Finding the needle in a haystack: Extraction of Informative COVID-19 Danish Tweets

Benjamin Olsen and **Barbara Plank**
Department of Computer Science
IT University of Copenhagen
{beao, bapl}@itu.dk

Abstract

Finding informative COVID-19 posts in a stream of tweets is very useful to monitor health-related updates. Prior work focused on a balanced data setup and on English, but informative tweets are rare, and English is only one of the many languages spoken in the world. In this work, we introduce a new dataset of 5,000 tweets for finding informative COVID-19 tweets for Danish. In contrast to prior work, which balances the label distribution, we model the problem by keeping its natural distribution. We examine how well a simple probabilistic model and a convolutional neural network (CNN) perform on this task. We find a weighted CNN to work well but it is sensitive to embedding and hyperparameter choices. We hope the contributed dataset is a starting point for further work in this direction.

1 Introduction

As of late July 2021, the COVID-19 Corona virus continues to spread with close to 200M infected patients and more than 4M people who have died to this horrible disease[1] striking fear into the hearts of all people around the world.

Due to this, COVID-19 has become the most discussed topic in the news and most of the "official" sources that keep track of the number of dead, infected, and recovered cases, are not frequently kept up to date (e.g. WHO only updates pandemic information once a day). Since the news or online monitoring systems need real-time updates, they turn to social media platforms such as Twitter, to gather real-time pandemic information (Banda et al., 2021; Aiello et al., 2021). However, given that this also happens to be one of the most popular topics on social media, many of these posts contain false or uninformative information (Krittanawong et al., 2020; Weinzierl et al., 2021) that can be difficult to filter out.

In 2020, a shared task at WNUT (Nguyen et al., 2020) was organized to filter out informative COVID-19 related information from an English dataset consisting of 10k COVID-19 related tweets. The shared task dataset has close to an equal class distribution between two labels: INFORMATIVE (4,719 tweets) and UNINFORMATIVE (5,281). A total of 55 teams submitted different models and it was shown in Nguyen et al. (2020) that the two best models solved this task with an F1 score of 90.96 using an English COVID-Twitter-BERT model (Kumar and Singh, 2020; Giovanni Møller et al., 2020). In this paper, we keep the real-world non-balanced distribution between informative and non-informative tweets and extend the task to a less-privileged language.

To answer the question whether this task is doable in light of the expected skewed distribution, this paper contributes a new dataset for Danish by starting from the same annotation guidelines (Nguyen et al., 2020). Due to the high imbalance between classes in this dataset (out of 5,000 tweets, at most 3% are informative tweets), the goal is to create a model that performs well for this task despite the high imbalance. No Danish COVID Twitter BERT model exists, hence we experiment with multilingual and Danish BERT and Danish Twitter word2vec embeddings.

2 Dataset

This section outlines how the dataset of Danish COVID-19 related tweets was created, annotated, pre-processed and split for the experiments.

2.1 Annotation guidelines

Our annotation guidelines depart from those by Nguyen et al. (2020). They define two labels: INFORMATIVE and UNINFORMATIVE. We introduce a new label named INFORMATIVE- to capture

[1] https://www.worldometers.info/coronavirus/

difficult and non-Danish cases,[2] and rename the INFORMATIVE label to INFORMATIVE+. In particular, INFORMATIVE+ and UNINFORMATIVE are identified identically to the guidelines presented in (Nguyen et al., 2020). INFORMATIVE- is a label which contains tweets that are difficult to label as either INFORMATIVE+ or UNINFORMATIVE and tweets that are informative but written in English (for a full description of the mapping between tweets and the three classes, see appendix A.1).

2.2 COVID-19 related tweet collection

We collected tweets that contain the hashtag: "#covid19dk" from March 2020 to October 2020. The hashtag suffix 'dk' is often used in Danish tweets. Here we opted for dataset construction on the basis of this hashtag to collect mostly Danish tweets, as distinguishing between closely related languages poses a significant challenge (Ljubesic et al., 2007; Tiedemann and Ljubešić, 2012; Haas and Derczynski, 2021; Jauhiainen et al., 2019). Extending the dataset with a keyword-based approach is an interesting future venue.

We follow the process of selecting tweets for this task as in Nguyen et al., 2020 (see appendix A.2), with the only exception of keeping tweets written by a user with less than 500 followers, since Danish Twitter users have fewer followers in general. This resulted in approximately 9,000 COVID-19 related Danish Tweets. The first 5,000 tweets were selected for annotation. We release the full filtered dataset (Twitter ids of all 9,000 tweets) for future work on this collection.[3] See the appendix for annotation guidelines and data quality.

2.3 Data partitions

From the 5,000 tweets, we ended up with a small dataset consisting of 500 tweets (125 INFORMATIVE+, 19 INFORMATIVE-, and 356 UNINFORMATIVE) and a larger dataset with 4,500 more UNINFORMATIVE tweets, thus increasing the class imbalance (see appendix A.3 for a full description of the annotation process). To evaluate data quality, 500 tweets were labelled by two annotators resulting in high agreement (0.93 raw agreement and 0.83 Cohen's Kappa score). These were split into training, validation, and final test sets as two 80/20 stratified splits by first creating an 80/20 split of the 500 tweets and then an 80/20 split of the 400 tweets to form the train/validation datasets. The stratified split ensures that each set contained the same proportion of INFORMATIVE+, INFORMATIVE-, and UNINFORMATIVE tweets.

The 400 tweets were merged with 80% of the 4,500 UNINFORMATIVE tweets to form a second larger dataset. An 80/20 stratified split is also applied on the large development dataset to form a larger set of train/validation datasets. The remaining 20% of the 4,500 UNINFORMATIVE tweets were added to the 100 tweets in the final test set to form a 2nd larger final test set.

To summarize, the final two datasets are: the small dataset of 100 tweets (labels: 25 INFORMATIVE+, 4 INFORMATIVE-, and 71 UNINFORMATIVE) and the large dataset of 1000 tweets (labels: 25 INFORMATIVE+, 4 INFORMATIVE-, and 971 UNINFORMATIVE).

3 Methodology

This section outlines the models that will be used.

3.1 Naive Bayes

We used Naive Bayes with word unigrams (Jurafsky and Martin, 2021) to compare our more complex model to a simple probabilistic baseline.

3.2 CNN

We experimented with a Convolutional Neural Network and test both word embeddings derived by word2vec and contextualized BERT embeddings. The CNN is based on the work of Kim (2014). The embedding layer is initialized with pretrained word2vec embeddings of size 400 that are based on Danish Twitter data (available from another WNUT 2021 shared task).[4] We also use pre-trained BERT embeddings as static input to the CNN model with a class weighted loss. Both Multilingual BERT (MBERT [5]) and Danish BERT (DBERT [6]) embeddings are used because we want to determine which is the best fit for this particular dataset.

We extensively tuned our CNN with a hyperparameter search (outlined in 3.4), which resulted in a CNN with 4 different filter sizes 1,3,5, and 7 with

[2] One could also opt to remove any such instances in future work, as such cases are very infrequent and difficult to model. However, as it does contain relevant information and we wanted to gauge their frequency and modelling difficulty, it was decided to keep it in.

[3] Tweet ids and annotations are available at: https://github.com/beaol/

[4] https://t.co/n9Ha1t26tg?amp=1

[5] https://huggingface.co/bert-base-multilingual-uncased

[6] https://huggingface.co/Maltehb/danish-bert-botxo

100 feature maps each, max-pooling which takes the most important feature of each filter, a dropout layer with a rate of 0.5, and a fully connected layer with an output dimension equal to the number of classes (i.e. 3). During training, we use a word dropout (to UNK) with probability $\frac{1}{100}$.

3.3 Class imbalance

Due to the high imbalance in the dataset, we experiment with a weighted loss function that reacts more to wrong predictions on the smaller classes (i.e. INFORMATIVE+ and INFORMATIVE-) and less on the larger classes (i.e. UNINFORMATIVE). This is done by using the inverse square root number of samples:

$$\frac{1}{\sqrt{class\_samples}}$$

Preliminary experiments confirmed this was superior to weighting inversely proportional to size.

3.4 Hyperparameter search

Given that the chosen NN architecture has many hyperparameters, i.e. number of filters, filter sizes, L2 regularization, embedding size, etc, we needed to test multiple variations to gain insight into which factors are the most impactful on performance.

A general take-away from the hyperparameter search is that the CNN showed the best performance with an L2 regularization of 0 (even though this makes the model overfit) and either few feature maps and a larger embedding size or more feature maps and a smaller embedding size.

These tests also showed that this type of model is quite sensitive to changes in hyperparameters as well as using different manual seeds. The performance can vary a lot in a positive or negative direction by e.g. changing the learning rate by a small amount. To illustrate this, we report both mean performance and max performance (macro F1) achieved by the best run.

The same observations were made when choosing between different class weighting techniques, such as inverse number of samples, which can be just as significant as changing the other parameters.

4 Evaluation

This section outlines the conducted experiments to examine how well each model performs on the given datasets.

Model	**Mean F1**	**Max F1**
Naive Bayes	74.11 ±2.00	76.53
CNN (no class weights)	63.88 ±7.87	72.58
CNN	73.45 ±4.01	79.10
CNN MBERT	**82.34** ±5.10	**87.13**
CNN DBERT	75.28 ±2.27	78.27

Table 1: Model results on development dataset with 400 tweets. Mean F1 macro score over three runs.

Model	**Mean F1**	**Max F1**
Naive Bayes	49.94 ±4.22	53.72
CNN (no class weights)	52.35 ±5.60	59.78
CNN	**63.47** ±5.86	**68.29**
CNN MBERT	56.93 ±4.07	62.57
CNN DBERT	60.45 ±1.61	62.22

Table 2: Model results on development dataset with 4,000 tweets. Mean F1 macro score over three runs.

4.1 Analysis disclaimer

To make sure no possibly privacy-sensitive information is shown, the tweets will not be shown in this paper. However, they will still be analyzed based on the predictions made by the model, but instead of showing the tweet it will be described.

4.2 Evaluation on development data

As found in Table 1, first we observe that class weighting is essential for the CNN. The class weighted CNN model with MBERT embeddings has the best mean F1 macro score of 82.34 and the best max F1 macro score of 87.13 on the smaller and more balanced dataset. When using the Danish DBERT embeddings we see a large drop in performance, which is surprising when considering that most of the dataset is in Danish.

Again, weighting is essential for the even more skewed larger dataset. However, a different pattern emerges in Table 2, where the best performing model on the full dataset on average and on a single run is the class weighted CNN model which uses pre-trained Danish Word2Vec embeddings to initialize the embedding layer. However, this is only based on three runs and since the standard deviation is high, it might not be the best performing model after doing more runs and could be outperformed by one of the CNN models with BERT embeddings. A combination of the Word2Vec and BERT embeddings could lead to better results in future, similar to multi-channel CNNs proposed by (Kim, 2014).

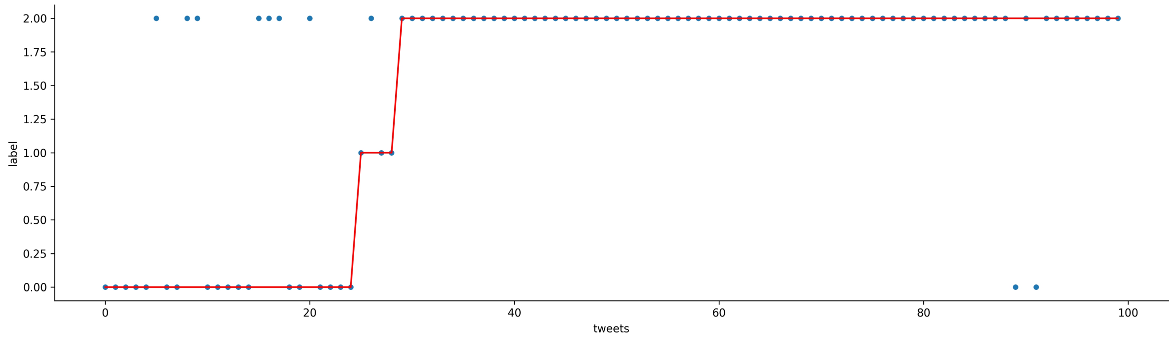

Figure 1: Staircase plot of testing on the final test dataset of 100 tweets made by the CNN model with class weighting using MBERT embeddings. Y-axis: 0=INFORMATIVE+,1=INFORMATIVE-,2=UNINFORMATIVE. Dots: predictions; red line: gold standard.

4.3 Evaluation on final test data

The results on the final test (Table 3) show that the CNN model with MBERT embeddings remains the best performer on a single run. However, the model trained on DBERT embeddings has become the best perfomer on average on this dataset while keeping the lowest standard deviation as seen on the development datasets as well. This shows that in challenging highly skewed distributions, having an optimal CNN configuration is difficult to obtain.

Similarly for the final test dataset with 1000 tweets (Table 4), the CNN model with MBERT embeddings is the best performer on a single run. However, for this dataset we also see that it is the best performer on average although it continues to get the highest standard deviation of the different approaches on the test dataset, thus showing to be the most unstable model on the test datasets.

5 Analysis

Given that INFORMATIVE- is very infrequent, we analyse the per-class predictions. We examine whether the model learns to accurately predict this class at all.

We create a different representation of the confusion matrix, which we call the *Staircase plot*, to better visualize exactly which tweets were labelled correctly and incorrectly by a given model. This enables us to visually compare individual tweet classifications between the different models. An example in Figure 1 shows the Staircase plot of the best single run produced by the weighted CNN model with static MBERT. We notice that the model is very good at INFORMATIVE-, in fact only misses one instance. Most confusions stem from the INFORMATIVE+ label, which most often got predicted as UNINFORMATIVE.

Some of these tweets are in fact difficult for all of the models to classify correctly (see appendix A.4). For example, the sixth INFORMATIVE+ is misclassified by all models as an UNINFORMATIVE tweet. This particular tweet shortly mentions how many people have been put in intensive care due to COVID-19 and then questions the need for continuing with the lockdown. We believe that it is the final part of the tweet that confuses the models, possibly also because the word “intensive” has not been weighted high enough in terms of significant feature descriptors of the INFORMATIVE+ class.

Another common misclassification is on the INFORMATIVE- tweets. The training dataset contains a total number of 15 INFORMATIVE- tweets where 5 of these are in Danish and 10 are in English. The imbalance between Danish and English in this case is present in the misclassification, since the common tweet that all models get wrong is a Danish tweet and the remaining 3 tweets are in English. Based on these results it appears that the model learns to distinguish between Danish and English and simply predicts English tweets as INFORMATIVE-. We would therefore also expect that English UNINFORMATIVE tweets are misclassified as INFORMATIVE-. This is the case for all models except the CNN MBERT model, which uses multilingual embeddings hence having the ability to distinguish between English INFORMATIVE- and UNINFORMATIVE tweets.

Two challenging UNINFORMATIVE tweets are: one containing some numbers and the word “infected”, and the other tweet contains some numbers as well and the word “death”. It makes sense that

Model	Mean F1	Max F1
Naive Bayes	74.78 ±2.63	77.28
CNN (no class weights)	74.40 ±4.23	78.93
CNN	74.05 ±7.04	82.50
CNN MBERT	74.10 ±13.27	**86.31**
CNN DBERT	**78.12** ±1.13	79.30

Table 3: Model results on final test set with 100 tweets for the model trained on the small development data.

Model	Mean F1	Max F1
Naive Bayes	54.13 ±4.19	60.07
CNN (no class weights)	50.42 ±1.08	51.77
CNN	61.35 ±6.01	68.08
CNN MBERT	**70.72** ±9.63	**84.15**
CNN DBERT	60.17 ±1.68	62.55

Table 4: Model results on final test set with 1000 tweets for the model trained on the larger development data.

the models would find these tweets difficult given that numbers and the words: “corona”, “infect”, “recover”, “death”, and “test”, are considered to be strong indicators of an informative tweet when used in the same tweet.

6 Conclusion

We contribute a new Danish dataset consisting of COVID-19 related tweets and benchmarked neural models. The imbalanced dataset made the task very challenging, but most realistic. The most effective model tested in this paper was the CNN model with class weighting. However, there is no single best embedding setup that consistently outperforms all others on the two development datasets. The CNN model was found to be very sensitive to randomness and hyperparameters, yet loss weighting was essential in this setup as otherwise the CNN performs similar to a simple Naive Bayes model. Future work includes extending the created dataset, comparing supervised training with semi-supervised methods, and a comparison of the CNNs used in this work to fine-tuning models.

Acknowledgements

This research is in part supported by the Independent Research Fund Denmark (DFF) grant 9131-00019B. We thank ITU’s High-performance Computing Cluster team for computational resources used in this work, and the anonymous WNUT-2021 reviewers for their thorough feedback.

References

Luca Maria Aiello, Daniele Quercia, Ke Zhou, Marios Constantinides, Sanja Šćepanović, and Sagar Joglekar. 2021. How epidemic psychology works on twitter: evolution of responses to the covid-19 pandemic in the us. *Humanities and Social Sciences Communications*, 8(1):1–15.

Juan M Banda, Ramya Tekumalla, Guanyu Wang, Jingyuan Yu, Tuo Liu, Yuning Ding, Ekaterina Artemova, Elena Tutubalina, and Gerardo Chowell. 2021. A large-scale covid-19 twitter chatter dataset for open scientific research—an international collaboration. *Epidemiologia*, 2(3):315–324.

Anders Giovanni Møller, Rob van der Goot, and Barbara Plank. 2020. NLP north at WNUT-2020 task 2: Pre-training versus ensembling for detection of informative COVID-19 English tweets. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, pages 331–336, Online. Association for Computational Linguistics.

René Haas and Leon Derczynski. 2021. Discriminating between similar nordic languages. In *Proceedings of the Eighth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 67–75, Kiyv, Ukraine. Association for Computational Linguistics.

Tommi Jauhiainen, Marco Lui, Marcos Zampieri, Timothy Baldwin, and Krister Lindén. 2019. Automatic language identification in texts: A survey. *Journal of Artificial Intelligence Research*, 65:675–782.

Daniel Jurafsky and James H. Martin. 2021. *Speech and Language Processing*, (3rd ed. draft) edition, chapter 4. Pearson Prentice Hall.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Chayakrit Krittanawong, Bharat Narasimhan, Hafeez Ul Hassan Virk, Harish Narasimhan, Joshua Hahn, Zhen Wang, and WH Wilson Tang. 2020. Misinformation dissemination in twitter in the covid-19 era. *The American journal of medicine*, 133(12):1367.

Priyanshu Kumar and Aadarsh Singh. 2020. NutCracker at WNUT-2020 task 2: Robustly identifying informative COVID-19 tweets using ensembling and adversarial training. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, pages 404–408, Online. Association for Computational Linguistics.

Nikola Ljubesic, Nives Mikelic, and Damir Boras. 2007. Language indentification: How to distinguish similar languages? In *2007 29th International Conference on Information Technology Interfaces*, pages 541–546. IEEE.

Dat Quoc Nguyen, Thanh Vu, Afshin Rahimi, Mai Hoang Dao, Linh The Nguyen, and Long Doan. 2020. WNUT-2020 task 2: Identification of informative COVID-19 English tweets. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, pages 314–318, Online. Association for Computational Linguistics.

Jörg Tiedemann and Nikola Ljubešić. 2012. Efficient discrimination between closely related languages. In *Proceedings of COLING 2012*, pages 2619–2634.

Maxwell Weinzierl, Suellen Hopfer, and Sanda M Harabagiu. 2021. Misinformation adoption or rejection in the era of covid-19. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 15, pages 787–795.

A Appendices

A.1 Annotation guidelines

Nguyen et al. (2020) defined the "INFORMATIVE" label as Tweets which mentions suspected cases, confirmed cases, recovered cases, deaths, number of tests performed as well as location or travel history associated with the confirmed/suspected cases. "INFORMATIVE" Tweets also have to mention specific quantities (e.g. "1 new case" or "five deaths today") or a quantity which can be inferred (e.g. "100 people were tested today, 25% were positive"), but not percentages where the quantity cannot be inferred(e.g. an "UNINFORMATIVE" tweet: "0.5% of the danish population has tested positive"). We named this label: "INFORMATIVE+". Another requirement for an "INFORMATIVE+" tweet is that it cannot contain rumors or predictions regarding the COVID-19 related information.

We also define a new label: "INFORMATIVE-", which will be used for Tweets that are difficult to label as either "INFORMATIVE+" or "UNINFORMATIVE" or if the Tweet is "INFORMATIVE+" but in English instead of Danish. An example of a tweet that is difficult to label as "INFORMATIVE+" or "UNINFORMATIVE" is: "No number of deaths since the 10th death", which has an informative part where it is mentioned that there have been no new death cases, but it also has an uninformative part where it says that this is only the case since the 10th death. All models will also try to classify tweets which are labelled as "INFORMATIVE-", unlike previous research which only made binary classifiers for "INFORMATIVE+" and "UNINFORMATIVE", with the intention of discovering if the developed models are able to distinguish between English and Danish and/or extract the features which made the annotators uncertain about the class label.

The third label is: "UNINFORMATIVE", which is used for all Tweets that cannot be labelled as "INFORMATIVE+" or "INFORMATIVE-", which includes English tweets that do not contain any informative COVID-19 related information.

A.2 Data collection

Tweets containing less than 10 words (including hashtags and user mentions) are filtered out. This was done because it was determined that most tweets with less than 10 words in the dataset did not contain informative information, after labelling a large amount of tweets. Nguyen et al., 2020 also filtered out Tweets from users with less than 500 followers, which we decided not to do since Danish Twitter users have fewer followers in general. The dataset is iterated over from start to end (in terms of time posted) where hashtags, user mentions, and the re-tweet tag is removed (if it is there). The updated tweet text is then made all lower case before being stored. If two tweets are identical, then the first tweet is kept and the new tweet is filtered out. This resulted in around 9,000 COVID-19 related Danish Tweets.

A.3 Data annotation process

From the 9,000 Tweets, one annotator (with a computer science background) independently assigned 5,000 tweets with one of the three labels: "INFORMATIVE+", "INFORMATIVE-", and "UNINFORMATIVE". Around 100 of these tweets were labelled "INFORMATIVE+" and the remaining tweets were mostly labelled as "UNINFORMATIVE". From these 5,000 Tweets, all of the "INFORMATIVE+" and "INFORMATIVE-" were selected and the "UNINFORMATIVE" Tweets were selected at random until the dataset contained 500 Tweets in total. The only criteria for the "UNINFORMATIVE" Tweets were that half of them should contain numbers to prevent the future model from learning that "INFORMATIVE+" Tweets contain numbers.

These 500 Tweets were also labelled by second annotator with a linguistics background, which then made it possible to measure the inter-annotator agreement to assess the quality of the annotations.

From the 500 tweets, there were only 35 disagreements (raw agreement of 0.93) between the annotators, which resulted in a Cohen's Kappa score of 0.83 and this can be interpreted as almost perfect agreement.

A third opinion was seeked on the 35 Tweets which caused disagreement to finalize the annotation (majority votes for a label wins). After this step, we had 500 tweets (125 "INFORMATIVE+", 19 "INFORMATIVE-", and 356 "UNINFORMATIVE"), which had a Cohen's kappa score of 0.949 with the first annotator's labels and 0.880 with the second annotator's labels.

Given that the first Cohen's kappa score was almost perfect, we decided to also use the 4500 other "UNINFORMATIVE" tweets that the first annotator labelled.

A.4 Dataset figures

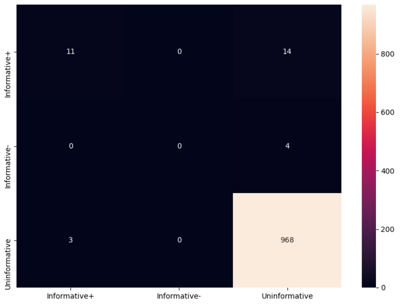

Figure 2: CNN without class weighting confusion matrix from the larger final test set of 1000 tweets

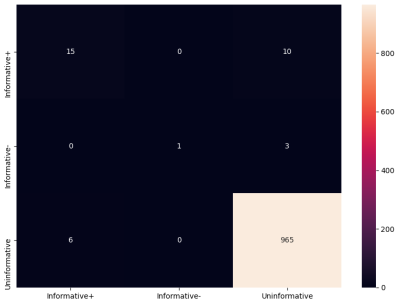

Figure 3: CNN with class weighting confusion matrix made from the larger final test dataset of 1000 tweets

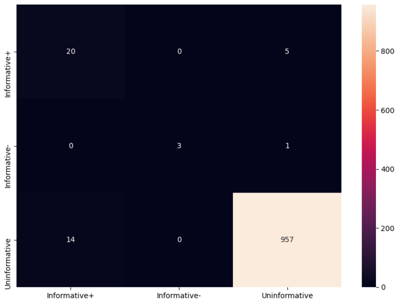

Figure 4: CNN with class weighting using MBERT confusion matrix on the larger final test set of 1000 tweets

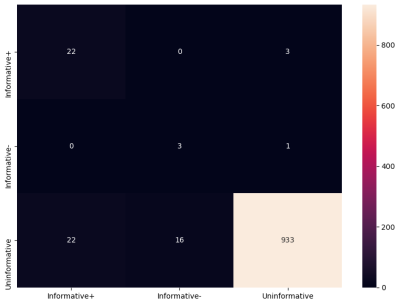

Figure 5: CNN with class weighting using DBERT confusion matrix from the larger final test set of 1000 tweets

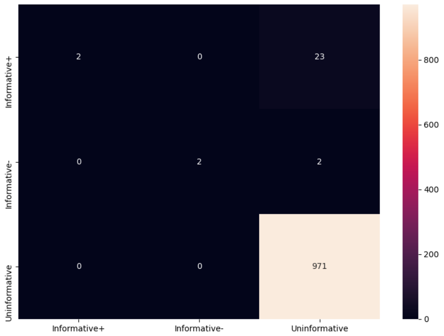

Figure 6: Naive Bayes confusion matrix made from the larger final test dataset of 1000 tweets

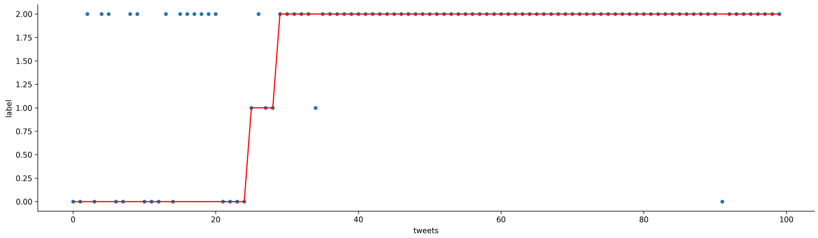

Figure 7: Staircase plot of the testing on the final test dataset of 100 tweets made by the Naive Bayes model

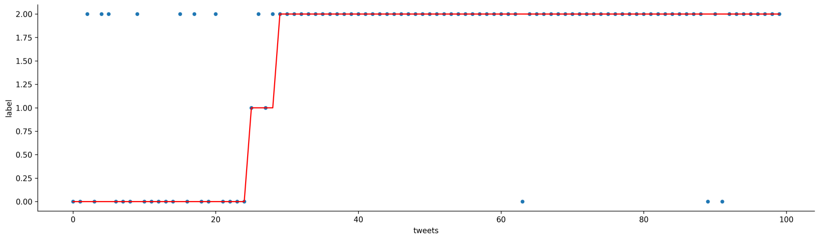

Figure 8: Staircase plot of the testing on the final test dataset of 100 tweets made by the CNN model without class weighting

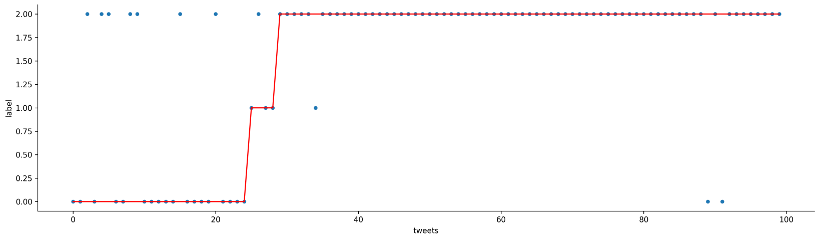

Figure 9: Staircase plot of the testing on the final test dataset of 100 tweets made by the CNN model with class weighting

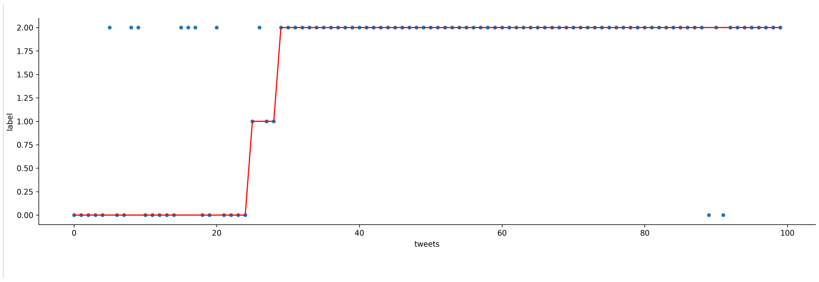

Figure 10: Staircase plot of the testing on the final test dataset of 100 tweets made by the CNN model with class weighting using MBERT embeddings

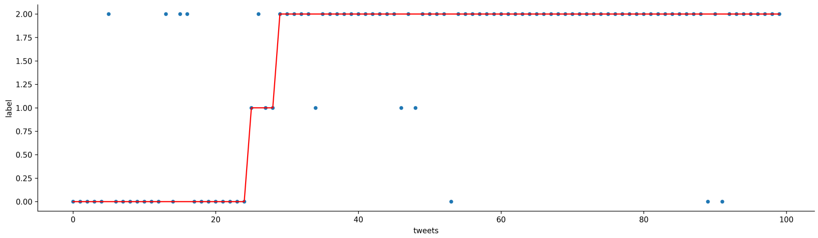

Figure 11: Staircase plot of the testing on the final test dataset of 100 tweets made by the CNN model with class weighting using DBERT embeddings