The Impact of the COVID-19 Pandemic on Bordering Discourses Regarding Migration and Mobility in Europe
Classification report

Olivier Grognuz

February 15, 2022

Introduction

During the first half of the project, tweets of political actors and institutions were manually classified using a predefined set of rules. Consequently, we decided to leverage this labeled database to try and extend the time scope of the project, by using Natural Language Processing (NLP) techniques to automatize the tasks previously done by human coders. This report acts as a summary of the experiments conducted in that regard.

The work was divided into five main classification tasks:

1. **CovidOrNot**: classify tweets as being about covid or not;
2. **Cat**: classify the category of the tweet;
3. **Subcat**: classify the mobility subcategory of the tweet;
4. **Position**: classify the position of the tweet regarding mobility;
5. **Frame**: classify the frame of the tweet regarding mobility.

Note that a complete description of the variables and their code is available in the project's codebook. A brief description is reported in Appendix A. The complete code for this report is hosted on Github[1].

Methodology

For the **CovidOrNot** task, different supervised machine learning models were tested and three linear models were eventually retained: `SGDClassifier`, `LogisticRegression` and `RidgeClassifier`. The `SGDClassifier` was trained with a `hinge` loss, which is equivalent to a linear SVM. We used the implementation provided by the `sklearn` python library [5]. Before running the above models, each tweet was preprocessed with classical text classification methodology (punctuation and stopwords removal), and each token (i.e. word) converted into a vector using the Tf–idf technique, which allows to weigh down frequent tokens.

The models were trained language-wise, by separating tweets in English and French. For other languages, which are not the primary target of the project, we trained another generic model. This detection was done using the `fastText` library [1].

[1] https://github.com/ogrnz/covid-project-helpers

Once trained, the final classification decision of a given tweet was made by using the result of a majority vote of the three classifiers. In other words, a tweet is classified as being about covid only if at least two of the three models predicted that class.

The subsequent tasks concern only tweets that are about covid and in English. The best results were achieved by fine-tuning the `CT-BERTv2` model [4]. Due to the high imbalance of certain classes, we decided to assess the performance of our results with a weighted F1 score instead of plain accuracy. Briefly, this score is the harmonic mean between precision and recall, which is then weighted with the number of tweets in that class.

We decided to set a F1 score threshold at 85%. Any score below that is deemed unsatisfactory. In that regard, we created a subtask for **Subcat**, named **Subcat02**, for which we aggregated all the classes below the 85% threshold into a unique one. Similarly, we devised another strategy for the **Cat** task, called **Cat02**, for which we focused only on identifying the mobility class (code `601`) while grouping the others into class `99`.

Note that only a few different architectures could be tested due to time constraints. We strongly believe that being able to search for optimal hyperparameters and architecture could substantially increase the performance of the model.

Results

The results are shown in Table 1. The **CovidOrNot** task does not have an *epochs* value because the models used do not train by iterating over the set multiple times. The results are obtained by splitting all the retrieved tweets into a training and test set, with respectively 85% and 15% of the tweets. Note that for **Cat02**, even the weighted average F1 score does not correctly reflect the model's performance. In that case, the mobility class (code `601`) could only be correctly identified 85% of the time. Detailed, per-class results are available in Appendix B.

Task	Weighted avg. F1 score (%)	Epochs
CovidOrNot	95	-
Cat	81	13
Cat02	95 (`601`: 85)	10
Subcat	89	35
Subcat02	89	20
Position	94	16
Frame	82	20

Table 1: Weighted average F1 score for the different tasks. The *Epochs* column correspond to the number of iterations done over the whole training set.

Discussion

Overall, this experiment yields satisfactory results for all tasks except **Cat** and **Frame**. However, when looking at the scores per class (B), we see that the results are more ambiguous. Indeed, many scores are just around the 85% threshold, and only few are over it by a large margin. This is in part because the support (number of elements detected in a given class) highly differs. In some cases, especially for **Subcat**, it is clear that the number of elements in a class were not enough to allow the model to correctly train. The tasks **Cat02** and **Subcat02** were designed to alleviate this effect. This quite crude trick does not fundamentally change our results, but provides a hint as to

how to design an updated version of the codebook, should we want to continue the project with this automatic tweets classification.

In that case, a detailed analysis of the results imperatively needs to be conducted. We need to determine if the different actors are somewhat equally classified, and especially to make sure that no specific actor is all the time missclassified. Although not showing in the overall numbers, this could heavily bias the resulting qualitative interpretation.

While further analysis needs to be done, the above risk can be mitigated by improving the overall performance of the models on the different tasks. This could include ways to enhance the current dataset by using (i) **data augmentation techniques**, to essentially generate new tweets of the classes with (very) low support and allowing the models to train on them. Also, we believe that an extensive (ii) **search for the optimal hyperparameters** and architecture could substantially improve the performance. Others suggestions may include trying a completely (iii) **different family of models**, for example XLNet [6], which currently achieves state-of-the-art performance on text classification tasks.

Also, the current experiment focuses on English tweets only. Ideally, a similar analysis should be carried at least on French tweets, because they represent a significant portion of the whole database. Possible models for this include FlauBERT [3] or BERT-multilingual [2].

Conclusion

This experiment shows the value of leveraging latest advancements in other fields to drastically enlarge the original scope of the project and spend more time on the data analysis, rather than on the coding part. In its current state, the experiment could be used as an assistant to a human coder. Further work still needs to be conducted in order to completely replace humans for those tasks, especially when determining the **Frame** of a tweet.

References

[1] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[2] J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[3] H. Le, L. Vial, J. Frej, V. Segonne, M. Coavoux, B. Lecouteux, A. Allauzen, B. Crabbé, L. Besacier, and D. Schwab. Flaubert: Unsupervised language model pre-training for french, 2020.

[4] M. Müller, M. Salathé, and P. E. Kummervold. Covid-twitter-bert: A natural language processing model to analyse covid-19 content on twitter. *arXiv preprint arXiv:2005.07503*, 2020.

[5] 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, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[6] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le. Xlnet: Generalized autoregressive pretraining for language understanding, 2020.

A Variables description

Task	Class	Description
Cat	601	Mobility
	602	Health
	603	Economy
	604	Democracy
	605	Society
	606	Youth/Education
	607	Environment
Subcat	60100	Border Closing
	60101	Border Opening
	60102	Schengen
	60103	International Travel
	60104	European Travel
	60105	Local Travel
	60106	Immigration
	60107	Cross-border Workers
	60108	Tourism
	60109	Asylum
	60110	Visas
	60111	Family Exceptions
	60112	Citizen Exceptions
	60113	Lockdown
	60114	Isolation
	60115	Social Distance
	60116	Commercial Flux
Position	0	Permissive position on mobility
	1	Restrictive position on mobility
Frame	1	Pragmatic
	2	Utilitarian
	3	Normative
	4	Communitarian
	5	European
	6	Cosmopolitan

B Detailed results

Task	Class	F1 score (%)	Support
Cat	601	84	1404
	602	86	4213
	603	78	692
	604	71	1098
	605	72	1370
	606	74	70
	607	57	40
Cat02	601	85	1441
	99	97	7446
Subcat	60100	57	6
	60101	60	5
	60102	86	11
	60103	87	176
	60104	76	68
	60105	90	329
	60106	77	22
	60107	79	15
	60108	80	22
	60109	89	68
	60110	83	5
	60111	67	2
	60112	0	0
	60113	85	128
	60114	92	171
	60115	92	267
	60116	95	167
Subcat02	60103	86	170
	60105	89	328
	60109	90	67
	60113	85	121
	60114	91	173
	60115	92	274
	60116	94	171
	60199	85	158
Position	1	94	828
	0	93	634
Frame	1	85	625
	2	77	136
	3	82	293
	4	82	167
	5	75	68
	6	82	173