

CLEF eRisk 2022: Detecting Early Signs of Pathological Gambling using ML and DL models with dataset chunking

Tudor-Andrei Dumitraşcu¹

¹University of Bucharest, Romania

Abstract

The first shared task of CLEF eRisk 2022 proposes to investigate how gambling addiction can be detected from texts posted by users on the social platforms. It is an interesting proposal because the connection between language and mental health is studied for many years, and interesting NLP insights could be revealed in this case. Any finding can be used in the early treatment of a mental disease, avoiding the chances of developing a serious condition. Our approach to this task includes both standard Machine Learning and Deep Learning solutions applied to the provided dataset. We then modified the dataset in different ways in order to find better correlations.

Keywords

machine learning, deep learning, dataset processing, chunking

1. Introduction


If early signs of any pathological behaviour would be transparent to us, most of the suffering caused by diseases can be minimized. Any preventive intervention is beneficial to the overall health of an individual. Pathological gambling affects the quality of life of anyone who excessively engages in casino/sports betting and video games activities.

2. Related work

The current task is a continuation of the previous year's T1 task. The training data is composed of all 2021's T1 test users. In 2021 no training data was provided to the participants and they needed to search online sources and to create a training data set from scratch.

Reddit is a social platform that allows researchers to use the public data it hosts for academic purposes. The work of Bucur et al. [1], the last year's participating BLUE team, included a training data set constructed from the r/GamblingAddiction and r/problemgambling subreddits as positive examples. The number of negative examples was chosen to match the number of posts in the positive class and it was gathered from various users active on other subreddits.


CLEF 2022 – Conference and Labs of the Evaluation Forum, September 05–08, 2022, Bologna, Italy

 tudorandrei.dumitrascu@gmail.com (T. Dumitraşcu)

 <https://github.com/TudorAndrei> (T. Dumitraşcu)



© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

The authors fine-tuned a pre-trained BERT classifier with an AdamW optimizer having a learning rate of 0.0002. In predicting the positive class, they used an aggressive threshold of 0.99 and 0.98 on the output probability to decide if the user is at risk of developing a gambling addiction. The purpose of this high threshold is to minimize false positives, ensuring a decision will be made only if a post contains any addiction indicators.

The results they obtained had a high recall score with a low precision score and low F1 score. They concluded that the problem of predicting the early signs of gambling addiction is far from being solved and further research is strongly needed.

The work of Maupone et al.[2], the last year's participating RELAI team, also included creating a data set from scratch using social media posts for both training and making predictions. They used two authorship attribution approaches that assess whether a test users belongs to two different sets of gambling users.

The first set of gambling users was constructed from testimonials published on the Gambler's Help site ¹. The second set was constructed based on questionnaires found on websites such as Gamblers Anonymous Montreal ².

The Embedding Topic Model (ETM) presented in the article written by Dieng et al. [3] was the authors' model selection for extracting the topics present in their training corpus. ETM is a model that combines traditional topic models such as Latent Dirichlet Allocation (LDA) with word embeddings (word2vec).

RELAI team's results were similar to the BLUE team's results, obtaining a high recall score, but a low precision score and F1 score, further enforcing the need for additional NLP research in this mental health area.

3. Dataset

The dataset consists of multiple XML files containing the user posts, timestamps for each post and the title of the post. In order to use the data in a meaningful manner, we extracted the user posts using three different methods:

1. Combining all the posts into a single text file. The number of files is equal to the number of users.
2. Each user posts is saved in file, seen as an individual sample which inherits the label from the user. This results in a great number of individual files and each file of various length.
3. Chunking the user posts into fixed size lengths and create multiple entries for the minority class. This is detailed in subsection 3.1.

3.1. Chunking

In order to balance the disparity of the two classes, one method we have implemented was to split the users into sub users using a chunking algorithm. Due to the fact that collecting all posts of a user in a single file wasn't feasible because the length of the posts would vary greatly,

¹<https://gamblershelp.com.au/>

²<http://gamontreal.ca/>

we decided to split the texts into fixed sizes, defined as *Max Len*. For users with label 1, we also split the whole text into chunks of *Max Len* but the starting point of the next chunk is defined by a *Chunk Delay*. Meaning that *USER Y_1* would retain text from *USER Y_0*. We did this in order to capture more information about gambling addiction. At the end of the processing we ended up with 22k positive examples and 44k negative examples. A visual representation of the algorithm can be seen in Figure 1.

4. Machine learning approach

The Machine Learning approach included the following text normalization techniques applied to the subjects' posts: lowercasing, whitespace removal, URL removal, word tokenization, stop-words removal, punctuation removal and word stemming.

After processing the provided XML files, the resulting training dataset had an unbalanced number of examples for each of the two classes we needed to make predictions (0=NOT ADDICTED, 1=ADDICTED). It contained 2184 examples with label 0 and 164 examples with label 1. We also used a stratified 5-fold cross-validation to try to reduce the class imbalance in the train/validation splits. For feature extraction we used the Bag-of-Words (BOW) and Term Frequency-Inverse Document Frequency (TF-IDF) models with additional properties for extracting relevant features such as removing the rare words ($\text{min_df}=0.2$) or frequent words ($\text{max_df}=0.8$), constructing 2-grams and 3-grams.

In the first run of the ML models, we used the unbalanced dataset with no other modification made to the distribution of the examples across the two classes. The best results we obtained for this unbalanced dataset are presented in Table 1.

We observed that the scores are too high, and we were skeptic about these results. One reason for such high scores could be the unbalanced nature of the dataset. We also observed that training the models with 2-grams and 3-grams brought down the overall performance in all cases. Also, in some cases, dropping the most frequent words or the rare words improved the results.

Next, we explored other ways of constructing and balancing the training dataset. We extracted from the collection of posts with label 1 each post in individual data points labeled with 1. From the collection of posts with label 0 we formed chunks of 20 posts in individual examples. The total examples were 54840 with label 1 and 52989 with label 0, resulting in a more balanced dataset. After the second run of the ML models, the results we obtained for this version of the training dataset are presented in Table 2.

We observed that the new results were similar to the results from the first run, they were also too high. Next, we decided to further modify the training dataset to include for the negative examples (labeled with 0) only the first 25 posts, stored individually (not chunked). We discarded the remaining negative posts from the collection. This approach resulted in a training dataset balanced differently, with the same number of 54840 for positive examples and a total of 52723 posts for the negative examples.

After the third run of the ML models, the results we obtained were lower than the first two runs and are presented in Table 3.

We concluded that this third run may not be entirely accurate because it discarded a big part

Table 1

Best results for the ML approach in the first run, on the unbalanced dataset.

| ML + feature extractor | Precision | Recall | F1 |
|---------------------------------------|------------------|---------------|-------------|
| LinearSVC + TF-IDF (min_df=0.2) | 0.99 | 0.83 | 0.9 |
| LogisticRegression + BOW (max_df=0.8) | 0.87 | 0.87 | 0.87 |
| RandomForest + BOW (min_df=0.2) | 1.00 | 0.46 | 0.63 |
| KNeighbors + BOW (max_df=0.8) | 0.96 | 0.73 | 0.83 |
| DecisionTree + TF-IDF | 0.9 | 0.91 | 0.91 |

Table 2

Best results for the ML approach in the second run, on the balanced dataset.

| ML + feature extractor | Precision | Recall | F1 |
|---------------------------------|------------------|---------------|-------------|
| LinearSVC + TF-IDF | 0.86 | 0.89 | 0.88 |
| LogisticRegression + BOW | 0.87 | 0.96 | 0.91 |
| RandomForest + TF-IDF | 0.91 | 0.89 | 0.9 |
| KNeighbors + BOW | 0.63 | 0.87 | 0.73 |
| DecisionTree + TF-IDF | 0.9 | 0.91 | 0.91 |

Table 3

Best results for the ML approach in the third run, on the balanced dataset with discarded posts.

| ML + feature extractor | Precision | Recall | F1 |
|------------------------------------|------------------|---------------|-------------|
| LinearSVC + TF-IDF | 0.78 | 0.66 | 0.71 |
| LogisticRegression + TF-IDF | 0.79 | 0.66 | 0.72 |
| KNeighbors + BOW | 0.67 | 0.55 | 0.60 |
| DecisionTree + TF-IDF | 0.7 | 0.65 | 0.68 |

of the posts from the 0 labeled collection of posts and information was lost. The problem with the unbalanced dataset needs to be addressed in a different manner and to find new ways of augmenting the positive examples or to collect new examples altogether.

5. Deep learning approach

For the final implementation of the system, a single deep learning model was used. We started by implementing transformer based systems, with a focus on pretrained BERT based models such as BERT-uncased [4] RoBERTa [5] and ALBERT [6]. Unfortunately, these models were too powerful for the available hardware and it proved to be a real challenge to transfer learning (e.g. the training time for a single epoch for a RoBERTA based model was around 1 hour and for a BERT model was around 3 hours). Due to this issue, we have decided to implement the simple transformer that is very similar to the one presented in the Transformer paper[7].

6. Results

When using deep learning models, the best outcome was obtained by a model containing a hidden dimension of 128 of the linear projection, an embedding size of 64 tokens, and four attention heads. The model achieved an **F1 Score** of **88.8%** and a loss of 0.218 after training for 56 epochs. One significant observation is that all the models used a single transformer block. One significant finding is that when more than one transformer block was used, the score dropped to 0%.

7. Conclusions

In conclusion, the problem of assessing the risk of a user for developing a gambling addiction is in strong need for further research. The particularities of the dataset need to be addressed individually, like the high occurrences of web links, emoji symbols or misspelled words. Also, the problem could benefit from collecting new positive data points in order to reduce the data imbalance. Further research could be directed towards including other points of interest, and not only relying on pure text, such as frequency of posting and time of day with respect to the timezone of the user.

References

- [1] A.-M. Bucur, A. Cosma, L. P. Dinu, Early Risk Detection of Pathological Gambling, Self-Harm and Depression Using BERT, arXiv:2106.16175 [cs] (2021). URL: <http://arxiv.org/abs/2106.16175>, arXiv: 2106.16175 version: 1.
- [2] D. Maupomé, M. D. Armstrong, F. Rancourt, T. Soulas, Early Detection of Signs of Pathological Gambling, Self-Harm and Depression through Topic Extraction and Neural Networks (????) 15.
- [3] A. B. Dieng, F. J. R. Ruiz, D. M. Blei, Topic Modeling in Embedding Spaces, 2019. URL: <http://arxiv.org/abs/1907.04907>, number: arXiv:1907.04907 arXiv:1907.04907 [cs, stat].
- [4] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, Technical Report arXiv:1810.04805, arXiv, 2019. URL: <http://arxiv.org/abs/1810.04805>. doi:10.48550/arXiv.1810.04805, arXiv:1810.04805 [cs] type: article.
- [5] 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, Technical Report arXiv:1907.11692, arXiv, 2019. URL: <http://arxiv.org/abs/1907.11692>. doi:10.48550/arXiv.1907.11692, arXiv:1907.11692 [cs] type: article.
- [6] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, R. Soricut, ALBERT: A Lite BERT for Self-supervised Learning of Language Representations, Technical Report arXiv:1909.11942, arXiv, 2020. URL: <http://arxiv.org/abs/1909.11942>. doi:10.48550/arXiv.1909.11942, arXiv:1909.11942 [cs] type: article.
- [7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention Is All You Need, Technical Report arXiv:1706.03762, arXiv, 2017. URL: <http://arxiv.org/abs/1706.03762>. doi:10.48550/arXiv.1706.03762, arXiv:1706.03762 [cs] type: article.

A. Machine Learning Experiments

We present additional ML experiments with different values for feature extraction.

Table 4

Additional results for the ML approach in the second run, on the balanced dataset.

| ML + feature extractor | Precision | Recall | F1 |
|--|------------------|---------------|-------------|
| LinearSVC + BOW | 0.83 | 0.92 | 0.87 |
| LinearSVC + BOW (min_df=0.2) | 0.75 | 0.93 | 0.83 |
| LinearSVC + BOW (max_df=0.8) | 0.83 | 0.92 | 0.87 |
| LinearSVC + BOW (n_gram=2) | 0.95 | 0.95 | 0.82 |
| LinearSVC + BOW (n_gram=3) | 0.58 | 1.00 | 0.74 |
| LinearSVC + TF-IDF | 0.86 | 0.89 | 0.88 |
| LinearSVC + TF-IDF (min_df=0.2) | 0.77 | 0.78 | 0.77 |
| LinearSVC + TF-IDF (max_df=0.8) | 0.86 | 0.89 | 0.88 |
| LinearSVC + TF-IDF (n_gram=2) | 0.85 | 0.87 | 0.86 |
| LinearSVC + TF-IDF (n_gram=3) | 0.76 | 0.91 | 0.83 |
| LogisticRegression + BOW | 0.87 | 0.96 | 0.91 |
| LogisticRegression + BOW (min_df=0.2) | 0.76 | 0.91 | 0.83 |
| LogisticRegression + BOW (max_df=0.8) | 0.87 | 0.96 | 0.91 |
| LogisticRegression + BOW (n_gram=2) | 0.81 | 0.98 | 0.89 |
| LogisticRegression + BOW (n_gram=3) | 0.61 | 1.0 | 0.75 |
| LogisticRegression + TF-IDF | 0.87 | 0.92 | 0.89 |
| LogisticRegression + TF-IDF (min_df=0.2) | 0.78 | 0.78 | 0.78 |
| LogisticRegression + TF-IDF (max_df=0.8) | 0.87 | 0.92 | 0.89 |
| LogisticRegression + TF-IDF (n_gram=2) | 0.83 | 0.9 | 0.86 |
| LogisticRegression + TF-IDF (n_gram=3) | 0.71 | 0.95 | 0.81 |
| RandomForest + BOW | 0.89 | 0.89 | 0.89 |
| RandomForest + BOW (min_df=0.2) | 0.77 | 0.89 | 0.82 |
| RandomForest + BOW (max_df=0.8) | 0.89 | 0.89 | 0.89 |
| RandomForest + BOW (n_gram=2) | 0.99 | 0.41 | 0.58 |
| RandomForest + BOW (n_gram=3) | 0.6 | 0.02 | 0.05 |
| RandomForest + TF-IDF | 0.91 | 0.89 | 0.9 |
| RandomForest + TF-IDF (min_df=0.2) | 0.77 | 0.88 | 0.82 |
| RandomForest + TF-IDF (max_df=0.8) | 0.91 | 0.89 | 0.9 |
| KNeighbors + BOW | 0.63 | 0.87 | 0.73 |
| DecisionTree + BOW | 0.84 | 0.85 | 0.85 |
| DecisionTree + BOW (min_df=0.2) | 0.73 | 0.87 | 0.8 |
| DecisionTree + BOW (max_df=0.8) | 0.87 | 0.87 | 0.87 |
| DecisionTree + TF-IDF | 0.9 | 0.91 | 0.91 |
| DecisionTree + TF-IDF (min_df=0.2) | 0.73 | 0.88 | 0.8 |
| DecisionTree + TF-IDF (max_df=0.8) | 0.9 | 0.83 | 0.86 |

Table 5

Additional results for the ML approach in the third run, on the balanced dataset with discarded posts.

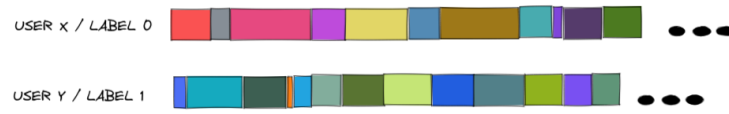
| ML + feature extractor | Precision | Recall | F1 |
|--|------------------|---------------|-------------|
| LinearSVC + BOW | 0.78 | 0.62 | 0.69 |
| LinearSVC + BOW (max_df=0.8) | 0.78 | 0.62 | 0.69 |
| LinearSVC + BOW (n_gram=2) | 0.77 | 0.47 | 0.58 |
| LinearSVC + BOW (n_gram=3) | 0.86 | 0.12 | 0.21 |
| LinearSVC + TF-IDF | 0.78 | 0.66 | 0.71 |
| LinearSVC + TF-IDF (max_df=0.8) | 0.78 | 0.66 | 0.71 |
| LinearSVC + TF-IDF (n_gram=2) | 0.77 | 0.54 | 0.64 |
| LinearSVC + TF-IDF (n_gram=3) | 0.82 | 0.26 | 0.39 |
| LogisticRegression + BOW | 0.8 | 0.62 | 0.7 |
| LogisticRegression + BOW (max_df=0.8) | 0.8 | 0.62 | 0.7 |
| LogisticRegression + BOW (n_gram=2) | 0.8 | 0.49 | 0.6 |
| LogisticRegression + BOW (n_gram=3) | 0.87 | 0.16 | 0.27 |
| LogisticRegression + TF-IDF | 0.79 | 0.66 | 0.72 |
| LogisticRegression + TF-IDF (max_df=0.8) | 0.79 | 0.66 | 0.72 |
| LogisticRegression + TF-IDF (n_gram=2) | 0.77 | 0.54 | 0.64 |
| LogisticRegression + TF-IDF (n_gram=3) | 0.8 | 0.32 | 0.46 |
| KNeighbors + BOW | 0.67 | 0.55 | 0.6 |
| KNeighbors + BOW (max_df=0.8) | 0.67 | 0.55 | 0.6 |
| KNeighbors + BOW (n_gram=2) | 0.63 | 0.39 | 0.41 |
| KNeighbors + BOW (n_gram=3) | 0.9 | 0.02 | 0.04 |
| KNeighbors + TF-IDF | 0.58 | 0.22 | 0.32 |
| KNeighbors + TF-IDF (max_df=0.8) | 0.58 | 0.22 | 0.32 |
| KNeighbors + TF-IDF (n_gram=2) | 0.63 | 0.16 | 0.25 |
| KNeighbors + TF-IDF (n_gram=3) | 0.73 | 0.32 | 0.24 |
| DecisionTree + BOW | 0.7 | 0.63 | 0.67 |
| DecisionTree + BOW (max_df=0.8) | 0.7 | 0.64 | 0.67 |
| DecisionTree + BOW (n_gram=2) | 0.75 | 0.47 | 0.58 |
| DecisionTree + TF-IDF | 0.7 | 0.65 | 0.68 |

Table 6

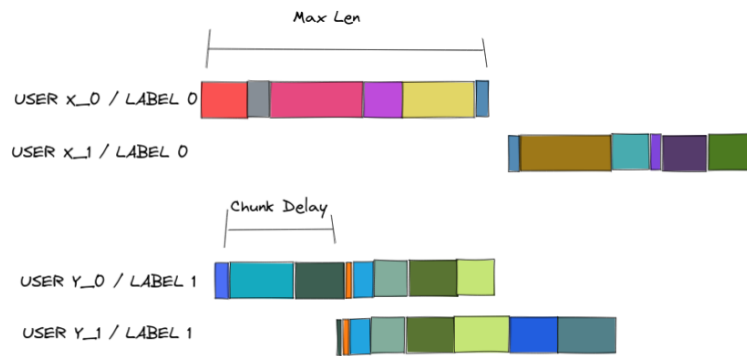
Additional results for the ML approach in the first run, on the unbalanced dataset.

| ML + feature extractor | Precision | Recall | F1 |
|--|------------------|---------------|-------------|
| LinearSVC + BOW | 0.82 | 0.88 | 0.85 |
| LinearSVC + BOW (min_df=0.2) | 0.79 | 0.85 | 0.82 |
| LinearSVC + BOW (max_df=0.8) | 0.83 | 0.9 | 0.86 |
| LinearSVC + BOW (n_gram=2) | 0.79 | 0.68 | 0.73 |
| LinearSVC + BOW (n_gram=3) | 0.98 | 0.15 | 0.25 |
| LinearSVC + TF-IDF | 1.0 | 0.81 | 0.89 |
| LinearSVC + TF-IDF (min_df=0.2) | 0.99 | 0.83 | 0.9 |
| LinearSVC + TF-IDF (max_df=0.8) | 1.0 | 0.81 | 0.89 |
| LinearSVC + TF-IDF (n_gram=2) | 0.99 | 0.51 | 0.67 |
| LinearSVC + TF-IDF (n_gram=3) | 0.8 | 0.07 | 0.13 |
| LogisticRegression + BOW | 0.85 | 0.87 | 0.86 |
| LogisticRegression + BOW (min_df=0.2) | 0.83 | 0.85 | 0.84 |
| LogisticRegression + BOW (max_df=0.8) | 0.87 | 0.87 | 0.87 |
| LogisticRegression + BOW (n_gram=2) | 0.89 | 0.61 | 0.72 |
| LogisticRegression + BOW (n_gram=3) | 1.00 | 0.11 | 0.19 |
| LogisticRegression + TF-IDF | 1.0 | 0.67 | 0.80 |
| LogisticRegression + TF-IDF (min_df=0.2) | 1.00 | 0.63 | 0.77 |
| LogisticRegression + TF-IDF (max_df=0.8) | 1.0 | 0.67 | 0.8 |
| LogisticRegression + TF-IDF (n_gram=2) | 0.8 | 0.05 | 0.1 |
| LogisticRegression + TF-IDF (n_gram=3) | 0.0 | 0.0 | 0.0 |
| RandomForest + BOW | 1.00 | 0.29 | 0.45 |
| RandomForest + BOW (min_df=0.2) | 1.00 | 0.46 | 0.63 |
| RandomForest + BOW (max_df=0.8) | 1.00 | 0.29 | 0.44 |
| RandomForest + BOW (n_gram=2) | 0.99 | 0.41 | 0.58 |
| RandomForest + BOW (n_gram=3) | 0.6 | 0.02 | 0.05 |
| RandomForest + TF-IDF | 1.0 | 0.67 | 0.57 |
| RandomForest + TF-IDF (min_df=0.2) | 1.00 | 0.57 | 0.72 |
| RandomForest + TF-IDF (max_df=0.8) | 1.0 | 0.34 | 0.5 |
| RandomForest + TF-IDF (n_gram=2) | 1.0 | 0.42 | 0.59 |
| RandomForest + TF-IDF (n_gram=3) | 0.6 | 0.04 | 0.08 |
| KNeighbors + BOW | 0.96 | 0.63 | 0.76 |
| KNeighbors + BOW (min_df=0.2) | 0.94 | 0.6 | 0.74 |
| KNeighbors + BOW (max_df=0.8) | 0.96 | 0.73 | 0.83 |
| KNeighbors + BOW (n_gram=2) | 0.91 | 0.12 | 0.2 |
| KNeighbors + BOW (n_gram=3) | 0.0 | 0.0 | 0.0 |
| KNeighbors + TF-IDF | 0.94 | 0.67 | 0.78 |
| KNeighbors + TF-IDF (min_df=0.2) | 0.9 | 0.76 | 0.82 |
| KNeighbors + TF-IDF (max_df=0.8) | 0.98 | 0.61 | 0.75 |
| KNeighbors + TF-IDF (n_gram=2) | 0.0 | 0.0 | 0.0 |
| KNeighbors + TF-IDF (n_gram=3) | 0.0 | 0.0 | 0.0 |
| DecisionTree + BOW | 0.84 | 0.85 | 0.85 |
| DecisionTree + BOW (min_df=0.2) | 0.67 | 0.69 | 0.68 |
| DecisionTree + BOW (max_df=0.8) | 0.86 | 0.86 | 0.86 |
| DecisionTree + BOW (n_gram=2) | 0.88 | 0.82 | 0.85 |
| DecisionTree + BOW (n_gram=3) | 0.71 | 0.45 | 0.55 |
| DecisionTree + TF-IDF | 0.9 | 0.91 | 0.91 |
| DecisionTree + TF-IDF (min_df=0.2) | 0.75 | 0.73 | 0.74 |
| DecisionTree + TF-IDF (max_df=0.8) | 0.87 | 0.88 | 0.87 |
| DecisionTree + TF-IDF (n_gram=2) | 0.87 | 0.86 | 0.86 |
| DecisionTree + TF-IDF (n_gram=3) | 0.51 | 0.63 | 0.56 |

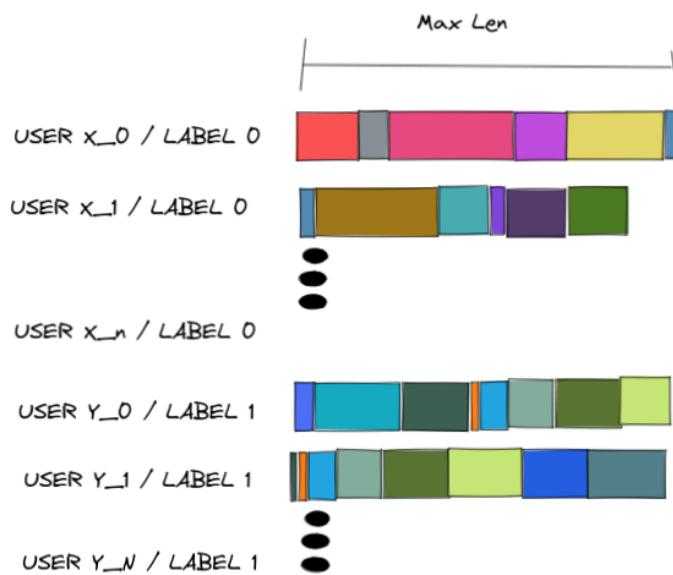
B. Chunking visualization



(a) All the aggregated posts for each user.



(b) Depending on the label, the posts are split to a maximum length. The posts of users with label 1, contain information from the previous post chunk, depending on the overlap.



(c) Final result of the chunking algorithm

Figure 1: A visual explanation of the algorithm described in subsection 3.1

C. Deep Learning Grid Search

Table 7
Deep Learning Grid Search Results

| Hidden Dimension | Embedding Size | Num. Att. heads | Trained Epochs | F1 score | Loss | |
|------------------|----------------|-----------------|----------------|--------------|-------|-------|
| 64 | 64 | 1 | 14 | 0.705 | 0.452 | |
| | | | 19 | 0.479 | 0.514 | |
| | | | 11 | 0.536 | 0.531 | |
| | | | 21 | 0.232 | 0.763 | |
| | | 4 | 41 | 0.872 | 0.228 | |
| | | | 36 | 0.850 | 0.259 | |
| | | | 22 | 0.848 | 0.262 | |
| | | | 15 | 0.843 | 0.287 | |
| | | | 43 | 0.851 | 0.289 | |
| | | | 29 | 0.859 | 0.336 | |
| | 128 | 1 | 1 | 33 | 0.813 | 0.403 |
| | | | | 10 | 0.537 | 0.559 |
| | | | 4 | 12 | 0.831 | 0.298 |
| | | | | 40 | 0.705 | 0.436 |
| 128 | 64 | 1 | 31 | 0.690 | 0.465 | |
| | | | 10 | 0.093 | 0.632 | |
| | | | 16 | 0.449 | 0.729 | |
| | | | 6 | 0.730 | 0.390 | |
| | | 2 | 56 | 0.888 | 0.218 | |
| | | | 18 | 0.816 | 0.360 | |
| | | 128 | 2 | 24 | 0.827 | 0.275 |
| | | | | 16 | 0.841 | 0.307 |
| | | | 4 | 40 | 0.859 | 0.257 |
| | | | | 27 | 0.858 | 0.257 |
| | 14 | 0.743 | 0.440 | | | |