HalBERT at PoliticEs 2022: Are Machine Learning Algorithms better for Author Profiling?

Cristina García Holgado[1,*,†], Aman Sinha[2,†]

[1]Lattice, École Normale Supérieure, Paris, France
[2]Université de Lorraine, Nancy, France

Abstract

This paper presents an investigation of different machine and deep learning models for Author Profiling for the Shared Task *PoliticEs: Spanish Author Profiling for Political Ideology* at IberLEF 2022. We provide a broad comparative study to see where both approaches stand. Our analysis show that ML methods are more effective than DL methods. Overall, our system was ranked 5th on the leaderboard based on the average macro F1-score across all sub-tasks.

Keywords

Author profiling, Political Ideology classification, Text classification, Natural Language Processing, Machine Learning

1. Introduction

Social networks have become an important place of political debate in the past few years, including issues that are not originally of a political nature. Recent events such as Covid-19 pandemic have displayed a complex sociopolitical scenario where individuals and political personalities' views and responses on a subject seem to differ significantly depending on their political ideology [1, 2]. Previous works have shown in fact a correlation between personality traits and political affiliation [3]. To improve communication campaigns from public authorities, studying the different demographic and behavioral attributes of the population could favor better communication considering the complex communicative setting of social networks.

From the NLP perspective, this is a problem of Author Profiling where political ideology is considered as a psychographic trait among demographic traits such as age, ethnicity, religion, gender, etc. In this context, the challenge *PoliticEs: Spanish Author Profiling for Political Ideology* [4] was organized within the workshop IberLEF 2022. The challenge targeted the identification of political ideology in Spanish tweets. Here, we refer to the political ideology of a given tweet's author regarding its political spectrum and consider two demographic traits such as profession and gender. The identification of political ideology is addressed from both a binary and multi-class perspective.

IberLEF 2022, September 2022, A Coruña, Spain
*Corresponding author.
†These authors contributed equally.
cristina.gholgado@gmail.com (C. G. Holgado); aman.sinha@univ-lorraine.fr (A. Sinha)
0000-0003-4882-2532 (C. G. Holgado); 0000-0003-1608-5734 (A. Sinha)

Dataset split	# Original Sample	# Utilized Sample
Train set	37,561	31,846
Development	5,000	4,496
Development _test set	1,000	1,123
Blind Test set	12,601	-

Table 1
Dataset statistics

The paper is structured as follows: Section 2 presents related work on Author Profiling. Section 3 provides a description of the original and the utilized dataset for the task. Section 4 describes the different preprocessing steps and presents the various techniques and features that compose our experimental setup. Section 5 covers the different machine and deep learning methods. Section 6 provides the implementation details including hyperparameters setting and ensembling technique for both machine and deep learning models. Section 7 presents the results of the experiments for the different analysis we conducted. Finally, we analyze the results of the experiments on section 8 and conclude the paper on section 9.

2. Related Work

In the last years, author profiling has become an important area in NLP as it provides powerful support in the analysis and fight of diverse problems on social media. Recent tasks have primarily focused on fake news [5], hate speech detection [6], aggressiveness analysis [7] or bot identification [8], while others have addressed author profiling for the identification of demographic information such as gender, age or profession of the authors [5, 9].

Machine learning approaches are found to be comparative and sometimes better than deep learning methods [5]. N-grams [5], topic information [10], linguistic cues [9] such as syntactic, semantic and stylistic features have been used previously for author profiling. Pre-trained embeddings such as fastText, LASER [11], Universal Sentence Encoder (USE) [12] have also been used to capture more information than term frequency-based features. Transformer-based embeddings from BERT [13] are also evaluated for their effectiveness in capturing contextual information. Transfer learning using multilingual BERT-based models [14] was also found to be effective in previous studies for other text classification problems.

3. Dataset

The dataset is an extension of the PoliCorpus 2020 [15] and it is composed of extracted tweets where the users are Spanish politicians and political journalists from different newspapers. Names and mentions of political parties on the text are anonymized. For each user, multiple tweets were extracted. The provided training set is composed of tweets from 257 different anonymous users, each one including 120 tweets on average. The development set consists of 100 users containing 50 tweets. The test set consists of 20 users containing 50 tweets. Each

Class	Label	Train samples	Dev-Test samples	Dev samples
Profession	Politician	25,694	3,404	1,003
	Journalist	6,152	1,092	120
Gender	Male	18,190	2,325	701
	Female	13,656	2,171	422
Ideology Binary	Right	13,093	2,648	360
	Left	18,753	1,848	763
Ideology Multiclass	Right	3,797	960	120
	Moderate Right	9,296	1,688	240
	Moderate Left	10,714	1,106	403
	Left	8,039	742	360

Table 2
Label distribution for the utilized dataset

user is labeled with four different classes: gender, profession, ideology binary, and ideology multiclass.

For our experiments, we modified the original dataset due to duplicate tweets present both in the training and the development set. We removed any duplicates and we did a new splitting for training and evaluation of the models. Table 1 shows the statistics of the original and the utilized dataset. The distribution of the labels for each class is shown in Table 2.

4. Experimental Setup

4.1. Data Preprocessing

Anonymization In the source data, most of the mentioned political figures and parties were anonymous. We anonymized the remaining ones using a custom list of Spanish political parties and predominant political figures.

Normalization As we considered hashtags provide meaningful information and are sometimes used as constituents of an utterance in the tweets, i.e. *Hoy en Comisión de #Transportes, #Movilidad y #AgendaUrbana* [...], we split them to a single word or to sentence. This latter refers to hashtags that are composed of multiple words and thus, represent a sentence, i.e. *#VidaDignaMuerteDigna*. Following this operation, we changed the abbreviated units such as ”q”, ”k”, ”d” to their full forms ”que”, ”de”. We normalized double-gender inflections such as ”-e/-a” or ”-os/-as” to the neutral masculine form ”-e” or ”-os” to avoid lemmatization errors.

Lemmatization The data was lemmatized using the Stanza NLP library [16]. Only lexical parts of speech were kept (nouns, verbs, adverbs and adjectives). Several stopwords were excluded, such as *”hacer”, ”estar”, ”decir”, ”tener”, ”haber”, ”hoy”, ”ayer”, ”mañana”*.

4.2. Data Augmentation

In order to cope up with the imbalance problem of the data (refer table 2), we experimented with a few augmentation techniques:

Synthetic Minority Oversampling Technique (SMOTE) For machine learning models, we used SMOTE [17] as the oversampling technique to fix the imbalance problem in the dataset at the representation level. This technique generates samples next to the original samples using a K-Nearest Neighbors classifier without making any distinction between easy and hard samples to be classified.

Text Augmentation For deep learning models, we used the `nlpaug` library [18] for augmenting at input level i.e. the text. This library provides various ways to augment text such as by character substitute, OCR, random word, antonym, synonym, contextual word substitute, etc. We use `SynonymAug` which substitutes randomly chosen words in a sentence with their synonyms from WordNeT/ Paraphrase Database (PPDB) resource to generate augmented samples.

4.3. Feature Engineering

We used various traditional text representation features such TF-IDF, count, N-gram features, GloVe embeddings, LASER embeddings and emoji features.

- **Count and TF-IDF Vectorizer**: We use the Scikit-learn library to create document-term and inverse document term frequency-based features from the corpus.
- **N-gram**: We used a combination of uni-, bi- and trigrams with count (cv) and TF-IDF (tfidf) vectorizers feature extractor to generate N-gram features.
- **GloVE**: We make use of custom GloVe embeddings [19] which we generated from the utilized dataset using the `Gensim` library to produce 100 length embeddings.
- **Laser**: We also used a pre-trained embedding and language agnostic sentence representation (LASER) to evaluate the representation power of BiLSTM against GloVe embeddings.
- **Emoji**: Considering emoji use (frequency and types) among users as a potential distinctive feature [20], we extracted them from every tweet. However, we did not take into consideration the usage pattern as consecutive or single-use emoji on a tweet. We use emoji2vec [21] to transform the extracted emoji into numerical features for the machine learning models.
- **Topics**: Under the hypothesis that some topics may be more prevalent among certain groups [10] and thus, help identify the political position, we extracted the topics of the dataset using the implementation of LDA Mallet on Gensim library. We found 14 topics to be the optimal number of topics based on the best topic coherence score per number of topics along with a manual validation of the topic interpretability. Specifically, if the provided keywords for each topic were able to delimiter themselves a topic which is differentiated from the others as shown in the example below:
  t_1 *crisis, empleo, plan, economico, empresa, trabajador, social, sector*
  t_2 *madrid, pandemia, salud, caso, sanitario, comunidad, covid, gestion*

t_3 *derecho, ley, social, cumplir, justicia, igualdad, aprobar, lucha*
t_4 *partido, votar, derecha, politico, voto, eleccion, izquierda, gobernar*
...

5. Methods

5.1. ML

We looked into various conventional machine learning algorithms such as Support Vector Machine [22], Naive Bayes [23], Random Forest [24], GBoost classifier [25], MLP [26] and SGD Classifier [27] for the traditional word representation features.

5.2. DL

For the deep learning models, we experimented with mainly two architectures described below:

- **BERTSimple**: We use a simple BERT architecture which is attached with different language model encoders, with a sequence classification head. This model is our baseline for the deep learning experimental setup.
- **BERTCNN**: Previous work [28] has shown that Convolutional Neural Network (CNN) can be useful for text classification. Therefore, we use a CNN encoder variant of BERT simple architecture as another model for our deep learning experimental setup.

For the input encoder, we experiment with the language models described below:

- **SPANISH BERT (BETO)** [29]: This is a BERT-base model trained with Spanish text from Wikipedia and source of the OPUS Project [30] with masked language modeling (MLM) objective using dynamic masking and whole word masking techniques.
- **MULTILINGUAL BERT (MBERT)** [31]: This is a BERT transformer model that was pre-trained with a large corpus of multilingual data from 104 languages in a self-supervised fashion with MLM and next sentence prediction (NSP) objective.
- **XLM-ROBERTA-BASE (XLMRB)** [32]: This is the mulitlingual version of the original RoBERTa [33] trained on CommonCrawl data containing 100 languages with MLM objective.

6. Implementation

In this section, we provide the hyperparameter details of all the models that we used in our experimental analysis. All machine learning models were trained and evaluated with user aggregated tweet features that had undergone preprocessing as inputs (ref. 4.1), whereas the deep learning models were trained and evaluated on a non aggregated and non-preprocessed version of the tweets [13] because of the 512 token limit in the BERT models. The code is available at our Github repository.

6.1. ML

For machine learning models, we performed an exhaustive search over hyperparameters using `GridSearchCV` for different models along with Stratified KFold data split. Below we list out the paramaters for each ML model.

- **SVM** For SVM, we considered *C* from 0.001 to 1000, *kernel* for linear and rbf, and *gamma* from 0.001 to 1000. Initialization was done only with random_state=1.
- **Naive Bayes** For NB, we considered *fit_prior* for True and False, and *alpha* from 0 to 1. Initialization was done only with fit_prior=True.
- **Multi Layer Perceptron** For MLP, we considered *activation* from logistic, tanh, relu; *alpha* from 0.0001 to 0.1; for *learning_rate* from constant , invscaling, adaptive; and *solver* from lbfgs, sgd, adam. Initialization was done only with random_state=1, max_iter=2000, early_stopping=True and hidden_layer_size=1000.
- **SGD** For SGD, we considered *class_weight* from balanced , None; *learning_rate* same as MLP, *loss* function from hinge, log, modified huber, squared hinge, perceptron and *penalty* from l2, l1, elasticnet. Initialization was done only with random_state=0, max_iter=5000, early_stopping=True and eta0=0.0001 .
- **GradientBoosting Classifier** For gb, we considered *learning_rate* from 0.001 to 0.1; *max_depth* from 1 to 3; *max_features* from sqrt and log2; and *n_estimators* from 100 and 500. Initialization was done only with random_state=0, and n_estimators=100.
- **Random Forest Classifier** For rf, we considered, *class_weight* same as SGD with additional option of balanced subsample;for *max_features* same as gb with additional 'auto' option; and *n_estimators* from 100 to 300. Initialization was done only with random_state=0, and max_depth=2.

6.2. DL

We used `HuggingFace` library for the transformer-based language model encoders. We ran all the deep learning models with the same hyperparameter setup. For `learning_rate` we use 1e-6, for loss function we used `CrossEntropy` loss, for optimizer we used `Adam`. We trained all the models for 100 epochs with `EarlyStopping` with `patience=5`.

- **SimpleBERT** model: We used `AutoModelForSequenceClassification` module from `HuggingFace` with a Rectified Linear Unit (ReLU) activation function.
- **BERTCNN** model: We used 5 convolutional blocks with sizes from 1 to 5, 4 input channels corresponding to the last 4 layers from language models, and 32 output channels. The output from convolution block is passed from a ReLU activation function before passing to 1-dimensional max pooling block. The max pooled output is concatenated and passed into a linear classifier layer with a dropout of 0.1. Finally, a Sigmoid activation function is applied over the logits obtained from the linear layer.

Ensemble We used a majority voting ensemble technique [34] to profit from the prediction power of different models. Further, for deep learning models, the models' predictions are later post-processed by merging them on a user-level by major voting.

		WITHOUT SMOTE				WITH SMOTE			
FEAT	MODEL	GENDER	PROF	IDEOB	IDEOM	GENDER	PROF	IDEOB	IDEOM
cv	mlp	0.60	0.83	**0.77**	0.59	0.63*	0.83'	**0.80***	0.58
glove	mlp	**0.71**	0.81	0.74	0.56	**0.72***	0.78	0.65	0.56'
laser	mlp	0.58	0.81	**0.77**	**0.62**	0.60*	0.83*	0.75	0.60
tfidf	mlp	0.67	**0.84**	0.69	0.57	0.68*	**0.84'**	0.75*	**0.62***
emoji	mlp	0.54	0.59	0.48	0.41	0.50	0.59'	0.55*	0.32
topics	mlp	0.58	0.57	0.50	0.30	0.31	0.56	0.62*	0.31
cv	nb	**0.62**	**0.89**	**0.72**	**0.60**	**0.62'**	**0.90***	**0.70**	0.59
tfidf	nb	0.58	0.88	0.67	0.58	0.60*	0.86	0.67'	**0.61***
cv	rf	**0.69**	0.65	0.58	0.54	0.46	0.68*	0.61*	0.50
emoji	rf	0.60	0.59	0.61	0.37	0.61*	0.62*	0.58	0.44*
glove	rf	0.66	**0.88**	0.61	0.37	**0.75***	**0.85**	0.59	0.41*
laser	rf	0.66	0.77	**0.72**	0.52	0.66'	0.83*	0.73*	**0.54***
tfidf	rf	0.67	0.68	0.68	**0.59**	0.74*	0.67	**0.80***	**0.54**
topics	rf	0.57	0.68	0.63	0.37	0.47	0.70*	0.62	0.34
cv	sgd	0.67	0.80	**0.72**	0.59	0.60	**0.86***	**0.74***	0.59'
emoji	sgd	0.61	0.60	0.60	0.43	0.68*	0.64*	0.55	0.38
glove	sgd	**0.75**	**0.90**	0.64	0.52	**0.70**	0.83	0.62	0.49
laser	sgd	0.62	0.84	0.70	**0.61**	0.59	0.83	0.70'	0.54
tfidf	sgd	0.66	0.86	0.67	0.57	0.68*	0.85	0.44	**0.66***
topics	sgd	0.61	0.57	0.52	0.28	0.55	0.56	0.63*	0.20
cv	svm	0.43	**0.84**	0.78	0.55	0.47*	0.82	**0.76**	0.54
emoji	svm	0.61	0.56	0.45	0.35	0.61'	0.62*	0.57*	0.32
glove	svm	**0.75**	**0.84**	0.70	0.52	0.68	0.78	0.69	0.51
laser	svm	0.67	0.81	**0.83**	**0.62**	**0.69***	**0.88***	0.72	**0.64***
tfidf	svm	0.68	**0.84**	0.73	0.58	0.64	0.84'	0.75*	0.57
topics	svm	0.60	0.57	0.62	0.21	0.64*	0.67*	0.59	0.28*
cv	gb	0.43	0.68	**0.78**	**0.61**	0.47*	0.65	**0.75**	**0.67***
emoji	gb	0.58	0.58	0.51	0.41	0.53	0.59*	0.54*	0.35
glove	gb	**0.72**	**0.77**	0.69	0.48	0.71	**0.79***	0.73*	0.53*
laser	gb	0.66	0.76	0.72	0.59	**0.73***	0.88*	0.71	0.55
tfidf	gb	0.64	0.65	0.66	**0.61**	0.71*	0.73*	0.75*	0.60
topics	gb	0.65	0.70	0.63	0.28	0.47	0.67	0.67*	0.21

Table 3
ML weighted F1-scores on Dev-Test; * refers to increased and ' refers to unchanged weighted F1-score compared to WITHOUT SMOTE setting

Metrics For all the experiments, we considered weighted F1-score as the metric for evaluation because of the imbalance present in the dataset. For the final leaderboard, the organizers used macro F1-score for the overall performance ranking of the systems.

7. Results

Before conducting the machine learning experiments, we analyzed whether the combination of the text with emoji features has any beneficial effect on the classification tasks. We have

		WITHOUT DATA AUG				WITH DATA AUG			
MODEL	LM	GENDER	PROF	IDEOB	IDEOM	GENDER	PROF	IDEOB	IDEOM
BERTSimple	mbert	0.5469	0.7853	0.6669	0.4854	0.5290	0.7854*	0.6457	0.4617
BERTSimple	beto	**0.5630**	**0.7950**	**0.7002**	**0.4887**	0.5418	0.7977 *	**0.6943**	**0.4900***
BERTSimple	xlmrb	0.5582	0.7905	0.6713	0.3382	0.5436	0.8006*	0.6806*	0.3787*
BERTCNN	beto	0.3408	0.4308	0.2912	0.0987	0.5449*	0.5679*	0.5331*	0.2432*

Table 4
DL weighted F1-scores on Dev-Test; * refers to increased weighted F1-score compared to WITHOUT DATA AUG setting

Rank	Team	Overall-f1	Gender-f1	Profession-f1	Ideology-B	Ideology-M
1	LosCalis	0.9022	0.9028 (1)	0.9443 (1)	0.9616 (1)	0.8002 (4)
2	NLP-CIMAT-GTO	0.8909	0.7848 (6)	0.9212 (3)	0.9614 (2)	0.8962 (1)
3	Alejandro Mosquera	0.8891	0.8267 (3)	0.9334 (2)	0.9515 (3)	0.8450 (3)
4	CIMAT_2021	0.8797	0.8368 (2)	0.8950 (5)	0.9416 (4)	0.8455 (2)
5	**Our System**	0.8253	0.7260 (13)	0.8977 (4)	0.9217 (5)	0.7557 (6)
20	Baseline	0.5112	0.5762 (19)	0.4324 (18)	0.5956 (19)	0.4406 (19)

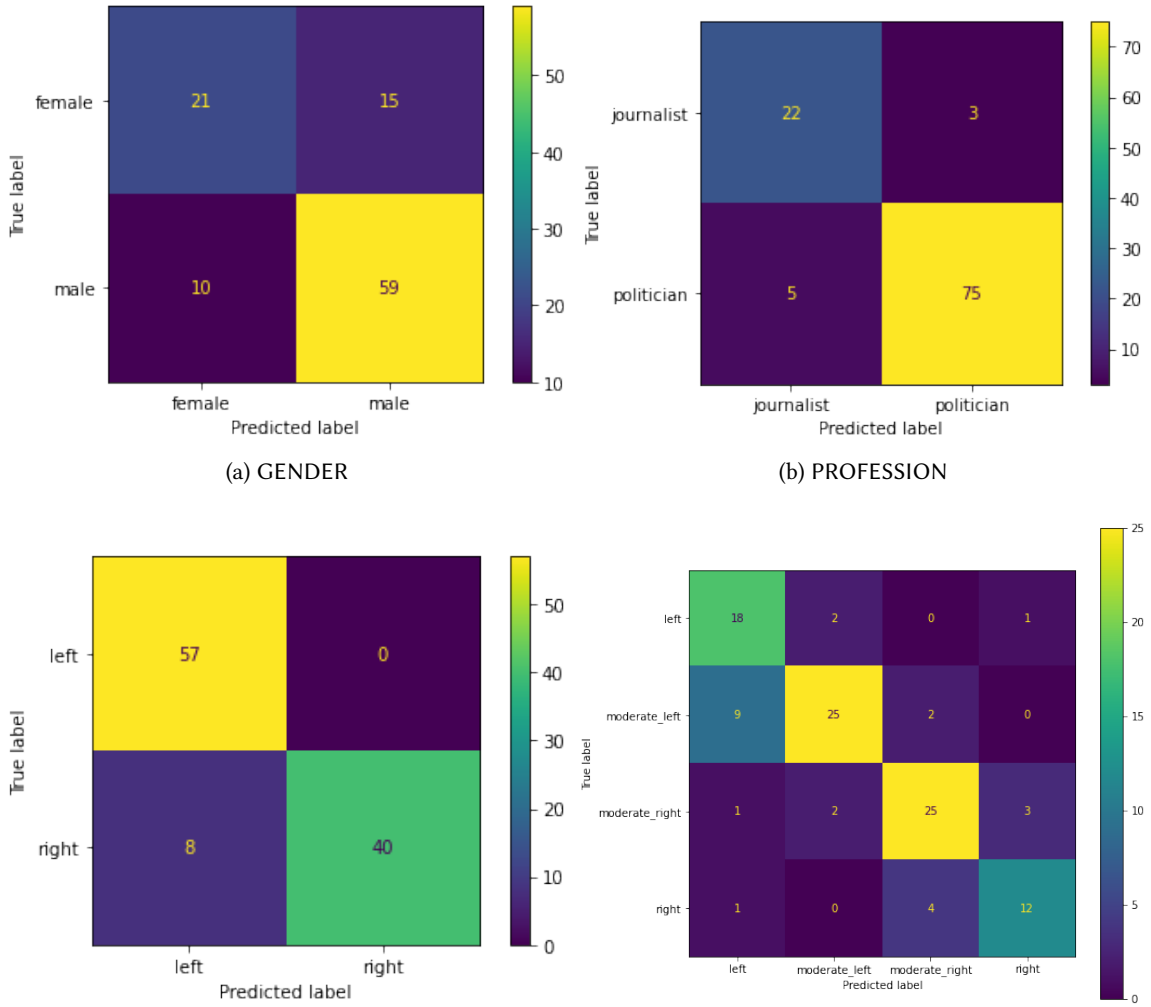
Table 5
Final Leaderboard results corresponding to macro F1-score on blind-test set

considered this analysis given the use of emojis is an essential part of the language used on tweets. For this analysis, we used an SVM classifier and considered two settings: text features without emoji and text with concatenated emoji features (Table 6). The text features were generated on the preprocessed data.

For machine learning experiments, we present all the analyses in table 3. This table contains two sets of results; with and without data augmentation (SMOTE). Each value corresponds to the weighted F1-score obtained by an exhaustive hyperparameter search done for every feature and the hyperparameters associated with the classifier model (refer Section 6.1).

For deep learning experiments, we present all the analyses in table 4. It presents two sets of results corresponding with and without data augmentation. We experimented with three language models (refer 6.2) and further, we continued our experiments with BETO based on its overall performance across the different subtasks.
For the final submission on the leaderboard (Table 5), we submitted the following:

- For GENDER, we submitted an ensemble of the best 5 ML models trained with SMOTE technique based on the overall weighted F1-score. The models were : `laser-svm`, `tfidf-mlp`, `laser-gb`, `cv-mlp` and `cv-nb`.
- For PROFESSION, we submitted the predictions from BERTSimple with BETO encoder trained without data augmentation. Even though training and evaluation were done on a tweet level as mentioned in section 6, we further merged the predictions on user-level choosing the dominant label for the submission.
- For IDEOLOGY-BINARY, we submitted an ensemble of the predictions from `cv-mlp` model setup and BERTSimple with BETO trained without data augmentation from deep learning model setup.



(a) GENDER

(b) PROFESSION

(c) IDEOLOGY-BINARY

(d) IDEOLOGY-MULTICLASS

Figure 1: Confusion matrix visualization of our leaderboard submission

- For IDEOLOGY-MULTICLASS, we submitted the predictions from `cv-gb` model trained with SMOTE technique.

8. Discussion

We observed that classic ML algorithms perform well with GloVe, LASER embeddings, and term frequency based features. We found out that for GENDER classification, GloVE embeddings obtain the best weighted F1-scores. This could be attributed to its capability to capture co-occurrence information of words. SMOTE technique provides no significant improvement in the performance. We found that GloVE embeddings features performed similar for PROFESSION

	WITHOUT EMOJI				WITH EMOJI			
FEAT	GENDER	PROF	IDEOB	IDEOM	GENDER	PROF	IDEOB	IDEOM
tfidf	0.59	0.83	0.70	0.72	0.60*	0.78	0.72*	0.64
cv	0.51	0.62	0.68	0.62	0.51	0.62	0.68	0.62
laser	0.66	0.79	0.75	0.53	0.64	0.68	0.65	0.41
glove	0.75	0.76	0.66	0.46	0.68	0.80*	0.67*	0.46
average	0.63	0.75	0.70	0.58	0.60	0.72	0.68	0.53

Table 6
Effect of combination of emoji and text features ; * refers to increase in weighted F1-score compared to WITHOUT EMOJI setting

classification, but term based ones such as CV and TF-IDF obtained comparable results. SMOTE technique had no substantial improvement over the best results obtained by the models trained without smote. However, it does improve the confusion matrix (see fig 1) providing better classification models. For the IDEOLOGY-BINARY classification task, we found CV and LASER to be overall the most effective features. SMOTE technique had little effect and was found not to be very helpful in improving the performance. A consistent performance was observed with CV, TF-IDF and LASER features for IDEOLOGY-MULTICLASS classification. For this class, using the SMOTE technique outperformed the highest score that we obtain without using SMOTE. Overall, we found that using topic and emoji features have not provided any insight on the four tasks. In an additional investigation, we found out that machine learning models trained with concatenation of emoji and text features do not perform better. On the contrary, it tends to reduce its weighted F1-score (refer Table 6).

For the deep learning experiments, we found `BETO` to be overall the best among the other choices for language models. Contrary to `mbert` and `xlmrb` that are trained on multilingual corpora, `BETO` is only trained on Spanish corpora. This might be a reason for its better performance over the others. We observed that `BERTSimple` with `BETO` performed best for every subtask. Text Augmentation had a low effect on `BERTSimple`, but it significantly increased the model score of BERTCNN.

Regarding the submission results, our system did not perform well on the gender classification. This can be attributed to the fact that the model seemed biased towards the male label as we can observe on the confusion matrix (see fig. 1). Also, we suspect that there may be an effect in the preprocessing stage, due to the gender neutralization caused by the lemmatization. Furthermore, our system performed well for the PROFESSION and IDEOLOGY-BINARY classification tasks besides the data imbalance. Specially on the profession class where is more noticeable (refer Table 2). Regarding the IDEOLOGY-MULTICLASS class, our system performed well over the four labels. However, the weighted F1-score was lower compared to the binary class. We observe in the confusion matrix that the model underperforms at discriminating between multiple labels on a political spectrum . Thus, while it captures the correct political spectrum regardless of the label, it shows problems at capturing more grained information which could be due to the lack of enough data samples for each label (refer Table 2).

9. Conclusion

This paper provided an analysis of a broad experimental setup where we compared the use of different machine and deep learning approaches for the IberLEF 2022 shared task. While deep learning methods have been widely reported to be effective on text classification tasks, we found that GloVE embedding features and term-frequency based features like TF-IDF can be very helpful and can provide comparative results to deep learning approaches. In future work, we plan to investigate the quality of combination of different features. Transfer learning has shown to be promising for text classification [14]. It would be therefore interesting to train models with similar domain corpora which may include an hybrid of oral-written and formal-informal language use due to the communicative setting of tweets and the lack of language models trained on Spanish political tweets. Finally, exploring multitask learning for ideology classification could be an effective strategy to capture more grain-level nuances.

References

[1] M. Debus, J. Tosun, Political ideology and vaccination willingness: implications for policy design, Policy Sciences 54 (2021). doi:10.1007/s11077-021-09428-0.

[2] S. Gadarian, S. Goodman, T. Pepinsky, Partisanship, health behavior, and policy attitudes in the early stages of the covid-19 pandemic, SSRN Electronic Journal (2020). doi:10.2139/ssrn.3562796.

[3] M. Fatke, Personality traits and political ideology: A first global assessment, Political Psychology 38 (2017) 881–899. URL: http://www.jstor.org/stable/45095184.

[4] J. A. García-Díaz, S. M. Jiménez-Zafra, M. T. Martín-Valdivia, F. García-Sánchez, L. A. Ureña-López, R. Valencia-García, Overview of PoliticES 2022: Spanish Author Profiling for Political Ideology, Procesamiento del Lenguaje Natural 69 (2022).

[5] F. Rangel, A. Giachanou, B. H. H. Ghanem, P. Rosso, Overview of the 8th author profiling task at pan 2020: Profiling fake news spreaders on twitter, in: CEUR Workshop Proceedings, volume 2696, Sun SITE Central Europe, 2020, pp. 1–18.

[6] F. Rangel, G. Sarracén, B. Chulvi, E. Fersini, P. Rosso, Profiling hate speech spreaders on twitter task at pan 2021, in: CLEF, 2021.

[7] 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.

[8] F. Rangel, P. Rosso, Overview of the 7th author profiling task at pan 2019: bots and gender profiling in twitter, in: Working Notes Papers of the CLEF 2019 Evaluation Labs Volume 2380 of CEUR Workshop, 2019.

[9] U. Sapkota, T. Solorio, M. Montes-y-Gómez, G. Ramírez-de-la-Rosa, Author profiling for english and spanish text notebook for PAN at CLEF 2013 1179 (2013). URL: http://ceur-ws.org/Vol-1179/CLEF2013wn-PAN-SapkotaEt2013.pdf.

[10] A. Poulston, M. Stevenson, K. Bontcheva, Topic models and n–gram language models for author profiling, in: Proceedings of CLEF, 2015.
[11] F. Vitiugin, G. Barnabo, Emotion detection for spanish by combining laser embeddings, topic information, and offense features, in: Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2021). CEUR Workshop Proceedings, CEUR-WS, Málaga, Spain, 2021.
[12] A. F. Sotelo, H. Gómez-Adorno, O. Esquivel-Flores, G. Bel-Enguix, Gender identification in social media using transfer learning, in: Mexican Conference on Pattern Recognition, Springer, 2020, pp. 293–303.
[13] E. Alzahrani, L. Jololian, How different text-preprocessing techniques using the BERT model affect the gender profiling of authors, CoRR abs/2109.13890 (2021). URL: https://arxiv.org/abs/2109.13890. arXiv:2109.13890.
[14] P. L. Úbeda, M. C. Díaz-Galiano, L. A. U. López, M. T. Martín-Valdivia, T. Martín-Noguerol, A. Luna, Transfer learning applied to text classification in spanish radiological reports, in: Proceedings of the LREC 2020 Workshop on Multilingual Biomedical Text Processing (MultilingualBIO 2020), 2020, pp. 29–32.
[15] J. A. García-Díaz, R. Colomo-Palacios, R. Valencia-García, Psychographic traits identification based on political ideology: An author analysis study on Spanish politicians' tweets posted in 2020, Future Generation Computer Systems 130 (2022) 59–74.
[16] P. Qi, Y. Zhang, Y. Zhang, J. Bolton, C. D. Manning, Stanza: A Python natural language processing toolkit for many human languages, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, 2020. URL: https://nlp.stanford.edu/pubs/qi2020stanza.pdf.
[17] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, Smote: synthetic minority over-sampling technique, Journal of artificial intelligence research 16 (2002) 321–357.
[18] E. Ma, Nlp augmentation, https://github.com/makcedward/nlpaug, 2019.
[19] J. Pennington, R. Socher, C. D. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014, pp. 1532–1543.
[20] Z. Chen, X. Lu, S. Shen, W. Ai, X. Liu, Q. Mei, Through a gender lens: An empirical study of emoji usage over large-scale android users, CoRR abs/1705.05546 (2017). URL: http://arxiv.org/abs/1705.05546. arXiv:1705.05546.
[21] B. Eisner, T. Rocktäschel, I. Augenstein, M. Bosnjak, S. Riedel, emoji2vec: Learning emoji representations from their description (2016) 48–54. URL: https://doi.org/10.18653/v1/W16-6208. doi:10.18653/v1/W16-6208.
[22] C. J. C. Burges, A tutorial on support vector machines for pattern recognition, Data Min. Knowl. Discov. 2 (1998) 121–167. URL: https://doi.org/10.1023/A:1009715923555. doi:10.1023/A:1009715923555.
[23] H. Zhang, The optimality of naive bayes, Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference, FLAIRS 2004 2 (2004) 562–567.
[24] L. Breiman, Random forests, Mach. Learn. 45 (2001) 5–32. URL: https://doi.org/10.1023/A:1010933404324. doi:10.1023/A:1010933404324.
[25] J. H. Friedman, Greedy function approximation: A gradient boosting machine, The Annals of Statistics 29 (2001) 1189–1232. URL: http://www.jstor.org/stable/2699986.

[26] F. Rosenblatt, Principles of neurodynamics. perceptrons and the theory of brain mechanisms, Technical Report, Cornell Aeronautical Lab Inc Buffalo NY, 1961.
[27] S. Ruder, An overview of gradient descent optimization algorithms, CoRR abs/1609.04747 (2016). URL: http://arxiv.org/abs/1609.04747. arXiv:1609.04747.
[28] Y. Kim, Convolutional neural networks for sentence classification (2014) 1746–1751. URL: https://doi.org/10.3115/v1/d14-1181. doi:10.3115/v1/d14-1181.
[29] J. Cañete, G. Chaperon, R. Fuentes, J.-H. Ho, H. Kang, J. Pérez, Spanish pre-trained bert model and evaluation data, in: PML4DC at ICLR 2020, 2020.
[30] J. Tiedemann, Parallel data, tools and interfaces in OPUS, in: N. Calzolari, K. Choukri, T. Declerck, M. U. Dogan, B. Maegaard, J. Mariani, J. Odijk, S. Piperidis (Eds.), Proceedings of the Eighth International Conference on Language Resources and Evaluation, LREC 2012, Istanbul, Turkey, May 23-25, 2012, European Language Resources Association (ELRA), 2012, pp. 2214–2218. URL: http://www.lrec-conf.org/proceedings/lrec2012/summaries/463.html.
[31] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding (2019) 4171–4186. URL: https://doi.org/10.18653/v1/n19-1423. doi:10.18653/v1/n19-1423.
[32] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov, Unsupervised cross-lingual representation learning at scale, CoRR abs/1911.02116 (2019). URL: http://arxiv.org/abs/1911.02116. arXiv:1911.02116.
[33] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized BERT pretraining approach, CoRR abs/1907.11692 (2019). URL: http://arxiv.org/abs/1907.11692. arXiv:1907.11692.
[34] L. I. Kuncheva, C. J. Whitaker, Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy, Mach. Learn. 51 (2003) 181–207. URL: https://doi.org/10.1023/A:1022859003006. doi:10.1023/A:1022859003006.