LUC at ComMA-2021 Shared Task: Multilingual Gender Biased and Communal Language Identification without using linguistic features

Rodrigo Cuéllar-Hidalgo
CENIDET/COLMEX
rcuellar@colmex.mx

Julio de Jesús Guerrero-Zambrano
IIM
julio@jzambrano.xyz

Dominic Forest
Université de Montréal
EBSI
dominic.forest@umontreal.ca

Gerardo Reyes-Salgado
CENIDET
gerardo.rs@cenidet.tecnm.mx

Juan-Manuel Torres-Moreno
Université d'Avignon
LIA
juan-manuel.torres@univ-avignon.fr

Abstract

This work aims to evaluate the ability that both probabilistic and state-of-the-art vector space modeling (VSM) methods provide to well known machine learning algorithms to identify social network documents to be classified as aggressive, gender biased or communally charged. To this end, an exploratory stage was performed first in order to find relevant settings to test, i.e. by using training and development samples, we trained multiple algorithms using multiple vector space modeling and probabilistic methods and discarded the less informative configurations. These systems were submitted to the competition of the ComMA@ICON'21 Workshop on Multilingual Gender Biased and Communal Language Identification.

1 Introduction

The introduction of the Internet and its democratization in the public sphere has fostered the emergence of many sociological phenomena. This opens the possibility of forming friendly relations and information sharing from online networking platforms (Arroyo-Fernández et al., 2018). As the organizers say: *"Aggression and its manifestations in different forms have taken unprecedented proportions with the tremendous growth of Internet and social media."* [1]. The challenge ComMA@ICON 2021 is an interesting task in order to automatically discover and understand the pragmatic and structural aspects of such forms of language usage (Waseem et al., 2017; Dadvar et al., 2013).

Our international LUC[2] team[2] have worked in such challenge using several approaches mainly without linguistic features.

This paper is organized as follows: the section 2 presents some relevant state of the art work related to automatic aggression identification, section 3 describes the methodology and the ComMA@ICON'21 dataset used, section 4 shows the results and finally the section 5 analyzes and discusses the contributions of this paper.

2 Related work

There already exist software designed to identify aggression and cyberbullying in social medias, e.g. CyberPatrol. The main drawback of these systems is that they are based on keyword filtering, which is a limitation because no statistical features of texts are taken into account. Further, these keyword filtering methods require manual maintenance. To overcome the limitations of keyword filtering systems, (Yin et al., 2009) is one of the former attempts to detect cyberbullying by using statistical features: word frequency, analysis of feelings (use of pronouns in the second person, insults, etc.) and context. (Dinakar et al., 2021) built a system that can detect bullying elements in commentaries of YouTube videos. These were classified according to different representative categories (sexuality, intelligence, race and physical attributes). The classification revealed weaknesses and an increase in false positive cases. Researchers emphasized the importance of using common sense to understand users' goals, emotions, and relationships, thereby disambiguating and contextualizing language. In (Berry and Kogan, 2010) the authors were also interested in a word search method based on a bag of words (BoW) system incorporating sentiment and contextual analysis. They build a decision tree that predicted intimidating messages with an accuracy rate of 0.93. The researchers also have developed

[1] https://competitions.codalab.org/competitions/35482

[2] Team composed by LIA (Laboratoire Informatique d'Avignon/Université d'Avignon, France), EBSI/Université de Montréal (Québec, Canada), CENIDET (Centro Nacional de Investigación y Desarrollo Tecnológico, Cuernavaca, México) and COLMEX (the Colegio de México, CDMX, México).

Proceedings of the 18th International Conference on Natural Language Processing: Shared Task on Multilingual Gender Biased and Communal Language Identification, pages 41–45
National Institute of Technology - Silchar, December 16-19, 2021.

the Chatcoder software to detect malicious activities online (Kontostathis et al., 2012).

Another study tested a system that allows users of a website to control the messages posted on their web pages: it customized vocabulary filtering criteria using a machine learning method that automatically labeled the contents. This approach had limitations because it was unable to measure relationships between terms beyond a certain semantic level (Dadvar et al., 2012). (Nahar et al., 2014) provided a concrete method for detecting online harassment by measuring the score of sent and received messages (and thus their degree of involvement in a conversation) using the Hyper-link Induced Topic Research algorithm (HITS). The authors also proposed a graphical model that identifies the aggressors and their most active victims. Other studies have attempted to go further by seeking to take into account more specific characteristics. (Dadvar et al., 2012) tried to establish a system based on language features characterizing the author's genre of comments on MySpace. Their results revealed an improvement in the detection of bullying when this information is taken into account. As we can see, recent work defines the means to respond to the cyberbullying phenomenon that is becoming more and more widespread as the use of the web does.

The problem of identifying offensive and abusive language is a more difficult task than expected due to the prevalence of 5 factors, according to (Nobata et al., 2016), described below:

1. The intentional obfuscation of words that lead to false positives.
2. The difficulty in identifying racial slurs since depending on the target group, this can be offensive or flattering.
3. The grammatical fluidity that leads to false negatives.
4. The limit of sentences, that is, abusive language can be articulated in more than one sentence.
5. The sarcasm, which is even difficult for a human to interpret, implies a lot of knowledge about the context of the message (geographical, historical, social, etc).

Other aspects detected by Nobata et al. (2016) corresponds to the heterogeneity in the approach to the problem itself that causes too much noise and confusion, such as the fact of only addressing specific aspects, specific contexts, different definitions for certain terms and / or domain, and finally different sets of assessment.

At present, the task of classifying text in "agglutinating" or "morphologically rich" languages, leaves aside classical preprocessing and is replaced by the use of deep neural networks and word embeddings, since they take into account the semantic distance of the words in context, which is very useful in categorization tasks, which is not the case with the classic bag of words methods. A clear example of this is the implementation of fastText for the Turkish language by Kuyumcu et al. (2019).

3 Methodology and data-sets

In this section we present the methodology and the data-sets used in our study.

3.1 Data-sets

For this task, the ComMA organizers have provided a multilingual data-set with a total of 12,000 instances for training and development and an overall 3,000 instances for testing. The corpus is in three Indian languages Meitei, Bangla (an Indian variety) and Hindi and English. Several instances are expressed in two our more languages. Refer to the challenge website for further information[3].

From the organizer's website, the corpus is labelled as follows:

1. **Aggression level**. This category gives a classification in 'Overtly Aggressive'(OAG), 'Covertly Aggressive'(CAG) or 'Non-aggressive'(NAG) text.
2. **Gender**. This category classify the text as 'gendered'(GEN) or 'non-gendered'(NGEN).
3. **Communal**. The task is to develop a binary classifier for classifying the content as 'communal' (COM) or 'non-communal'(NCOM).

We confirmed that the data-set furnished (both training and validation) are too small in order to employ Deep Learning methods. We then chose to use mainly classical probabilistic and VSM (Salton et al., 1983) oriented algorithms.

[3] https://competitions.codalab.org/competitions/35482#learn_the_details-datasets

Corpus	Instances	Tokens	Chars
Training	9,000	186,017	1 585,979
Developing	3,000	55,996	473,403
Testing	3,000	82,367	815,104

Table 1: Basic statistics from ComMA corpus.

3.2 Pre-processing

Because there are a mixture of several languages (Meitei, Bangla, Hindi and English) and often the data-set instances presents two or more languages mixed, we decided of avoid any linguistic pre-processing. Indeed, neither stemmer, filtering or lemmatizer was used in our study. The only pre-processing that was carried out was the removal of NaN and extraction of basic characteristics of each message, which are listed below:

- Number of words.
- Number of sentences.
- Number of scores.
- Number of numbers.
- Number of unrecognizable characters (emojis).

Using a simple tokenizer written in Python.

3.3 Algorithms

To tackle the problem presented in this challenge, we develop LUC, a multi-classifier that uses the following algorithms:

- Nearest-Neighbor algorithm (KNN) (Altman, 1992);
- Naive Bayes method (Lewis, 1998);
- Support Vector Machine algorithm (SVM) (Cortes and Vapnik, 1995);
- Random Forest algorithm (Breiman, 2001);
- Generalized Boosted Regression Models (GBM) [4];
- Adaboost (Freund and Schapire, 1997);
- Neural Networks (NN) based algorithms (Hopfield, 1982).

[4]https://github.com/gbm-developers/gbm

In the first system (S1), the individual outputs of the algorithms Naive Bayes, SVM, Random Forest, GBM, Adaboost and a multi-layer perceptron were combined in a single output using a mixing strategy that assembles all of the models that were created using the previous algorithms. In order to combine the predictions of the previously mentioned estimators, it was necessary to stack the outputs of each individual estimator and use a final estimator to compute the final prediction.

The stacking classifier responsible to compute the final estimation takes every individual estimator as input and creates a final estimator by training cross-validated predictions of the base estimators. For each estimator, the final classifier computes the prediction probability and final prediction, to return a final estimation based on a logistic regression of the inputs.

In order to achieve better results, each one of the tasks were trained and executed independently (by language and category) and the results were combined at the end. Accuracy was measured in order to keep track of the metrics of the results.

In the second system (S2), we also explored the relevance of the K nearest neighbors (KNN) algorithm alone to perform these supervised classification tasks. This algorithm is well known in the field of machine learning. It is both simple and efficient. It consists in assigning to each document of the test set the category with the highest frequency among its K nearest neighbors. The cosine measurement was used to evaluate the distance between the vectors representing each document. In addition, we varied two main parameters during the learning phase. On the one hand, we have varied the value of K, that is to say the number of neighbors to be considered. We systematically compared the results using 1, 2, 3, 4, 5, 10, 15, 20, 25 and 50 neighbors. We also varied the number of features to describe the documents. As mentioned previously, no preprocessing was applied to reduce the size of the initial lexicon. Based on the frequency of strings in the entire corpus and by evaluating the correlation between the most frequent strings and the categories to predict (using a simple Chi2 measure), we used from 500 to 30,000 strings of characters (in increments of 500) to describe the training corpus. During the learning phase, we obtained the best results using 30,000 features and K = 1. It is therefore this combination that we applied to the test corpus (system 5).

	KNN (S2)	Combined (S1)
Accuracy	0.98	0.7746

Table 2: Developing set evaluation of our best classifiers.

F1	KNN (S2)	Combined (S1)
Instance F1	0.234	0.213
Overall micro-F1	0.615	0.616
A. micro-F1	0.446	0.389
G. bias micro-F1	0.675	0.693
C. bias micro-F1	0.726	0.766

Table 3: Testing set evaluation for our classifiers. See: https://competitions.codalab.org/competitions/public_submissions/35482.

4 Results

We present our results in two parts. In the first one we show the accuracy of methods on the training set. In the second one, the official F1 measures from the CodaLab website of the organizers are showed in their "Results" page[5].

4.1 Training and Developing data-sets

For the training set, our main results are showed in the Table 2.

In the training phase, the accuracy was our principal criterion to optimize on the Developing dataset.

4.2 Testing set

The main results on the testing set are shown in the Table 3. It is possible to see that our classifier LUC obtains the 3rd rank on the multilingual task, with a overall-F1 measure of **0.234** (see Addendum).

5 Discussion and Conclusions

This task was very challenging. The corpus (train, developing and test) are very noisy at several levels (syntactic, lexical, semantic, etc.), and the instances presents two or more languages mixed. There are instances that make no sense. By example, the instances from train and developing sets:

```
C575.31 rrrrrrrrrrrrrrrrrrrrrrrrr
rrrrrrrrrrrrrrrrrrrrrrrrrrrrrurrr
rrrrrrrrrrrrryrrrr (NAG,NGEN,NCOM)

C575.690.1  XXXXXXXXXXXXXXXXXXXXX
(NAG,NGEN,NCOM)

C579.732.1 G (NAG,NGEN,COM)
```

where the string of "r" of the instance C575.31 represents a string of a dog's emojis, is a good example of semantic noise. What was the logic (if any) of the annotators in assigning the classes (NAG,NGEN,NCOM) to this instance?

However, our strategy has obtained the third rank on overall-F1 measure and the second rank on the Aggression micro-F1 measure. The methodology that we adopted in this study allowed us to observe in broad strokes the complexity of the challenge.

We think that our model may be enriched by other model sentences representation, as followed in (Arroyo-Fernández et al., 2019), in order to outperform our results. Finally, sentences classified with labels 'aggression', 'genre' or 'communal levels' may be used in other NLP tasks, by example on the summaries generation guided by a specific context and coming from social network text data (Torres-Moreno, 2014).

Acknowledgments

This work was partially financed by the COLMEX (Mexico), CENIDET (Mexico), Instituto de Ingenieros de Morelos (Mexico), LIA (Université d'Avignon, France) and by the EBSI (Université de Montréal, Québec).

Addemdum

We dispute the results mentioned on the challenge's website. Indeed, the organizers of the challenge allowed certain teams to benefit from an additional day for the submission of their result. This violates the original rules of the challenge and has favored some teams. Strangely, the teams that benefited from this advantage finished in the top 3 positions. We therefore consider, by virtue of the initial regulations that we respected, that we finished in 3rd place in the task on the multilingual corpus. We are also very amazed at how the results have evolved over time. The results were first published online on the challenge website (we were then 3rd), then these results were modified (we were then ranked 5th). Finally, the non-public excel sheet was sent to the teams with new results, without any explanation (we ended up in 6th place).

[5]See: https://competitions.codalab.org/competitions/35482#results

References

N. S. Altman. 1992. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185.

Ignacio Arroyo-Fernández, Dominic Forest, Juan-Manuel Torres-Moreno, Mauricio Carrasco-Ruiz, Thomas Legeleux, and Karen Joannette. 2018. Cyberbullying detection task: the EBSI-LIA-UNAM system (ELU) at coling'18 TRAC-1. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying, TRAC@COLING 2018, Santa Fe, New Mexico, USA, August 25, 2018*, pages 140–149. Association for Computational Linguistics.

Ignacio Arroyo-Fernández, Carlos-Francisco Méndez-Cruz, Gerardo Sierra, Juan-Manuel Torres-Moreno, and Grigori Sidorov. 2019. Unsupervised sentence representations as word information series: Revisiting tf–idf. *Computer Speech Language*, 56:107–129.

Michael W. Berry and Jacob Kogan, editors. 2010. *Text Mining: Applications and Theory*. Wiley, Chichester, UK.

Leo Breiman. 2001. Random forests. *Machine Learning*, 45(1):5–32.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning*, 20(3):273–297.

M. Dadvar, Franciska M.G. de Jong, Roeland J.F. Ordelman, and Rudolf Berend Trieschnigg. 2012. Improved cyberbullying detection using gender information. In *Proceedings of the Twelfth Dutch-Belgian Information Retrieval Workshop (DIR 2012)*, pages 23–25. Ghent University. Null ; Conference date: 24-02-2012 Through 24-02-2012.

Maral Dadvar, Dolf Trieschnigg, Roeland Ordelman, and Franciska de Jong. 2013. Improving cyberbullying detection with user context. In *Advances in Information Retrieval*, pages 693–696, Berlin, Heidelberg. Springer Berlin Heidelberg.

Karthik Dinakar, Roi Reichart, and Henry Lieberman. 2021. Modeling the detection of textual cyberbullying. *Proceedings of the International AAAI Conference on Web and Social Media*, 5(3):11–17.

Yoav Freund and Robert E Schapire. 1997. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139.

J J Hopfield. 1982. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558.

April Kontostathis, Andy Garron, Kelly Reynolds, Will West, and Lynne Edwards. 2012. Identifying predators using chatcoder 2.0. In *CLEF*.

Birol Kuyumcu, Cuneyt Aksakalli, and Selman Delil. 2019. An automated new approach in fast text classification (fasttext): A case study for turkish text classification without pre-processing. *PervasiveHealth: Pervasive Computing Technologies for Healthcare*, pages 1–4.

David D. Lewis. 1998. In *Proceedings of ECML-98, 10th European Conference on Machine Learning*, 1398, pages 4–15, Chemnitz, DE. Springer Verlag, Heidelberg, DE.

Vinita Nahar, Xue Li, Hao Lan Zhang, and Chaoyi Pang. 2014. Detecting cyberbullying in social networks using multi-agent system. *Web Intell. Agent Syst.*, 12(4):375–388.

Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. 2016. Abusive Language Detection in Online User Content. In *Proceedings of the 25th International Conference on World Wide Web*, pages 145–153. International World Wide Web Conferences Steering Committee.

Gerard Salton, Edward A Fox, and Harry Wu. 1983. Extended boolean information retrieval. *Communications of the ACM*, 26(11):1022–1036.

Juan-Manuel Torres-Moreno. 2014. *Automatic Text Summarization*. ISTE Ltd, John Wiley & Sons, Inc., London.

Zeerak Waseem, Thomas Davidson, Dana Warmsley, and Ingmar Weber. 2017. Understanding abuse: A typology of abusive language detection subtasks. In *Proceedings of the First Workshop on Abusive Language Online*, pages 78–84, Vancouver, BC, Canada. Association for Computational Linguistics.

Dawei Yin, Zhenzhen Xue, Liangjie Hong, Brian Davison, April Edwards, and Lynne Edwards. 2009. Detection of harassment on web 2.0.