DSTI 2024 - DEEPLEARNING PROJECT

Training a Covid Fakenews Detector

COSTA L., ROMEO L., LEN N., HERBET A.

1. Introduction

As Saif Abed wrote it in January 2024 [1] for WHO, “we’re not just fighting an epidemic; we’re fighting an infodemic. Fake news spreads faster and more easily than this virus and is just as dangerous”. The effects of this infodemic can but tragic. In UK, “innacurate information on mask-wearing between April and November 2020 was associated with 21947 additional COVID-19 cases, 2187 hospitalizations and 509 deaths “[2]. In USA, “vaccination rated dropped from 73,8% to 52,2% when people was exposed to misinformation“[3]. Help detecting misinformation could then be a good way to help. We decided to go for a text-classification problem and try to solve it with a NLP model that we would train.

a. Objective

The purpose of this project is to build an effective deep learning model for detecting COVID-19-related fake news. The goal is to accurately classify information as either real or fake, thereby combating misinformation, which can have significant societal consequences.

With the overwhelming spread of false information during the COVID-19 pandemic, deep learning presents a powerful solution for automating the detection of such content by leveraging large datasets and advanced language models.

b. Context

The COVID-19 pandemic led to a global health crisis that was exacerbated by a parallel "infodemic"—the rapid spread of misinformation and disinformation across various digital platforms. This misinformation, ranging from false medical advice to conspiracy theories, often caused public confusion, undermined health measures, and in some cases, led to harmful behavior. Traditional methods of curbing misinformation, such as manual fact-checking, are resource-intensive and slow.

As a result, there is an increasing need for automated systems that can detect and flag fake news in real-time. Deep learning, with its ability to process and analyze large amounts of textual data, is well-suited for this task, enabling more accurate and scalable solutions.

c. Problem statement

The primary problem addressed by this project is to classify individual pieces of information or sentences related to COVID-19 as real or fake using deep learning techniques. The challenge lies in designing a model that can learn to recognize the linguistic cues and contextual differences that are indicative of fake news, ensuring high accuracy and generalizability to new, unseen data. This will enable more effective and scalable detection of misinformation, contributing to efforts in mitigating its harmful effects.

2. Literrature review

Li and his team described in 2020 [4] the whole evolution from shallow model to Deep learning. In the literature shadow models not only contain neural networks with only 1 hidden layer but also classic models like K-Nearest Neighbors, Support Vector Machine or Random Forest. Figure 1 shows the classic pipeline for text classification.

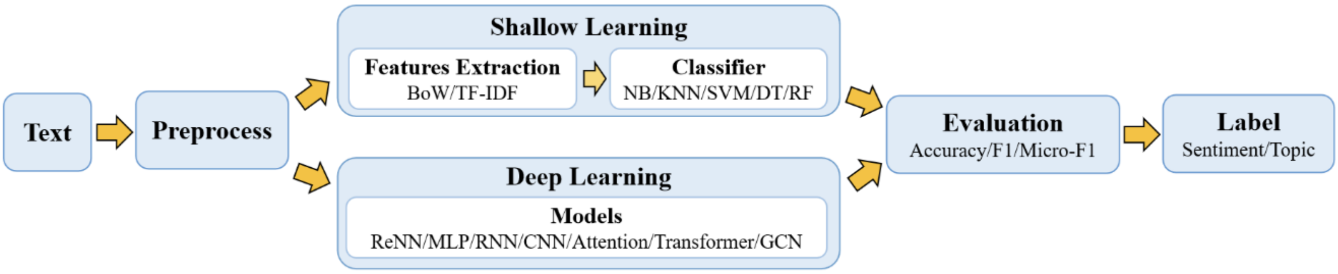


Figure 1. Flowchart of the text classification with classic methods in each module. It is crucial to extract essential features for shallow models, but features can be extracted automatically by DNNs

Figure 2 shows that Deeplearning Neural Networks started to appear in 2011 with RAE and rapidly some other models like LSTM or TextRCNN arrived. Transformers appeared in 2017 with attention mechanisms [5] and changed many things. From this moent arrived ELMo, OpenAI GPT and BERT. BERT-based models were used for text-classification not because they were designed for but because they were considered more effective than others on that type of problem.

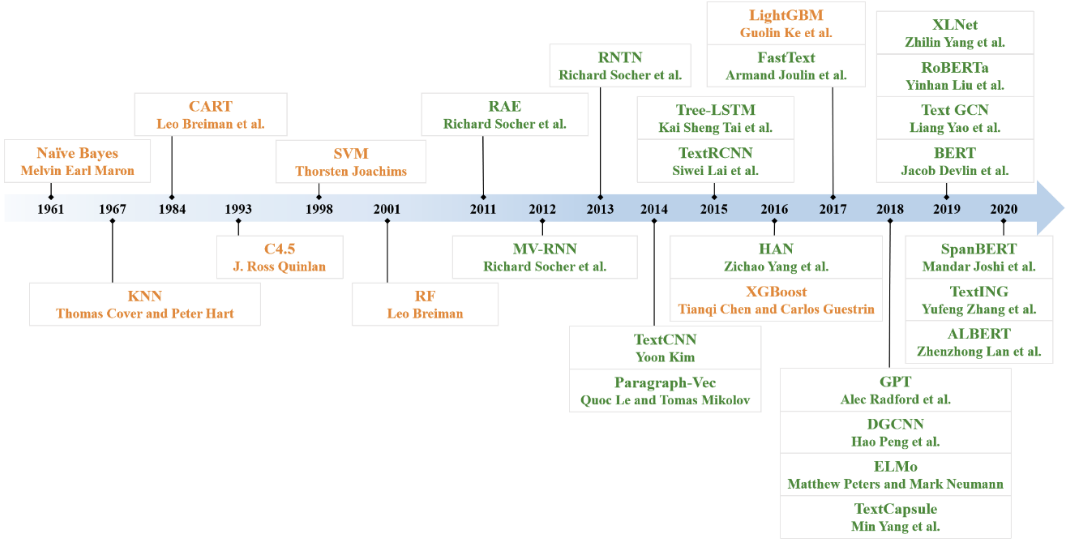


Figure 2. Schematic illustration of the primary text classification methods from 1961 to 2020. Before 2010, almost all existing methods are based on shallow models (orange color); since 2010, most work in this area has concentrated on deep learning schemes (green color).

Another important subject is datasets. We can't train or retrain a model without labeled datasets. To choose a good dataset for our subject we need to define clearly what kind of text-classification we are dealing with. We want to encode sentences about COVID-19 and define if this sentence is a Fake News of not.

Gasparetto and al. [6] defined different text classification tasks:

- Sentiment analysis (SA): the task of understanding affective states and subjective information contained in a piece of text (stirred emotions)
- Topic labelling (TL): the task of recognising one or more themes for a piece of text
- News classification (NC): the task of assigning categories to news pieces
- Question answering (QA): the task of selecting an answer to a question, selecting from potential candidate sentences (usually extracted from a context document). This task is usually framed as binary or multiclass classification
- Natural language inference (NLI): the task of determining whether two sentences entail one another
- Named entity recognition (NER): the task of locating named entities within unstructured text, labelling them with predefined categories
- Syntactic parsing (SP): a series of tasks related to predicting morpho-syntactic properties of words, such as part-of-speech (PoS) tagging, speech dependencies and semantic role labelling.

Our task is a News Classification (NC).

Another crucial point is the evaluation metric. Lu [7] and Padalko [8] only use general metrics for their fakenews detector:

- A confusion matrix to see True Positives, False Positives, True Negatives and False Negatives.
- Calculated metrics: accuracy, precision, recall, F1-score, ROC curve and AUC.

There are no specific metrics like BLUE (Bilingual Evaluation Understudy Score), ROUGE (Recall Oriented Understudy for Gisting Evaluation), METEOR (Metric for Evaluation of Translation with Explicit Ordering) or perplexity that are used in text generation in translation.

Xi and Zhang [9] already trained a COVID-19 fake news detector using attention-based transformer in 2021. They managed to have a 75,9 accuracy and an AUC of 0.774.

3. Experiments

a. Datasets

This project utilized two datasets related to COVID-19 news:

- COVID-19-FNIR Dataset:

The COVID-19 Fake News Infodemic Research Dataset consists of true and fake news articles, with a total of 7,588 items. The dataset is class-balanced, with 49.99% of the items labeled as real and 50.01% as fake. The fake news was collected from PolitiFact, while the real news was sourced from verified Twitter handles of authentic news publishers. The dataset contains various columns, including Text, Date, Region, Country, Explanation, Origin, and Label, but only the Text and Label columns were used in this study.

- COVID-19 Fake News Detection Dataset:

This dataset includes a collection of 10,700 social media posts related to COVID-19, with labels indicating whether the news is real or fake. Real news items were gathered from verified news sources, while fake items were collected from fact-checking platforms such as NewsChecker and PolitiFact. The dataset is also balanced, with 52.34% of the posts labeled as real and 47.66% as fake.

For the purpose of this project, a combined corpus was created from both datasets. Column names were standardized, and the labels were adjusted to ensure consistency (replacing 'fake' with 0 and 'real' with 1). The combined dataset consists of 18,288 news items, with 51% labeled as real and 49% as fake. This balanced dataset was then split into training and testing sets for the comparative analysis.

b. Methodology

We first trained Random Forest and Support Vector machine with a TD-IDF encoding without working a lot on the hyperparameters, which are said to be time consuming by Li and al [4], to have an idea of their performances. We obtained 0.907 accuracy and 0.9 F1-score with Support Vector Machine and 0.905 accuracy and 0.9 F1-score with Random Forest. We then had a baseline to begin with deeplearning [Y].

Most of the autors use more a less the same pipeline described by Xia and al. In 2023 [10]:

- Prepare the data: splitting, EDA,...
- Pre-processing: segmentation (with jieba), stop words to make disappear all unrelevant words (the, and, him...), lemmatization or stemming [11], text alignment, vectorization, embedding...
- Prepare a model to finetune
- Evaluate the model
- Analyze metrics

As we were in a Deep learning project, we focused on the 3 last steps.

We then wrote a whole code working on our text classification task. We began with RoBERTa for both encoding and decoding part and got 0.882 accuracy. We chose RoBERTa because BERT-based models are known to be good models on text classification task. They are also considered as “small models”. We decided to try several encoders to see which one was the most adapted to improve the model. We tried Vertex embedding which was improving the model a lot but we couldn’t use it freely, so we decided to go on other embedding models. We looked at the open large language model leaderboard to test some embedders for text classification in the top 50 and had the best results with Stella [12] with 0.945 accuracy.

We now had the model architecture with the metrics recommended, the embedder and we just had to finetune the model. We focused on many parameters of the neural network in the ‘simpleNN’ function:

- Number of hidden layers: we began with 2 hidden layers and finaly ended with 3 hidden layers and 1 output layer
- Number of neurons in the hidden layers: at the beginning we had 256 neurons and decreased to 2 neurons in the last layer. We had better results doubling the number of neurons in comparison to

the input size of Stella. We then established that it was better to drop the number of neurons we just 1 hidden layer (1536-3072-3072-768-2) instead of decreasing them slowly (1536-3072-1536-768-2)

- Activation function: paradoxically, we had better results with ReLU activation function. We tried to modify them with GELU which is recommended for text classification or with ReLU6 and even SiLU but they didn't improve the model.
- We tried to modify the output layer to a sigmoid as we were in a binary classification task but the results were not improved.
- Drop-out: we used drop-out regularization technique and found that it was best around 0.6
- We used AdamW optimizer and played with its learning rate and weight decay. Weight decay is L2 normalization [W]. We kept in the code the loop to show how we proceeded

We also used in the training function:

- We used cross-validation to make our model more robust to overfitting
- We implemented patience and early stopping to avoid overfitting
- We used weight decay parameter from AdamW optimizer to add L2 regularization.

We selected the best hyperparameters using the training and validation datasets with cross validation. We then evaluated the final performance using training + validation and test datasets.

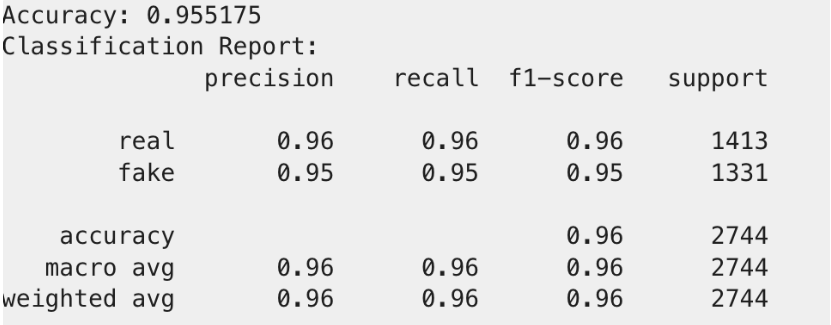
The model's performances were monitored using mainly:

- Training and validation loss curves to look at eventual overfitting,
- Validation accuracy,
- F1-score,
- ROC curve and AUC (Aera Under Curve).

4. Results

a. Our results

We managed to train a model with decent performances we can find in the figure 3:



```
Accuracy: 0.955175
Classification Report:
              precision    recall  f1-score   support

        real       0.96      0.96      0.96      1413
        fake       0.95      0.95      0.95      1331

    accuracy                           0.96      2744
   macro avg       0.96      0.96      0.96      2744
weighted avg       0.96      0.96      0.96      2744
```

Figure 3. Classification report and accuracy of the final model

- An accuracy of 95.5175%

- An F1-score of 95% to detect Fake News and 96% to detect real news.
- An AUC of 0.955124

The Loss curves in figure 4 let us think that the model is not overfitted:

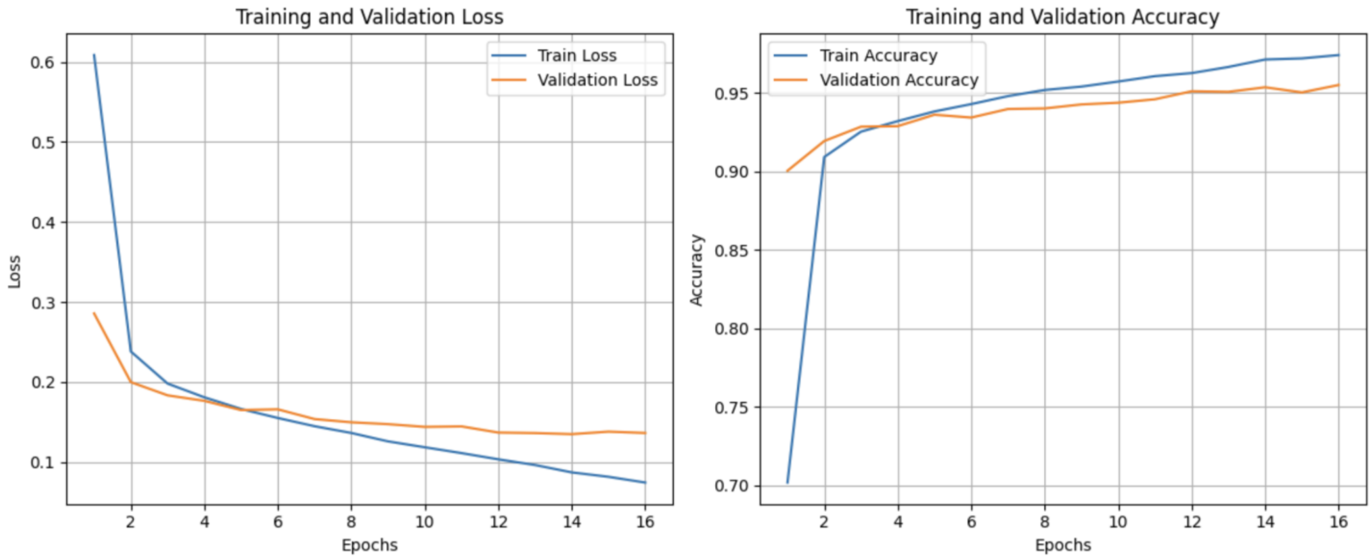


Figure 4 : Training and Validation Losses and Accuracies

We could also calculate from the confusion matrix in figure 5 other metrics like precision (sensibility), recall and specificity:

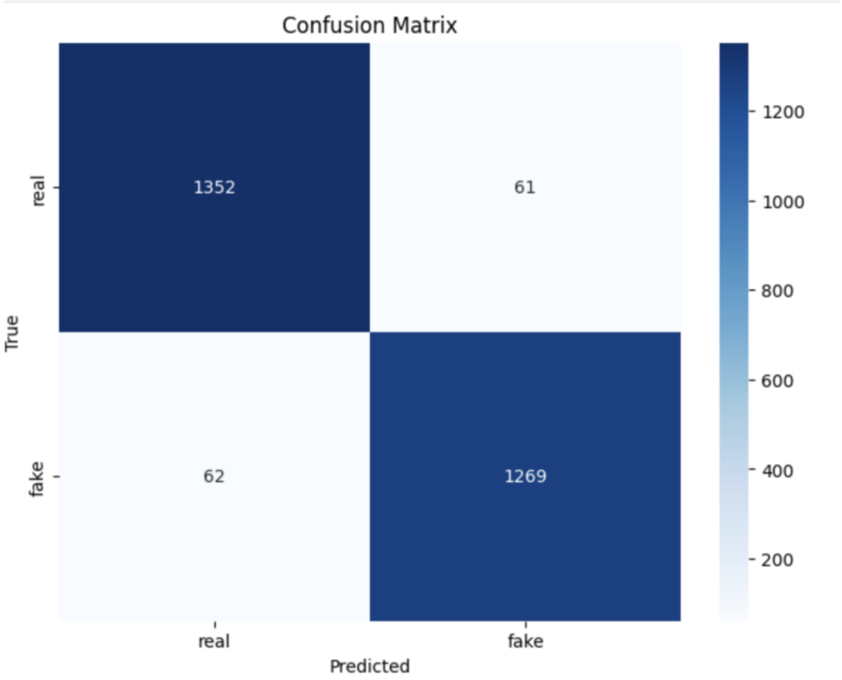


Figure 5 : Confusion Matrix

Precison / Sensibility = 1352 / (1352 + 62) = 0.95615

Recall = 1352 / (1352 + 61) = 0.95683

https://github.com/Anerol18/Fake_News_Detector_NLP_DeepLearning_Project

Specificity = 1269 / (1269 + 61) = 0.95414

We can also show one result obtained by Cross-Validation with the L2 regularization (Weight Decay from AdamW optimizer on figure 6.

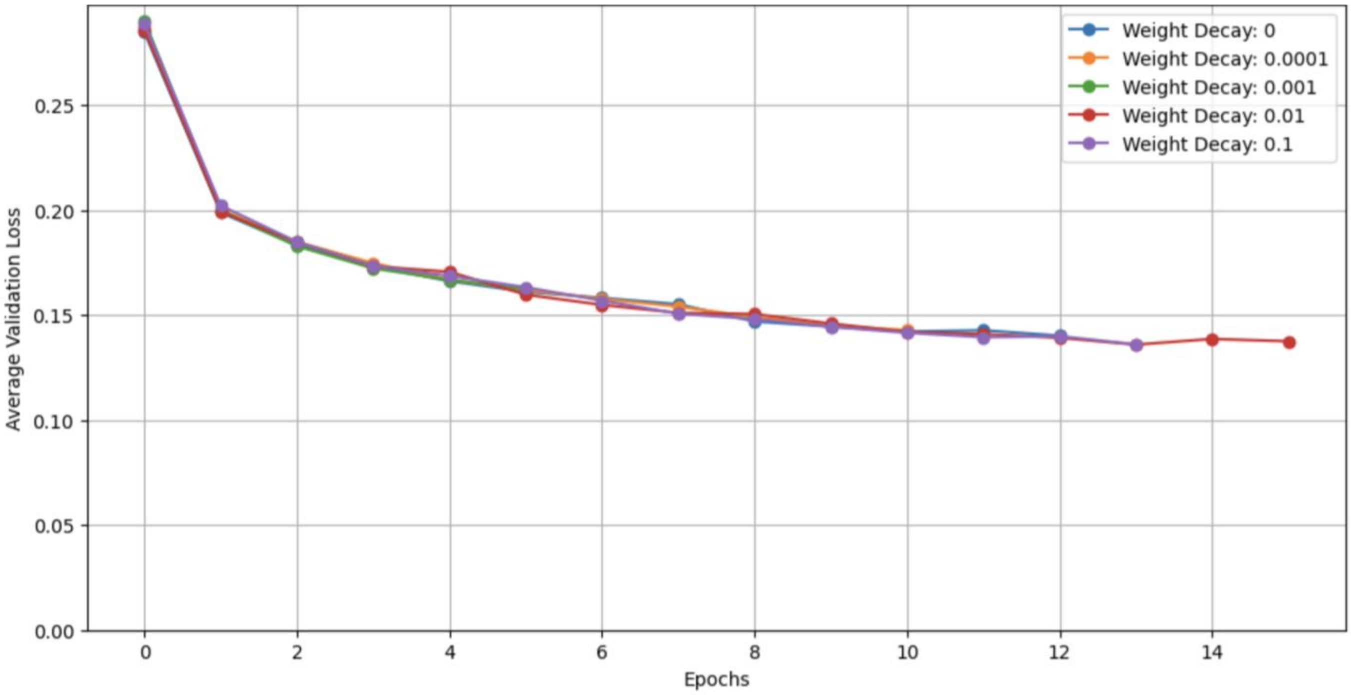


Figure 6 : Cross Validation results trying to optimize L2 regularization

Finally we produced a ROC curve showed in figure 7 which confirms that our model is homogeneous:

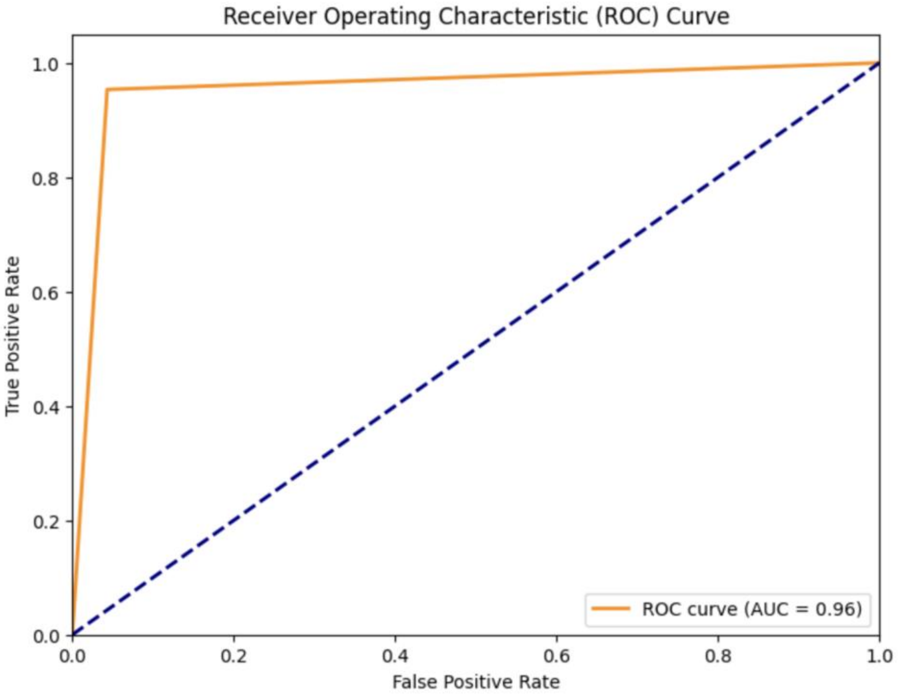


Figure 7: ROC Curve

https://github.com/Anerol18/Fake_News_Detector_NLP_DeepLearning_Project

b. Discussion

We kept in mind that we could have done a better work in the pre-processing step. We had in mind that the main purpose of this project was to manipulate the Deeplearning part. We also thought that LLMs were more resilients to this step than the shallow models. Another point is that pre-processing steps as stemming, lemmatization or stop words can change the context of transformers. Nevertheless, they seem to be important [13]. Figure 8 explains clearly what the 2 firsts are:

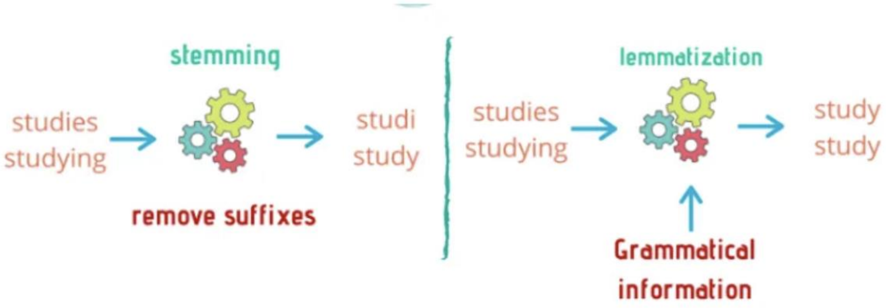


Figure 8. Stemming and Lemmatization

- Stemming consists in removing suffixes to facilitate embedding part,
- Lemmatization is like a standardization or the grammatical information,
- Stop words consists in removing words which are very frequently used but not bringing a lot of sens: “the”, “a”, “an” or “i”. It can also withdraw some meaning and need to be used carefully,
- Punctuation mark removal consists in removing the punctuation in all the text to avoid multiplication of some words: for exemple “her”, “her!”, “her.”, “her,” and “her?”. But it is also a step where we lose some meaning from the context,
- Lower casing consists in transforming all the text in lower case.

We thought about the generalization of the model. We trained on a certain dataset but we can’t be sure that if the model were deployed in real life it would be able to classify correctly any king of fake news about covid. There could be a drift of the model and a drift of the fake news. It should be necessary to monitor the model’s performances on the fly.

We tried a very simple architecture for our model. It was already hard to understand all the notions for such a specific task. It might take a lot more time to try to use more advanced architectures / mixed models with CNNs, NLPs and other parts like Xia’s [10].

Talking about architecture, we could also have worked with multihead attention mechanisms more or less with [14] mask but it would need to have a better comprehension of this topic.

We used the L2 regularization included in the weight decay argument of AdamW optimizer. We could have written our own regularization function including L1 regularization and even a combination of L1 and L2 (elasticnet) [15].

We tried a sigmoid activation function for our binary output but got strangely very bad results with it. We had the same kind of troubles with the criterion. Theoritically we should have used BinaryCrossEntropy loss function but we had a very poor learning with it so we stayed with the CrossEntropyLoss one.

5. Conclusion

We finally learnt a lot about Deeplearning and NLP for text classification tasks. We also discovered a very large and complex universe where knowledge can change very quickly.

It also made us confront the difficulty to work in group on a big project. Scrum and agility are difficult to organize when nobody clearly know how to proceed. Nevertheless, we managed to adapt in regard to each problem we faced which is the Agile way.

At the end we trained a model which fit well to our data, better than the model of Xi and Zhang [9] from 2021. Another proof that everything is going quick in this universe.

At the very end of the project we had some trouble with google colab where flash-attn package was not callable with T4 GPU from one day to the next. Hopefully one of us managed 2 days before the deadline to run the code from A100 GPU to have our final results.

To conclude, as Zhukov described it [16], the choice of the model really depends on the use case. Even shallow models can perform well depending on the situation. It is our job as data scientists to choose, configure and deploy the best well-suited model for a given use case considering the time we can spend on it.

6. Bibliography

[1] Abed S, Allain-Ioos S., Shindo N., Understanding disinformation in the context of public health emergencies: the case of COVID-19, Weekly epidemiological record no4, 26 January 2024.

[2] The cost of lies. Assessing the human and financial impact of COVID-19 related online misinformation on the UK. London: London Economics, December 2020.

[3] Neely SR and al. Vaccine hesitancy and exposure to misinformation: a survey analysis. J Gen Intern Med 2022;37(1):179-87.

[4] Li Q., Peng H. and al. A survey on Text Classification : from shallow to deep learning ; arXiv 2020: arXiv:2008:00364

[5] Vaswani A and al. Attention Is All You Need ; arXiv 2017: ArXiv:1706.03762

[6] Gasparetto A, Marcuzzo M, Zangari A, Albarelli A. A Survey on Text Classification Algorithms: From Text to Predictions. Information. 2022; 13(2):83. https://doi.org/10.3390/info13020083

[7] Lu H. Deep Learning for Fake News Detection: Theories and Models. In Proceedings of the 2022 6th International Conference on Electronic Information Technology and Computer Engineering (EITCE '22). Association for Computing Machinery, New York, NY, USA, 1322–1326. https://doi.org/10.1145/3573428.3573663

[8] Padalko H. and al. Misinformation Detection in Political News using BERT Model. ProfIT AI 2023 Nov 20-22, 2023, Waterloo, Canada.

[9] J. Xi and C. Zhang, "COVID-19 fake news detection using attention-based transformer," ICMLCA 2021; 2nd International Conference on Machine Learning and Computer Application, Shenyang, China, 2021, pp. 1-7.

[10] Xia H and al. COVID-19 fake news detection: a hybrid CNN-BiLSTM-AM model, Technological forecasting and sociale change, vol195,2023. https://doi.org/10.1016/j.techfore.2023.122746.

[11] https://www.ibm.com/topics/stemming-lemmatization

[12] https://huggingface.co/dunzhang/stella_en_1.5B_v5

[13] https://ayselaydin.medium.com/2-stemming-lemmatization-in-nlp-text-preprocessing-techniques-adfe4d84ceee

[14] https://medium.com/@wangdk93/multihead-attention-from-scratch-6fd6f99b9651

[15] https://github.com/christianversloot/machine-learning-articles/blob/main/how-to-use-l1-l2-and-elastic-net-regularization-with-pytorch.md

[16] Zhukov V Choosing the best architecture for your text classification task. https://medium.com/toloka/choosing-the-best-architecture-for-your-text-classification-task-aee30ecc7870

https://github.com/Anerol18/Fake_News_Detector_NLP_DeepLearning_Project