INFOTEC-LaBD at PoliticES 2022: Low-dimensional Stacking Model for Political Ideology Profiling

Hiram Cabrera, Eric Sadit Téllez and Sabino Miranda

INFOTEC Centro de Investigación e Innovación en Tecnologías de la Información y Comunicación, Circuito Tecnopolo Sur No. 112, Fracc. Tecnopolo Pocitos Aguascalientes, Ags., México

Abstract

This manuscript describes the low-dimensional stacking model approach used to profile users to solve gender and profession identification and binary and multiclass ideology prediction tasks. We developed these models in the scope of the *PoliticEs: Spanish Author Profiling for Political Ideology* carried out at *IberLEF@SEPLN 2022*. Our methodology stacks several low-dimensional representations that can be used to visualize the dataset and as the input dataset for a classifier. While the results were late in the challenge, our final evaluations achieved high performances in the training and test partitions. We believe they are promising approaches on the road to creating transparent and competitive user profiling models.

Keywords

author profiling political ideology, author profiling visualization, low-dimensional model stacking

1. Introduction

The task of modeling users employing their published messages in a social network is known as author profiling. The profiling model can be used to determine similarities and compute groups that help to know better characteristics like social demographic markers (gender, age range, origin, education level), religion, and political parties, among others, see [1]. With some associated error, user profiling models can predict these traits for never-seen individuals and groups.

While it is crucial to develop models with low associated errors, it is also desirable that a profiling model can help understand the knowledge dataset, i.e., human-labeled data, and why a model makes some decisions. In this sense, explainability is a major goal of a model.

1.1. Related work

Author profiling has been popular because of the nature of social networks and its applications in forensics (language as evidence), security, and marketing, for instance, to know the demographics of people that like or dislike their products [2]. In this sense, several competitions have been run contests for author profiling tasks, such as a series of PAN@CLEF and FIRE [2, 1, 3, 4, 5]. These forums address different problems such as age, gender, language variety identification, personality

IberLEF 2022, September 2022, A Coruña, Spain.

hiram.cabrerap@infotec.mx (H. Cabrera); eric.tellez@infotec.mx (E. S. Tellez); sabino.miranda@infotec.mx (S. Miranda)

0000-0003-1419-761X (H. Cabrera); 0000-0001-5804-9868 (E. S. Tellez); 0000-0002-0595-2401 (S. Miranda)

recognition in several languages and genres, including blogs, reviews, social media, and Twitter, among others; MEX-A3T (type of occupation and place of residence) [6], IberLEF@SEPLN 2022 (gender, profession, and political ideology) [7], and others.

Various methods have been proposed, typically using features based on content and style like part-of-Speech, stopwords, punctuation marks frequency, and length of the used words, among others. For example, Benzel [8] represents an author style as a vector of word sequences and their frequencies; word frequencies analysis in political speeches identifies terms that evidence an author's style [9]. Others use a deeper analysis considering syntactic-based [10], discourse-based [11], and topic-based [12] features. In text classification tasks, high-dimensional data may raise issues for the performance of systems. Representing data in low-dimensional spaces can reduce data complexity while maintaining similar information. In this sense, LSA-based approaches [13] are applied, finding the best parameters based on character n-gram lengths and local and global weighting schemes. In [14], documents are represented using stylistic-discriminative and thematic-descriptive high-level features through dimensionality reduction applying LSA.

1.2. The PoliticES task at IberLEF@SEPLN 2022

The PoliticES training corpus consist of 37,560 tweets in Spanish from 312 authors ($\sim$ 120 messages per author) [15], where each author is labeled with its *profession* (journalist or politician), *gender* (male or female), *binary political ideology* (left-right) and *multiclass political ideology* (left, moderate-left, moderate-right, right).

The training dataset is highly imbalanced for the *profession* class; there is 80.3% vs. 19.7% between politicians and journalists. The proportion is 43.5% vs. 56.5% regarding *gender*, i.e., a slight imbalance between female and male categories, and the same applies for *binary ideology* with 56.9% vs. 43.1% for left and right, respectively. There are 24.3%, 32.6%, 30%, and 13.1% between left, moderate left, moderate right, and right regarding *multiclass political ideology*.

We are asked to create models to predict these labels for a test dataset, i.e., a list of 150 users (120 messages per user); the label distribution of the test dataset is unknown.

1.3. Overview

Our manuscript is organized as follows. The current section is dedicated to introducing the task and briefly reviewing the state-of-the-art. Section 2 introduces our approach for representing, visualizing, and predicting users' political traits based on their messages written on social media. The experimental results are presented and discussed in Section 3. The manuscript is summarized and concluded in Section 4. We also added an appendix section with visualizations of the subtasks.

2. Our approach

Our approach has three main modules: i) the vector space model, ii) non-linear dimensional reduction and iii) the supervised learning stages.

Firstly, the vector space is created through preprocessing, tokenizing and weighting terms to obtain a vector space model based on entropy weighting; this vector space has very high

dimensionality. The next module uses the non-linear method *Uniform Manifold Approximation and Projection for Dimension Reduction* (UMAP) [16] to produce a low dimensional vector space (UMAP model). We produce three-dimensional projections to visualize the datasets but also, due to their low dimension, we use these projections as the input of (non-linear) classifiers (third module).

The rest of this section details our approach, mainly the training stage.

2.1. Module 1: preprocessing and vectorization

The vector space model is created using a bag of words. On the one hand, we normalize and apply several preprocessing functions. More detailed, messages were lowercased, blank spaces were normalized to a single space, and diacritic marks were removed. Token numbers 1-9 were preserved to capture important information on small numbers, while other numbers were replaced by 0 (to reduce dimensionality). Secondly, users and URLs were replaced by a special token, i.e., `_usr` and `_url`, respectively. In particular, we also normalize several kinds of big laughs to one of the following forms `jaja`, `jiji`, `jeje`, mostly based on how these expressions were written. Punctuation symbols were kept.

Regarding tokenization, we mix three kinds of tokens: unigrams, bigrams, and character q-grams of size four. The tokens were preserved in our vocabulary V if they appeared at least ten times in the training dataset. This procedure outputs a vocabulary of 764k unique tokens for the IberLEF 2022 *PoliticEs* task.

2.1.1. Weighting tokens

We use a global weighting scheme based on the entropy of the empirical distribution of each term on the classes [17]. The weighting expression for the token t is as follows:

$$\mathsf{weight}(t) = 1 - \frac{1}{\log \#\mathsf{classes}} \sum_{c \in \mathsf{classes}} p_{t,c} \log \frac{1}{p_{t,c}},$$

where $p_{t,c}$ is the probability of token t in class c estimated based on users as described in §2.1.2. The empirical distribution is estimated from the train set. Each weight value varies from 0 to 1. Weights close to zero mean that its entropy is high, i.e., the token has no discrimination power; weights close to 1 are highly discriminating tokens.

2.1.2. User modeling

We represent each user as its collection of messages; we applied our preprocessing step and tokenization to these messages to create a large bag of words. This bag of words is weighted to create a high dimensional vector where each component corresponds to a token entry t in the vocabulary V. The component corresponding to t is valued as $\mathsf{weight}(t)$ or zero, not represented if the token t does not appear in that particular user message. Each user is then represented by a large sparse vector normalized to have a unitary norm.

2.2. Module 2: Non-linear dimensional reduction

Our initial vector space has a very high dimensionality; this issue imposes problems for classifiers that work on each component, like decision trees or neural networks. Other methods workings on kernel functions are limited to linear approaches since the number of components significantly increases the training cost. Additionally, the large vocabularies also degrade the explainability obtained by bag-of-words approaches.

Therefore, we created low-dimensional projections with two objects in mind. On the one hand, if the projection dimension is two or three, we provide a simple way of visualizing a dataset and its labels. A latent cluster structure can also arise. Modern visualization tools can help discover the properties of each group with minor effort. On the other hand, the lower-dimensional vector database can be approached by component-based classifiers, or in general, with non-linear models due to its lower cost.

As commented above, we use the UMAP non-linear dimensional reduction method, which receives a graph of all k nearest neighbors in a collection and performs the projection trying to preserve the topology of the input graph. The graph is created on vector databases described in Module 1, which also uses the cosine between vector angles as a similarity function.

The number of neighbors k take values between one and the number of users minus one; few neighbors indicate that we are interested in the dataset's local structure and high values on the global one. We use $k = 40$ to preserve both the local and the global structure partially. We fixed the UMAP method to create three-dimensional embeddings using spectral layout for embedding initialization and then optimized with 100 epochs and three negative samples per point (user vector).

2.3. Module 3: Supervised approach

We computed a UMAP model for each subtask (i.e., gender, profession, binary and multiclass ideology). Each model produces a three-dimensional embedding of the dataset; we concatenate each embedding such that a 12-dimensional vector represents each user. Each vector was standardized before the actual concatenation (normalized to have zero mean and unit variance) to reduce scaling problems. This 12-dimension vector is our stacked final representation that can be used as input for any classifier.

Now that our representation is defined, we can observe that the dimension is quite low, and therefore it is possible to create complex models with moderate time costs. We can also afford to optimize the model's hyper-parameters relatively fast. For this objective, we used the Grid Search model selection with a k-folds cross-validation (using five stratified folds and three repetitions) to evaluate the performance of the best model optimizing for the macro F1 score. In particular, we used the `GridSearchCV` sklearn's class for this procedure.

We consider SVM classifiers with linear and non-linear kernels and the GradientBoosting classifier (decision-tree ensemble with boosting training) as classifiers, available in the sklearn library [18]. We perform a model selection procedure for tuning each classifier. Note that even when we use the same 12-dimensional vectors as input, we produce a model for each subtask (different labels).

2.4. Prediction stage

The prediction stage applies similar steps to training but uses previously learned models. For instance, it uses the vocabulary previously created to vectorize the test dataset and the already learned UMAP model to project this testing database instead of learning a new model. For each task, the classifier is then asked to predict with the 12-dimensional vectors produced in previous modules (vector standardization should be performed using the median and variance of the training set).

3. Experimental results

This section presents the experimental results of our approach for the IberLEF 2022 *PoliticEs* task. Source code is available at https://github.com/hiramcp/PoliticEs2022.

We perform a Grid Search model selection using five-fold stratified cross-validation, maximizing macro-F1 score as objective function (averaged on three repetitions). We consider Gradient Boosting-GB, RBF SVM, and Linear SVM classifiers. We evaluated two distinct user embedding representations regarding embeddings: a same-task 3-dimension vector or an all-tasks stacked embedding of 12-dimensions. Finally, we also compare the effect of using standardized embeddings.

We ran twelve different experiments for different combinations of classifiers, user embedding representations, and vector standardization to perform the model evaluation for each subtask as described in §2.3. The results are shown in Table 1.

Classifier	Dim.	Standardized	Model sel. time	F1 Gender	F1 Profession	F1 Ideology Binary	F1 Ideology Multiclass	Avg. Macro F1
GB	12-dim.	No	4621s	0.987±0.012	0.992±0.012	0.997±0.007	0.957 ± 0.03	0.9832 (4)
		Yes	5711s	0.987±0.012	0.993±0.011	0.997±0.007	0.958 ± 0.02	**0.9837 (2)**
	3-dim.	No	3121s	0.987±0.012	0.993±0.011	0.997±0.007	0.953±0.016	0.9826 (6)
		Yes	6429s	0.987±0.012	0.992±0.012	0.997±0.007	0.954±0.016	0.9825 (7)
SVM RBF	12-dim.	No	47s	0.99 ± 0.008	0.992±0.012	0.997±0.007	0.956±0.026	**0.9838 (1)**
		Yes	10s	0.99 ± 0.008	0.990±0.012	0.997±0.007	0.95 ± 0.028	0.9817 (8)
	3-dim.	No	9s	0.988±0.009	0.990±0.012	0.997±0.007	0.957 ± 0.02	0.9830 (5)
		Yes	8s	0.99 ± 0.008	0.990±0.012	0.997±0.007	0.957 ± 0.02	0.9835 (3)
SVM Linear	12-dim.	No	35s	0.99 ± 0.008	0.990±0.012	0.997±0.007	0.947±0.031	0.9810 (10)
		Yes	32s	0.988±0.011	0.990±0.012	0.997±0.007	0.951±0.028	**0.9815 (9)**
	3-dim.	No	31s	0.99 ± 0.008	0.990±0.012	0.997±0.007	0.938±0.028	0.9787 (12)
		Yes	32s	0.99 ± 0.008	0.990±0.012	0.997±0.007	0.944±0.024	0.9802 (11)

Table 1: Performance for the machine learning models using the cross-validation partitions; the higher, the better. Scores are presented as their average and standard deviation (using all folds for these statistics). Best scores per classifier are in bold and ranked inside the parenthesis.

Please observe how almost configurations perform the same for binary tasks, i.e., *gender*, *profession*, and *ideology*; the multiclass *ideology* is more diverse in this sense. When we observe the macro averaged F1 score, we can also observe that all classifiers achieve scores beyond 0.97 in some configurations and that RBF SVM achieves best performing configurations. We

can also highlight the effect of the 12-dimensional stacked embeddings and the effect of the standardization on the score. The Linear SVM also achieves high performance; however, it has a lower performance regarding multiclass ideology, but it remains competitive and fast. We need to clarify that the linear model has a slower model selection just because we consider more hyperparameters than the RBF motivated by this speed. However, the Gradient Boosting classifier performs exceptionally well with a high cost in the model selection stage. Although training time varies according to the dataset and task domain, we can observe that our stacked final representation allows us to create complex models with moderate time costs. The initial text vectorization and the UMAP model require less than ten seconds and are static across evaluations.

Subtask	**Model**	**Hyperparameters**
Gender	Gradient Boosting	learning rate: 0.01, max depth: 3, estimators: 1000, subsample: 0.5
	Linear SVM	C: 1, dual: false, max iter: 5000, penalty: l1, random state: 0, tol: 2
	RBF SVM	C: 10, gamma: 0.01
Profession	Gradient Boosting	learning rate: 0.1, max depth: 3, estimators: 100, subsample: 0.7
	Linear SVM	C: 0.1, dual: false, max iter: 5000, penalty: l1, random state: 0, tol: 2
	RBF SVM	C: 1, gamma: 0.1
Binary Ideology	Gradient Boosting	learning rate: 0.001, max depth: 3, estimators: 1000, subsample: 0.5
	Linear SVM	C: 0.0005, dual: false, max iter: 5000, penalty: l2, random state: 0, tol: 2
	RBF SVM	C: 0.1, gamma: 0.01
Multiclass Ideology	Gradient Boosting	learning rate: 0.1, max depth: 9, estimators: 1000, subsample: 0.5
	Linear SVM	C: 1, dual: false, max iter: 5000, penalty: l1, random state: 42, tol: 0.4
	RBF SVM	C: 1000, gamma: 0.001

Table 2: The best hyperparameters for the machine learning models for each subtask.

We chose the parameter combination in each subtask that had the highest macro F1 score during the cross-validation for the final selection. Thus, the final models were fitted on the entire training set. The best hyperparameters are summarized in Table 2.

Table 3 lists the summary of final scores for each experiment using the supplied test corpus. While the results were late in the challenge, our final evaluation achieved high performance with the test partition. Note that our stacking approach produces the highest performance configurations using standardization of embeddings. Linear SVM generalizes the better, but

suffers on *profession* on 3-dim vectors. Note that RBF achieves a better macro-F1 score on 12-dim and standardized vectors. In the rest of this section, we will show some of the internal functionality of our approach.

Classifier	Embedding Type	Standardized Vector	F1 Gender	F1 Profession	F1 Ideology Binary	F1 Ideology Multiclass	Avg. Macro F1
GB	12-dim.	No	0.7127	0.6111	0.9515	0.6216	0.7242 (9)
		Yes	0.7260	0.8333	0.9515	0.7162	0.8067 (5)
	3-dim.	No	0.7260	0.8333	0.9515	0.6029	0.7784 (7)
		Yes	0.7260	0.8333	0.9515	0.7299	**0.8102 (4)**
RBF SVM	12-dim.	No	0.7214	0.6026	0.9515	0.6259	0.7254 (8)
		Yes	0.7463	0.8333	0.9613	0.8306	**0.8429 (1)**
	3-dim.	No	0.6992	0.4324	0.6546	0.7566	0.6357 (12)
		Yes	0.7552	0.4324	0.9416	0.7605	0.7224 (10)
Linear SVM	12-dim.	No	0.7390	0.8333	0.9515	0.6580	0.7954 (6)
		Yes	0.7428	0.8391	0.9613	0.7802	**0.8308 (2)**
	3-dim.	No	0.7517	0.8333	0.9515	0.7319	0.8171 (3)
		Yes	0.7517	0.1923	0.9229	0.6953	0.6406 (11)

Table 3: Final results achieved with the test gold dataset. The higher, the better. Best scores per classifier are in bold and ranked globally inside the parenthesis.

3.1. Analysis of low-dimensional stacking models

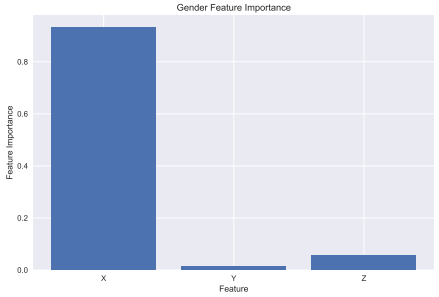
The results in Table 3 show that either the three-dimensional embedding (one model for each subtask) or the 12-dimensional stacked representation build competitive machine learning models in a timely fashion (see Table 1 for details).

Gradient boosting is a kind of ensemble of decision trees that access attributes directly instead of accessing data through a kernel function. An essential feature of GB is that it naturally measures the *attribute importance* useful to know the contribution of each attribute in the machine learning model. The best GB model was built with a three-dimensional standard user vector at a relatively low computational cost. In Figure 1 we can see that a single dimension is the main contributor to each predictive model in three of four subtasks.

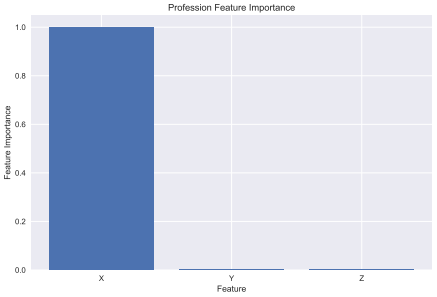
Linear Support Vector Machine creates a hyperplane that separates data into classes as best as possible. Figures 1e, 1f, 1g, and 1h expose the coefficients of the linear model (weights); it is possible to interpret them as an attractor to some side of the hyperplane and low weights as a reduction of the importance of the attribute. We can observe that weights are higher for those components related to the subtask being tackled. Also, most attributes have a weight different from zero, contributing to the entire decision.

An SVM model achieved the best performance with an RBF kernel that uses radial basis functions to separate data by hyper-spheres quickly. This model was built with a 12-dimension standardized user vector.

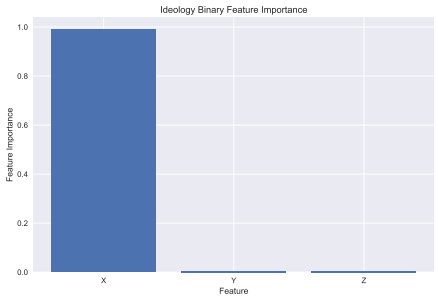
Appendix A illustrates the visualizations of both the projected train and test data, suggesting that the emerging data groups contribute to building models in a timely manner.



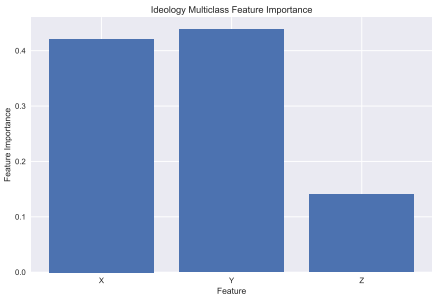
(a) Gender



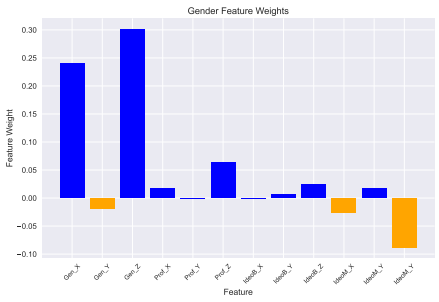
(b) Profession



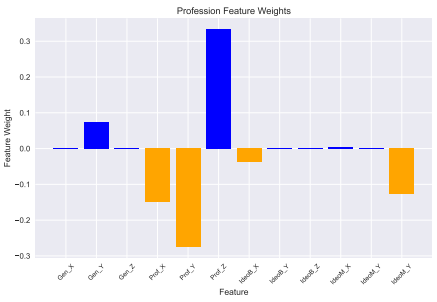
(c) Binary Ideology



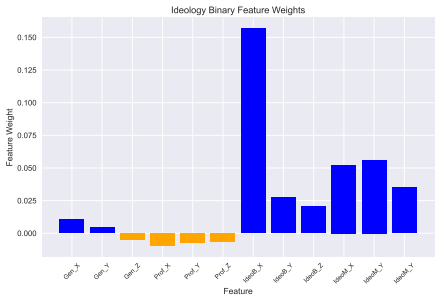
(d) Multiclass Ideology



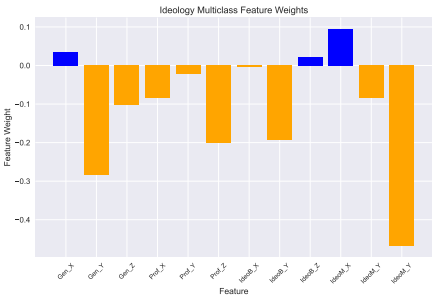
(e) Gender



(f) Profession



(g) Binary Ideology



(h) Multiclass Ideology

Figure 1: Feature importance and weights for the 3-dimensional Gradient boosting models and the 12-dimensional (stacked representation) Linear SVM models.

3.2. Experimental setup

Our experiments were run in a four core Laptop with 32 GB of RAM, more detailed, a Intel Core i7-1165G7 @ 2.80GHz using the Windows 10 operating system. We compute the vector space using the `TextSearch.jl` Julia package,[1] which also implements all preprocessing functions, tokenizers, and the entropy-based weighting scheme. The UMAP projections were computed with the `SimSearchManifoldLearning.jl` Julia package.[2] The model selection and classification was performed with the Python scikit-learn package [18].

4. Conclusions

This paper proposes Low-dimensional Stacking Model to tackle the Political Ideology Profiling challenge at IberLEF@SEPLN 2022. Our approach was designed to create both transparent and competitive user profiling models.

Due to confusion about deadlines, most of our results were not registered on the final leaderboard. The average Macro F1 score reported was 0.7242 and ranked in the fifteenth position. While the results were late in the challenge, our final evaluations achieved high performances in the test partition, using the same evaluation system, with an average Macro F1 score of 0.8429 that could be ranked the fifth place on the competition's scoring chart.

Additionally, our approach achieves a competitive trade-off between performance, explainability, and speed thanks to the low dimensional representation. We used this characteristic to apply the model selection procedure to improve our predictions and learn how models use the available features. Practitioners can use these characteristics to understand more about the studied problem.

References

[1] F. M. R. Pardo, P. Rosso, M. M. y Gómez, M. Potthast, B. Stein, Overview of the 6th author profiling task at pan 2018: Multimodal gender identification in twitter, in: CLEF, 2018.

[2] E. Stamatatos, M. Potthast, F. Rangel, P. Rosso, B. Stein, Overview of the pan/clef 2015 evaluation lab, in: Proceedings of the 6th International Conference on Experimental IR Meets Multilinguality, Multimodality, and Interaction - Volume 9283, CLEF'15, Springer-Verlag, Berlin, Heidelberg, 2015, p. 518–538. URL: https://doi.org/10.1007/978-3-319-24027-5_49. doi:10.1007/978-3-319-24027-5_49.

[3] P. Rosso, F. Rangel, Author profiling tracks at fire, SN Computer Science 1 (2020) 72. URL: https://doi.org/10.1007/s42979-020-0073-1. doi:10.1007/s42979-020-0073-1.

[4] J. Bevendorff, B. Chulvi, G. L. D. L. P. Sarracén, M. Kestemont, E. Manjavacas, I. Markov, M. Mayerl, M. Potthast, F. Rangel, P. Rosso, E. Stamatatos, B. Stein, M. Wiegmann, M. Wolska, , E. Zangerle, Overview of PAN 2021: Authorship Verification,Profiling Hate Speech Spreaders on Twitter,and Style Change Detection, in: 12th International Conference of the CLEF Association (CLEF 2021), Springer, 2021.

[1] https://github.com/sadit/TextSearch.jl

[2] https://github.com/sadit/SimSearchManifoldLearning.jl

[5] F. Rangel, G. L. D. L. P. Sarracén, B. Chulvi, E. Fersini, P. Rosso, Profiling Hate Speech Spreaders on Twitter Task at PAN 2021, in: G. Faggioli, N. Ferro, A. Joly, M. Maistro, F. Piroi (Eds.), CLEF 2021 Labs and Workshops, Notebook Papers, CEUR-WS.org, 2021.

[6] M. Á. Álvarez-Carmona, E. Guzmán-Falcón, M. Montes-y Gómez, H. J. Escalante, L. Villaseñor-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, September, 2018.

[7] 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).

[8] S. Benzel, A simple stylometric comparator: Nifty assignment, J. Comput. Sci. Coll. 31 (2015) 283–284.

[9] J. Savoy, Lexical analysis of us political speeches, Journal of Quantitative Linguistics 17 (2010) 123–141.

[10] I. Ameer, G. Sidorov, R. M. A. Nawab, Author profiling for age and gender using combinations of features of various types, Journal of Intelligent & Fuzzy Systems 36 (2019) 4833–4843. URL: https://doi.org/10.3233/JIFS-179031. doi:10.3233/JIFS-179031, 5.

[11] J. Soler-Company, L. Wanner, On the relevance of syntactic and discourse features for author profiling and identification, in: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, Association for Computational Linguistics, Valencia, Spain, 2017, pp. 681–687. URL: https://aclanthology.org/E17-2108.

[12] M. A. Álvarez-Carmona, A. P. López-Monroy, M. Montes-y Gómez, L. Villaseñor-Pineda, I. Meza, Evaluating topic-based representations for author profiling in social media, in: M. Montes y Gómez, H. J. Escalante, A. Segura, J. d. D. Murillo (Eds.), Advances in Artificial Intelligence - IBERAMIA 2016, Springer International Publishing, Cham, 2016, pp. 151–162.

[13] Satyam, Anand, A. K. Dawn, S. K. Saha, A statistical analysis approach to author identification using latent semantic analysis, in: CLEF, 2014.

[14] M. Álvarez-Carmona, A. López-Monroy, M. M. y Gómez, L. Villaseñor-Pineda, H. Escalante, INAOE's participation at PAN'15: Author Profiling Task—Notebook for PAN at CLEF 2015, in: L. Cappellato, N. Ferro, G. Jones, E. San Juan (Eds.), CLEF 2015 Evaluation Labs and Workshop – Working Notes Papers, 8-11 September, Toulouse, France, CEUR-WS.org, 2015. URL: http://ceur-ws.org/Vol-1391.

[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.

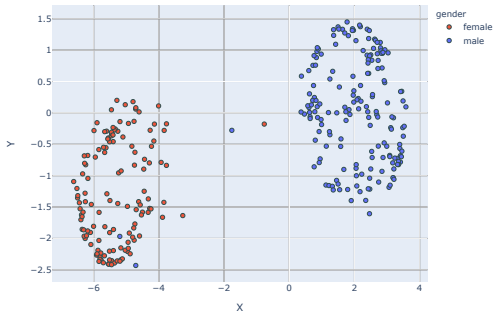
[16] L. McInnes, J. Healy, J. Melville, Umap: Uniform manifold approximation and projection for dimension reduction, 2018. URL: https://arxiv.org/abs/1802.03426. doi:10.48550/ARXIV.1802.03426.

[17] E. S. Tellez, S. Miranda-Jiménez, M. Graff, D. Moctezuma, Gender and language-variety identification with microtc., in: CLEF (Working Notes), 2017.

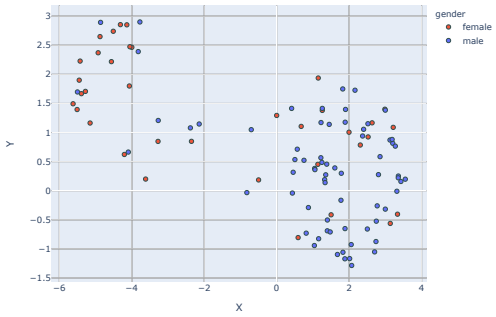
[18] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.

A. Visualization of the training and test partitions for all subtasks

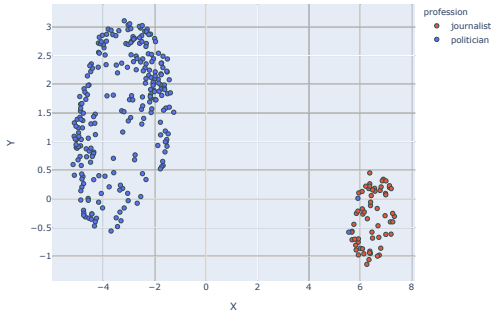
Figure 2 illustrates the low dimensional projections created with the UMAP algorithm using the preprocessing, tokenization, and weighting schemes presented in the manuscript as input. On the left column, we can observe the projections of training datasets. Here we can observe a nearly perfect separation of the datasets, which can also be observed in our Table 1. Note that multiclass ideology is a bit more complex, but it is easy to separate; here, we can observe how related classes share boundaries and how moderate ideologies touch finely. In the right column, we observe the actual projection of the testing set; here, we can observe some of the difficulties our models found to separate classes. We can observe multiclass ideology. Note that we found more crossing examples, where we can observe why both linear and non-linear models work relatively well, and found several issues with generalizations. Something similar is illustrated for gender and profession projections where examples cross boundaries.



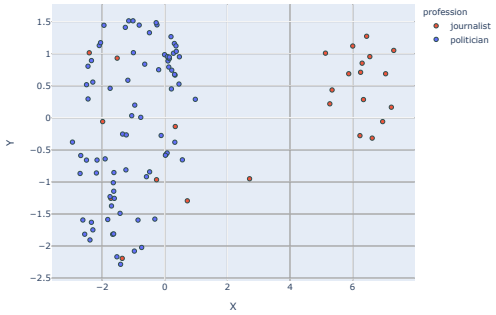
(a) Gender training dataset



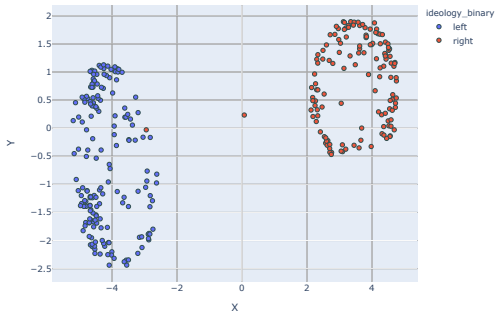
(b) Gender test dataset



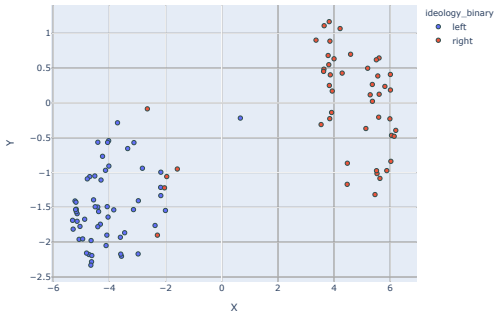
(c) Profession training dataset



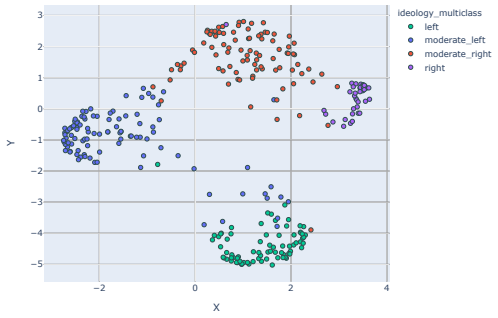
(d) Profession test dataset



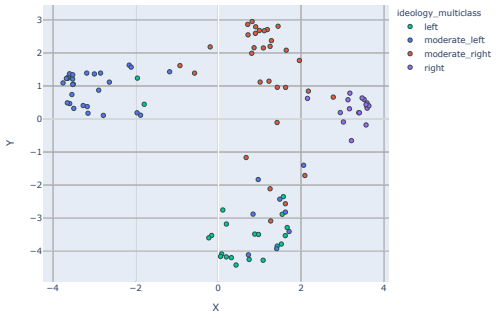
(e) Ideology training dataset



(f) Ideology test dataset



(g) Multiclass ideology training dataset



(h) Multiclass ideology test dataset

Figure 2: Two dimensional UMAP projection of our high dimensional user representation colorized by label.