based on neural networks (CNN, LSTM, and others) [4].

For the first edition, the organizers proposed two tasks: author profiling and aggressiveness identification. In both tasks, the baseline results were outperformed by most participants. For author profiling, the best results were obtained with an approach that emphasized the value of personal information for building the text representation. In the case of the aggressiveness identification, the winner team proposed an approach based on MicroTC and EvoMSA. MicroTC is a minimalistic text classifier independent from domain and language. EvoMSA is another text classifier that combines models (as MicroTC) with Genetic Programming [4].

For the second edition, the author profiling task added images information for each profile in the collection, whereas the aggressiveness identification task continued unchanged [5].

The participants proposed a variety of methodologies in the 2019 edition, from traditional supervised methods to deep learning approaches. For author profiling, the best results were obtained with an approach based on dimensionality reduction in text. However, their results did not overcome the best results from the 2018 edition. For aggressiveness identification, the top-ranked approach proposed two main kinds of features: character n-grams and word embeddings. Their results were equal to the previous year winner but employing a simpler approach [3].

For the 2020 edition, the tasks proposed are aggressiveness identification (but for this year, the organizers made a re-labeled over the collection) and fake news detection. Again, these tasks are proposed for the Mexican Spanish [3] [6].

Since, in the last edition, the simple approaches gave good results, we propose to apply basic methods of features selection and terms weighing to observe the scope of these simple but effective approaches in various tasks.

3. Methodology

Our methodology consists of three phases: data preprocessing, the weighting of words, and classification. Preprocessing deals with techniques to prepare the data sets by removing the elements that do not provide meaningful information in making the classification models. In this process, we first transform all the letters into lowercase, and then we tokenized the text with non-letter separators. After that, we removed the stop words such as “el”, “la”, “los”, “con”, which do not provide any valuable information in a sentence (commonly, words with less than two letters). Additionally, to discard most of the alignment errors and dismiss most of the less frequent words (which would cause a lot of false hits), we also deleted all the words that repeat less than five times. Finally, we performed the stemming of common words that could be meaningful for data interpretation. To test the effect of these last two steps (stemming and stop words deletion) in the performance of the proposed models, we made combinations of them throughout the experiments. We used the natural language toolkit (NLTK) for the steps of the tokenization and deletion of the stop words. While for the step of stemming, we used Snowball Stemmer.

The phase of the weighting of words includes measuring the relevance of each word concerning others. We used the following three feature-weighting methods: the binary occurrence (BO) [7], the term frequency (TF) [8], and the term frequency-inverse document frequency (TF-IDF) [9]. The binary occurrence, also called Boolean weighting, is an essential technique used to represent a word’s presence by using only two values: 0 and 1. We used this technique as a first approach because it is elementary and easy to implement. On the other hand, the TF approach weighs each feature based on the frequency in which a word (or term) appears. Finally, the TF-IDF approach proposes to assign reda lower weight to a term that appears in many documents than another occurring in a few documents. Both TF and TF-IDF feature-weighting methods were implemented via scikit-learn, a Python library for data mining and data analysis.

After the data was cleaned and weighted, we used well-known learning algorithms to build the classification models for the data sets. The learning algorithms we used were: support vector machine (SVM) [10] [11] [12], naive Bayes (NB), k-nearest neighbors (KNN) [13], and classification and regression trees (CART). We used the implementation of all these algorithms included in scikit-learn.

Finally, we trained our models by using 10-fold cross-validation, where each fold contains around 700 sentences, for each of the selected learning algorithms. We present our results in the next section.

4. Results and discussions

Table 1 shows the results for the data set of fake news [6] by performing a combination of the proposed classification models, feature-weighting methods, and preprocessing techniques. Concerning the classification models, we achieve the best result with the SVM model, followed by the decision tree and the KNN (with $k = 3$) models. Regarding the feature-weighting methods, it is not easy to see which one provides better results. However, it is noteworthy the poor results obtained by a particular combination of the BO method with the stemming technique and including the stop words. We also note that the use of stemming and the deletion of stop words do not have a significant impact on the results for most of the cases. The best result is achieved with an accuracy of 83% by using the SVM model with the TF method and the stemming technique without removing the stop words.

On the other hand, Table 2 shows the results for the data set of aggressiveness by using the same combinations performed on Table 1. Given these combinations, the SVM model always provides the best results for all the cases. In this regard, the decision tree and the KNN with $k = 5$ obtained the second and third places. Concerning the feature-weighting methods, we observe that the TF and TF-IDF methods significantly outperform the BO for all the cases. Using the TF-IDF, the SVM model achieves an accuracy of 84% when the deletion of the stop words is performed. For this best result, the stemming technique does not have a significant effect.

Note that in the aggressiveness table of the results exists rows that contain only zeros except in precision this because the model could not learn in the right way and always classify with the majority class. The results shown in both tables only belong to class 1; this is due to class 1 is the most important.

We decided to delete the words that repeat less than five times because this technique delivered the best value when we tested, doing that, we handled spelling mistakes, use of abbreviations, emoticons, etc. This is due to most of these do not repeat more than five times.

5. Conclusions

In this work, we address the tasks of fake news and aggressiveness identification for the 2020 MEX-A3T contest. For this aim, we built classification models using well-known machine learning algorithms such as SVM, NB, KNN, and decision trees. We conducted a comparative experimental procedure to study the impact of the proposed models using two data preprocessing techniques (stop words and stemming) and three feature-weighting methods (BO, TF, and TF-IDF). Experimental results indicate that, in general, the best classification model is SVM. They also show that the efficiency of the classification models was mainly influenced by the combination of data preprocessing methods instead of the feature-weighting methods. We noticed that it took a long time to execute all the sentences, so we decided to use 10-folds cross-validation to reduce the processing time. A limitation of this work is that the applied methods rely solely on a bag of words approach and not other textual representations. An interesting prospect for future work is to explore more advanced techniques, which hopefully allow us to get a better place in the next MEX-AT3 contests.

References

[1] M. Casavantes, R. López, L. C. González, Uach at mex-a3t 2019: Preliminary results on detecting aggressive tweets by adding author information via an unsupervised strategy, in: In Proceedings of the First Workshop for Iberian Languages Evaluation Forum (IberLEF 2019), CEUR WS Proceedings, 2019.

[2] D. M. Lazer, M. A. Baum, Y. Benkler, A. J. Berinsky, K. M. Greenhill, F. Menczer, M. J. Metzger, B. Nyhan, G. Pennycook, D. Rothschild, et al., The science of fake news, Science 359 (2018) 1094–1096.

[3] 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.

[4] M. Á. Álvarez-Carmona, E. Guzmán-Falcón, M. Montes-y Gómez, H. J. Escalante, L. Villasenor-Pineda, V. Reyes-Meza, A. Rico-Sulayes, Overview of mex-a3t at ibereval 2018: Authorship and aggressiveness analysis in mexican spanish tweets, in: Notebook Papers of 3rd SEPLN Workshop on Evaluation of Human Language Technologies for Iberian Languages (IBEREVAL), Seville, Spain, volume 6, 2018.

[5] M. E. Aragón, M. Á. Á. Carmona, M. Montes-y Gómez, H. J. Escalante, L. V. Pineda, D. Moctezuma, Overview of mex-a3t at iberlef 2019: Authorship and aggressiveness analysis in mexican spanish tweets., in: IberLEF@ SEPLN, 2019, pp. 478–494.

[6] J.-P. Posadas-Durán, H. Gómez-Adorno, G. Sidorov, J. J. M. Escobar, Detection of fake news in a new corpus for the spanish language, Journal of Intelligent & Fuzzy Systems 36 (2019) 4869–4876.

[7] H. Wang, P. Yin, J. Yao, J. N. Liu, Text feature selection for sentiment classification of chinese online reviews, Journal of Experimental & Theoretical Artificial Intelligence 25 (2013) 425–439. doi:10.1080/0952813X.2012.721139.

[8] Q. Ye, Z. Zhang, R. Law, Sentiment classification of online reviews to travel destinations by supervised machine learning approaches, Expert Systems with Applications 36 (2009) 6527–6535. doi:https://doi.org/10.1016/j.eswa.2008.07.035.

[9] F. Bravo-Marquez, M. Mendoza, B. Poblete, Meta-level sentiment models for big social data analysis, Knowledge-Based Systems 69 (2014) 86–99. doi:https://doi.org/10.1016/j.knosys.2014.05.016.

[10] B. E. Boser, I. M. Guyon, V. N. Vapnik, A training algorithm for optimal margin classifiers, in: Proceedings of the fifth annual workshop on Computational learning theory, 1992, pp. 144–152.

[11] V. Vapnik, The nature of statistical learning theory, Springer science & business media, 2013.

[12] C. Cortes, V. Vapnik, Support-vector networks, Machine learning 20 (1995) 273–297.

[13] L. E. Peterson, K-nearest neighbor, Scholarpedia 4 (2009) 1883. doi:10.4249/scholarpedia.1883, revision #137311.

Table 1
Results for the data set of fake news by using different classifier models, feature-weighting methods, and preprocessing techniques.

Model	Weighted	Stopwords	Stemming	Precision	Recall	F1-score	Accuracy
SVM	BO	Without	No	0.84	0.75	0.80	**0.81**
Naive Bayes	BO	Without	No	0.56	0.85	0.67	0.59
KNN (K=5)	BO	Without	No	0.52	0.34	0.41	**0.81**
KNN (K=3)	BO	Without	No	0.52	0.34	0.41	**0.81**
KNN (K=1)	BO	Without	No	0.52	0.35	0.42	0.51
Decision Tree	BO	Without	No	0.66	0.62	0.64	0.65
SVM	BO	With	No	0.49	0.49	0.49	0.49
NB	BO	With	No	0.49	0.27	0.35	**0.50**
KNN (K=5)	BO	With	No	0.49	0.27	0.35	**0.50**
KNN (K=3)	BO	With	No	0.50	0.28	0.36	**0.50**
KNN (K=1)	BO	With	No	0.48	0.46	0.47	0.48
Decision Tree	BO	With	No	0.48	0.43	0.45	0.48
SVM	BO	Without	Yes	0.81	0.74	0.77	**0.78**
NB	BO	Without	Yes	0.55	0.8	0.65	0.58
KNN (K=5)	BO	Without	Yes	0.10	0.01	0.02	0.51
KNN (K=3)	BO	Without	Yes	1.00	0.01	0.03	0.51
KNN (K=1)	BO	Without	Yes	0.50	0.33	0.39	0.5
Decision Tree	BO	Without	Yes	0.67	0.62	0.65	0.66
SVM	BO	With	Yes	0.79	0.74	0.77	**0.77**
NB	BO	With	Yes	0.55	0.8	0.65	0.58
KNN (K=5)	BO	With	Yes	1.00	0.01	0.02	0.51
KNN (K=3)	BO	With	Yes	1.00	0.02	0.03	0.51
KNN (K=1)	BO	With	Yes	0.49	0.32	0.39	0.49
Decision Tree	BO	With	Yes	0.67	0.64	0.66	0.66
SVM	TF	Without	No	0.82	0.81	0.82	**0.82**
NB	TF	Without	No	0.50	0.66	0.57	0.50
KNN (K=5)	TF	Without	No	0.82	0.62	0.70	0.74
KNN (K=3)	TF	Without	No	0.81	0.65	0.72	0.75
KNN (K=1)	TF	Without	No	0.73	0.63	0.67	0.70
Decision Tree	TF	Without	No	0.70	0.68	0.69	0.69
SVM	TF	With	No	0.83	0.81	0.82	**0.82**
NB	TF	With	No	0.51	0.66	0.57	0.51
KNN (K=5)	TF	With	No	0.83	0.61	0.70	0.74
KNN (K=3)	TF	With	No	0.80	0.64	0.71	0.74
KNN (K=1)	TF	With	No	0.73	0.63	0.68	0.70
Decision Tree	TF	With	No	0.67	0.69	0.68	0.67
SVM	TF	Without	Yes	0.86	0.74	0.80	**0.81**
NB	TF	Without	Yes	0.52	0.66	0.58	0.52
KNN (K=5)	TF	Without	Yes	0.94	0.22	0.35	0.60
KNN (K=3)	TF	Without	Yes	0.92	0.24	0.38	0.61
KNN (K=1)	TF	Without	Yes	0.73	0.27	0.39	0.58
Decision Tree	TF	Without	Yes	0.64	0.64	0.64	0.64
SVM	TF	With	Yes	0.85	0.81	0.83	**0.83**
NB	TF	With	Yes	0.52	0.67	0.58	0.52
KNN (K=5)	TF	With	Yes	0.79	0.62	0.69	0.73
KNN (K=3)	TF	With	Yes	0.78	0.64	0.70	0.73
KNN (K=1)	TF	With	Yes	0.73	0.64	0.68	0.70
Decision Tree	TF	With	Yes	0.68	0.68	0.68	0.68
SVM	TF-IDF	Without	No	0.82	0.69	0.75	**0.77**
NB	TF-IDF	Without	No	0.50	0.66	0.57	0.49
KNN (K=5)	TF-IDF	Without	No	0.58	0.09	0.16	0.51
KNN (K=3)	TF-IDF	Without	No	0.49	0.24	0.32	0.49
KNN (K=1)	TF-IDF	Without	No	0.47	0.75	0.58	0.45
Decision Tree	TF-IDF	Without	No	0.71	0.68	0.69	0.70
SVM	TF-IDF	With	No	0.82	0.81	0.81	**0.81**
NB	TF-IDF	With	No	0.50	0.66	0.57	0.50
KNN (K=5)	TF-IDF	With	No	0.66	0.55	0.60	0.63
KNN (K=3)	TF-IDF	With	No	0.62	0.56	0.59	0.74
KNN (K=1)	TF-IDF	With	No	0.50	0.59	0.54	0.50
Decision Tree	TF-IDF	With	No	0.72	0.71	0.72	0.72
SVM	TF-IDF	Without	Yes	0.80	0.72	0.76	**0.77**
NB	TF-IDF	Without	Yes	0.51	0.66	0.58	0.51
KNN (K=5)	TF-IDF	Without	Yes	0.57	0.05	0.09	0.51
KNN (K=3)	TF-IDF	Without	Yes	0.41	0.08	0.14	0.48
KNN (K=1)	TF-IDF	Without	Yes	0.43	0.22	0.29	0.47
Decision Tree	TF-IDF	Without	Yes	0.62	0.62	0.62	0.62
SVM	TF-IDF	With	Yes	0.80	0.78	0.79	**0.79**
NB	TF-IDF	With	Yes	0.51	0.67	0.58	0.52
KNN (K=5)	TF-IDF	With	Yes	0.81	0.28	0.41	0.61
KNN (K=3)	TF-IDF	With	Yes	0.71	0.31	0.43	0.59
KNN (K=1)	TF-IDF	With	Yes	0.56	0.39	0.46	0.54
Decision Tree	TF-IDF	With	Yes	0.64	0.66	0.65	0.64

Table 2
Results for the data set of aggressiveness using different classifier models, feature-weighting methods, and preprocessing techniques

Model	Weighted	Stopwords	Stemming	Precision	Recall	F1-score	Accuracy
SVM	BO	Without	No	0.00	0.00	0.00	**0.71**
NB	BO	Without	No	0.29	0.90	0.44	0.33
KNN (K=5)	BO	Without	No	0.00	0.00	0.00	**0.71**
KNN (K=3)	BO	Without	No	0.29	0.40	0.34	0.54
KNN (K=1)	BO	Without	No	0.29	0.90	0.44	0.33
Decision Tree	BO	Without	No	0.00	0.00	0.00	**0.71**
SVM	BO	With	No	0.00	0.00	0.00	**0.71**
NB	BO	With	No	0.29	0.90	0.44	0.33
KNN (K=5)	BO	With	No	0.00	0.00	0.00	**0.71**
KNN (K=3)	BO	With	No	0.29	0.70	0.41	0.42
KNN (K=1)	BO	With	No	0.00	0.00	0.00	**0.71**
Decision Tree	BO	With	No	0.00	0.00	0.00	**0.71**
SVM	BO	Without	Yes	0.00	0.00	0.00	**0.71**
NB	BO	Without	Yes	0.29	0.90	0.44	0.33
KNN (K=5)	BO	Without	Yes	0.28	0.10	0.14	0.67
KNN (K=3)	BO	Without	Yes	0.29	0.40	0.34	0.54
KNN (K=1)	BO	Without	Yes	0.28	0.10	0.14	0.67
Decision Tree	BO	Without	Yes	0.00	0.00	0.00	**0.71**
SVM	BO	With	Yes	0.00	0.00	0.00	**0.71**
NB	BO	With	Yes	0.29	0.90	0.44	0.33
KNN (K=5)	BO	With	Yes	0.28	0.10	0.14	0.67
KNN (K=3)	BO	With	Yes	0.29	0.40	0.34	0.54
KNN (K=1)	BO	With	Yes	0.28	0.10	0.14	0.67
Decision Tree	BO	With	Yes	0.00	0.00	0.00	**0.71**
SVM	TF	Without	No	0.72	0.64	0.68	**0.82**
NB	TF	Without	No	0.33	0.82	0.47	0.46
KNN (K=5)	TF	Without	No	0.66	0.29	0.40	0.75
KNN (K=3)	TF	Without	No	0.62	0.33	0.43	0.75
KNN (K=1)	TF	Without	No	0.52	0.39	0.45	0.72
Decision Tree	TF	Without	No	0.62	0.57	0.60	0.78
SVM	TF	With	No	0.71	0.64	0.67	**0.82**
NBayes	TF	With	No	0.33	0.83	0.47	0.46
KNN (K=5)	TF	With	No	0.65	0.27	0.38	0.75
KNN (K=3)	TF	With	No	0.62	0.30	0.41	0.75
KNN (K=1)	TF	With	No	0.52	0.37	0.43	0.72
Decision Tree	TF	With	No	0.63	0.62	0.63	0.79
SVM	TF	Without	Yes	0.70	0.61	0.65	**0.81**
NB	TF	Without	Yes	0.32	0.86	0.46	0.43
KNN (K=5)	TF	Without	Yes	0.75	0.27	0.40	0.76
KNN (K=3)	TF	Without	Yes	0.65	0.32	0.43	0.76
KNN (K=1)	TF	Without	Yes	0.57	0.42	0.48	0.74
Decision Tree	TF	Without	Yes	0.61	0.60	0.60	0.78
SVM	TF	With	Yes	0.71	0.64	0.68	**0.82**
NB	TF	With	Yes	0.31	0.86	0.46	0.42
KNN (K=5)	TF	With	Yes	0.68	0.27	0.39	0.75
KNN (K=3)	TF	With	Yes	0.60	0.30	0.40	0.74
KNN (K=1)	TF	With	Yes	0.53	0.37	0.44	0.72
Decision Tree	TF	With	Yes	0.62	0.61	0.62	0.78
SVM	TF-IDF	Without	No	0.82	0.53	0.64	**0.83**
NB	TF-IDF	Without	No	0.33	0.82	0.47	0.47
KNN (K=5)	TF-IDF	Without	No	0.69	0.24	0.36	0.75
KNN (K=3)	TF-IDF	Without	No	0.61	0.30	0.40	0.74
KNN (K=1)	TF-IDF	Without	No	0.50	0.38	0.43	0.71
Decision Tree	TF-IDF	Without	No	0.61	0.59	0.60	0.77
SVM	TF-IDF	With	No	0.82	0.55	0.66	**0.84**
NB	TF-IDF	With	No	0.33	0.82	0.47	0.47
KNN (K=5)	TF-IDF	With	No	0.70	0.18	0.28	0.74
KNN (K=3)	TF-IDF	With	No	0.62	0.25	0.35	0.74
KNN (K=1)	TF-IDF	With	No	0.50	0.31	0.38	0.71
Decision Tree	TF-IDF	With	No	0.61	0.59	0.60	0.77
SVM	TF-IDF	Without	Yes	0.79	0.54	0.64	**0.83**
NB	TF-IDF	Without	Yes	0.32	0.85	0.46	0.43
KNN (K=5)	TF-IDF	Without	Yes	0.71	0.15	0.25	0.74
KNN (K=3)	TF-IDF	Without	Yes	0.64	0.23	0.34	0.74
KNN (K=1)	TF-IDF	Without	Yes	0.52	0.37	0.43	0.72
Decision Tree	TF-IDF	Without	Yes	0.61	0.59	0.60	0.78
SVM	TF-IDF	With	Yes	0.81	0.58	0.67	**0.84**
NB	TF-IDF	With	Yes	0.32	0.85	0.46	0.42
KNN (K=5)	TF-IDF	With	Yes	0.82	0.12	0.22	0.74
KNN (K=3)	TF-IDF	With	Yes	0.71	0.17	0.27	0.74
KNN (K=1)	TF-IDF	With	Yes	0.54	0.26	0.35	0.72
Decision Tree	TF-IDF	With	Yes	0.61	0.58	0.60	0.77