Faculté des sciences et des Techniques
Module: Data Mining
Academic Year : 2023/2024

Topic Detection and topic tracking

prepared by :
Bouabd Chaimaa

supervised by :
Pr. Khadija BOUZAACHANE

Le Plan :

1. Introduction
2. Topic Detection
3. Datasets
4. Méthodes
5. Evaluation
6. Implementation
7. Conclusion

1 . Introduction

2 . What is Topic Detection ?

Topic detection, also known as topic modeling or topic analysis, is a natural language processing (NLP) technique used to identify and extract topics or themes from a collection of text documents. The goal is to uncover the main ideas or subjects discussed within the documents without prior knowledge of the specific content. Topic detection is commonly applied to large datasets, such as news articles, social media posts, research papers, or any other text corpora.

The output of topic detection algorithms is a set of topics, each represented by a list of keywords or terms. These topics can help users understand the main themes present in a large collection of text data and can be valuable for tasks such as information retrieval, content recommendation, and trend analysis.

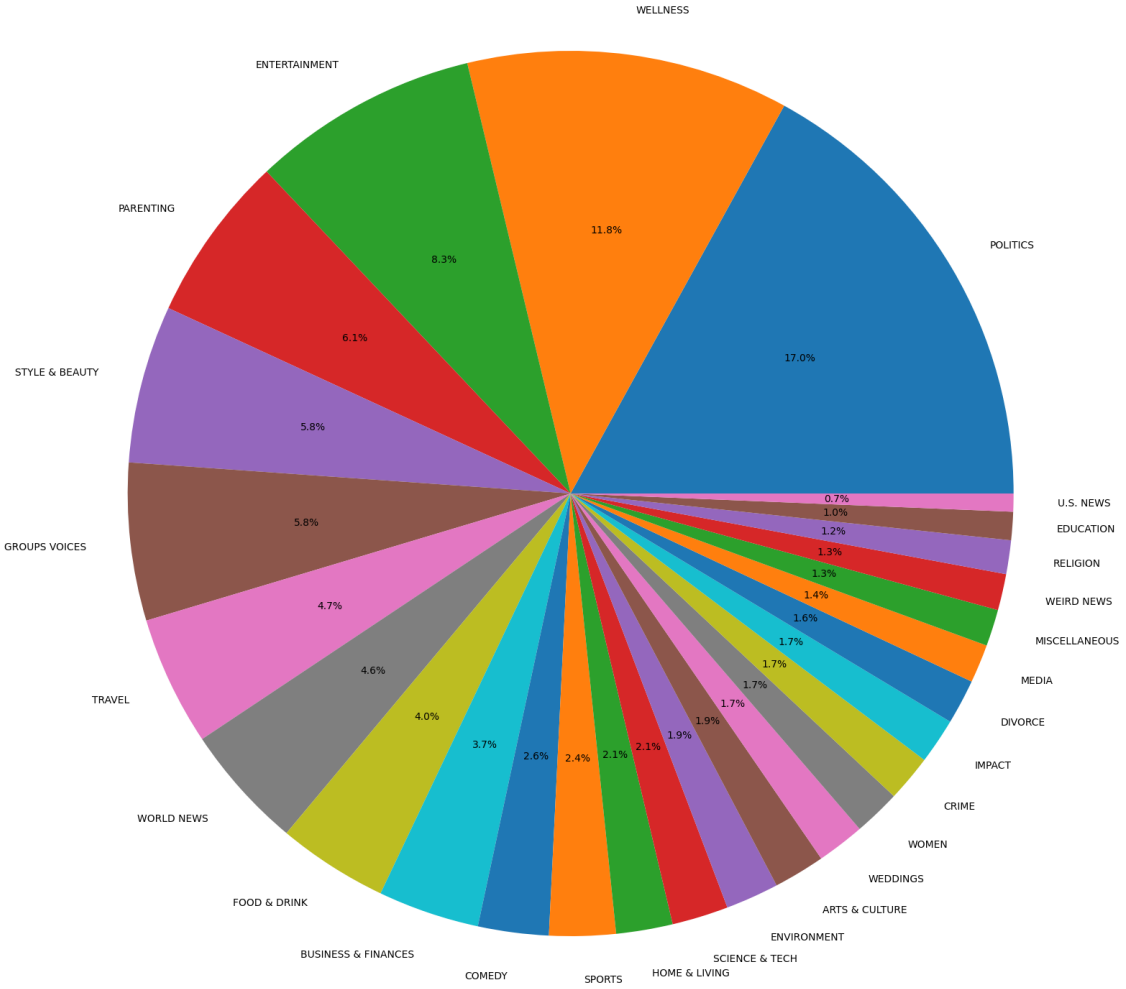
3 . Datasets

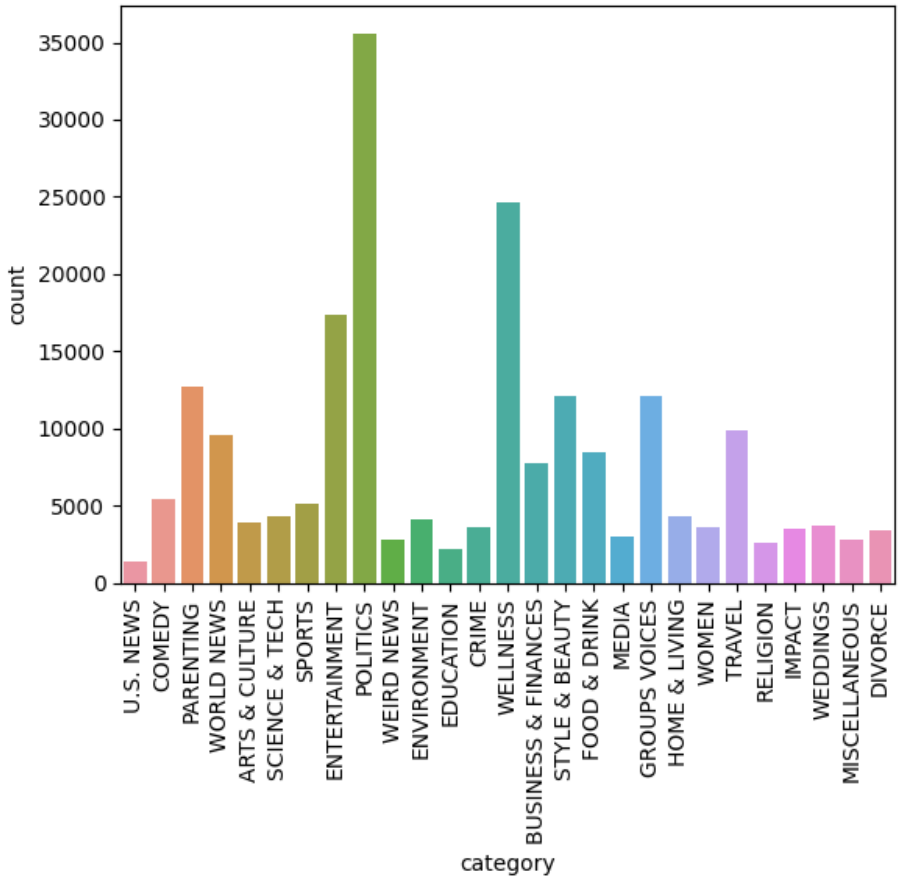
The dataset used in this Project is ***News Category Dataset*** contains around 210k news headlines from 2012 to 2022 from HuffPost. This is one of the biggest news datasets and can serve as a benchmark for a variety of computational linguistic tasks. HuffPost stopped maintaining an extensive archive of news articles sometime after this dataset was first collected in 2018, so it is not possible to collect such a dataset in the present day. Due to changes in the website, there are about 200k headlines between 2012 and May 2018 and 10k headlines between May 2018 and 2022.

Content :

Each record in the dataset consists of the following attributes:

- category: category in which the article was published.
- headline: the headline of the news article.
- authors: list of authors who contributed to the article.
- link: link to the original news article.
- short_description: Abstract of the news article.
- date: publication date of the article.


WELLNESS
ENTERTAINMENT
PARENTING
STYLE & BEAUTY
GROUPS VOICES
TRAVEL
WORLD NEWS
FOOD & DRINK
BUSINESS & FINANCES
COMEDY
SPORTS
HOME & LIVING
SCIENCE & TECH
ENVIRONMENT
ARTS & CULTURE
WEDDINGS
WOMEN
CRIME
IMPACT
DIVORCE
MEDIA
MISCELLANEOUS
WEIRD NEWS
RELIGION
EDUCATION
U.S. NEWS
POLITICS
11.8%
8.3%
6.1%
5.8%
5.8%
4.7%
4.6%
4.0%
3.7%
2.6%
2.4%
2.1%
2.1%
1.9%
1.9%
1.7%
1.7%
1.7%
1.6%
1.4%
1.3%
1.3%
1.2%
1.0%
0.7%
17.0%


count
35000
30000
25000
20000
15000
10000
5000
0
U.S. NEWS
COMEDY
PARENTING
WORLD NEWS
ARTS & CULTURE
SCIENCE & TECH
SPORTS
ENTERTAINMENT
POLITICS
WEIRD NEWS
ENVIRONMENT
EDUCATION
CRIME
WELLNESS
BUSINESS & FINANCES
STYLE & BEAUTY
FOOD & DRINK
MEDIA
GROUPS VOICES
HOME & LIVING
WOMEN
TRAVEL
RELIGION
IMPACT
WEDDINGS
MISCELLANEOUS
DIVORCE
category

4 . Méthodes :

1 . DistilBERT Model :

We chose DistilBERT for finetuning and transfer learning for our task due to several compelling reasons that make it the preferred choice over BERT, RoBERTa, and XLNet, especially considering the limited training time:

1. Faster Inference Speed: DistilBERT offers faster inference speed compared to BERT, RoBERTa, and XLNet. This is a critical advantage for my task, as it involves real-time or low-latency processing. The faster inference speed of DistilBERT ensures that I can obtain results quickly and efficiently.

2. Efficient Resource Utilization: DistilBERT is a smaller network that retains 95% of BERT's language understanding capabilities while using 60% fewer parameters. This parameter efficiency makes it more suitable for tasks with limited computational resources, enabling me to achieve comparable performance with reduced memory and computational requirements.

3. Acceptable Trade-off in Prediction Metrics: While DistilBERT may have a slight compromise on prediction metrics compared to BERT, RoBERTa, and XLNet, the difference in performance is minor. Given the gains in inference speed and resource efficiency, this trade-off is well justified for my task.

4. Transfer Learning Benefits : DistilBERT benefits from pretraining on a large corpus, which provides it with a strong language understanding foundation. This makes it well-suited for transfer learning, allowing me to fine-tune the model on my specific task with potentially less labeled data.

5. Training Time Limitation: Since I have limited training time, DistilBERT's faster training speed compared to BERT, RoBERTa, and XLNet becomes a crucial factor. It allows me to efficiently fine-tune the model within my time constraints while still achieving satisfactory performance.

DistilBERT uses a technique called knowledge distillation to transfer the knowledge from a large BERT model (called the teacher) to a smaller BERT model (called the student). Knowledge distillation is a process of training a student model to mimic the output distributions of a teacher model, using a loss function that measures the similarity between the two models.

DistilBERT uses a triple loss function that combines three objectives:

language modeling, distillation, and cosine-distance. Language modeling is the standard objective for pre-training BERT models, which involves predicting masked tokens in a sentence. Distillation is the objective that measures the difference between the teacher and student models using the Kullback-Leibler divergence between the softened probabilities of the student model and the softened probabilities of the teacher model. Cosine-distance is an additional objective that measures the similarity between the hidden states of the teacher and student models using the cosine similarity. The authors found that adding this objective improved the performance of DistilBERT on downstream tasks.

In DistilBERT, the activation function used is the GELU (Gaussian Error Linear Unit) activation function. It is used in the transformer blocks of the model.The GELU function is smoother than other functions like ReLU (Rectified Linear Unit) and is differentiable at every point. This smoothness helps the model to learn more complex patterns during training.

FORMULA:

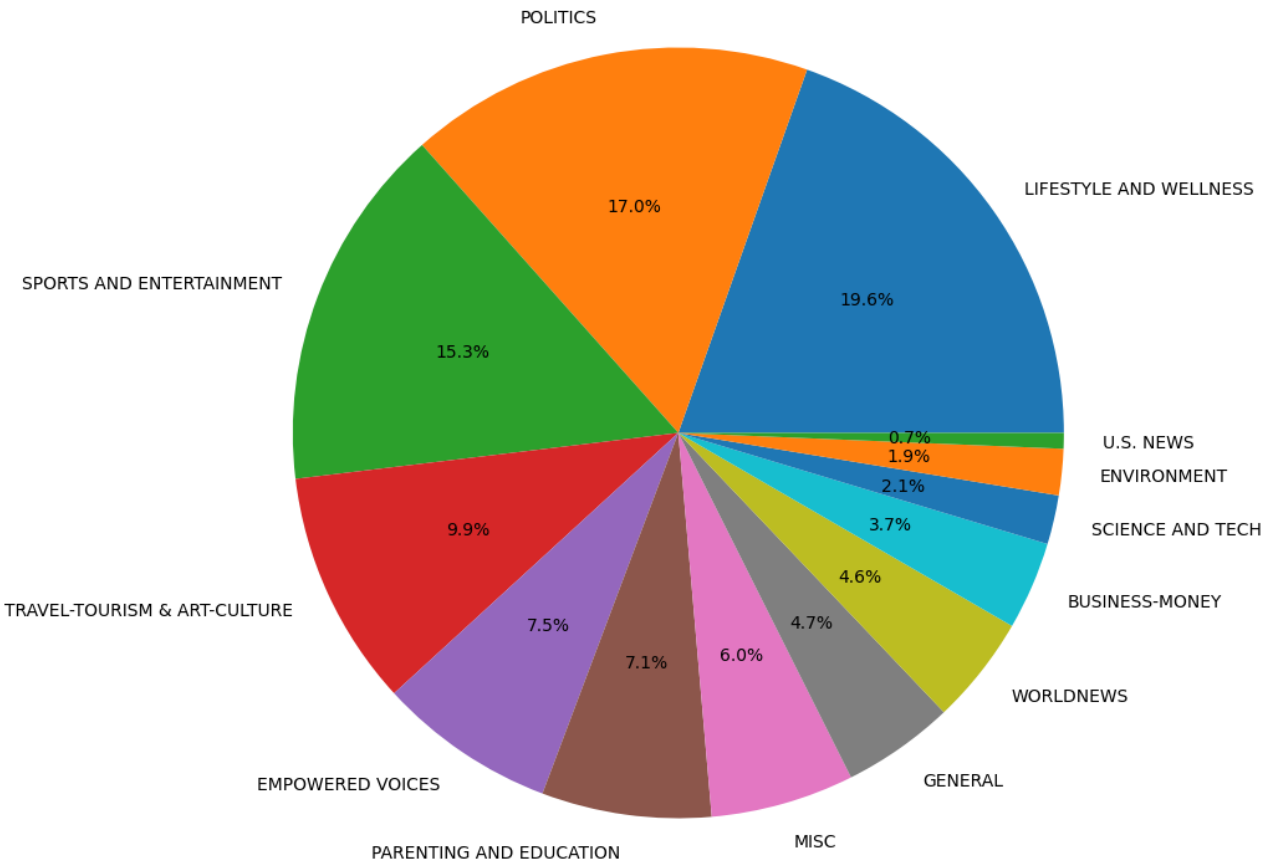
$$f(x) = 0.5 * x * (1 + tanh(\sqrt{2/\pi} * (x + 0.044715 * x^3)))$$

2 .Topic Detection with LSTM Model :

An LSTM (Long Short-Term Memory) model is a type of recurrent neural network (RNN) that is particularly effective in capturing and learning long-term dependencies in sequential data. LSTM networks are widely used in natural language processing (NLP) tasks, including topic detection.

In the context of topic detection, an LSTM model can be employed to learn patterns and relationships within a sequence of words or tokens in a document. The goal is to automatically identify the underlying topic or theme of the document based on its textual content.

We compressed the size of the categories for this model from 42 categories to 13 categories for better practice.



Visualization of the percentage of each category in our datasets.

Let's break down the architecture and steps folowed in our model ：

1. **Load GloVe Embeddings:**
   - Read GloVe embeddings from a pre-trained file (path_to_glove_file).
   - Create an embeddings index (embeddings_index) mapping words to their corresponding embedding vectors.
2. **Create Embedding Matrix:**
   - Initialize an embedding matrix (embedding_matrix) with zeros, where the rows correspond to words in the tokenizer's word index, and columns represent embedding dimensions.
   - Populate the embedding matrix with GloVe vectors for words present in both the tokenizer and GloVe embeddings.
3. **Define Model Architecture:**
   - Create a Sequential model using Keras.
   - Add an Embedding layer with the initialized embedding matrix.
   - Set mask_zero=True to handle padding in input sequences.
   - The Embedding layer is set to non-trainable since it uses pre-trained embeddings.
   - Add a Bidirectional LSTM layer with 256 units and a dropout of 0.4.
   - Add a Dense layer with 13 units and softmax activation for multi-class classification (13 categories).
   - The model summary is displayed to show the architecture.
4. **Compile the Model:**
   - Use the Adam optimizer with a learning rate of 0.001.
   - Categorical crossentropy loss is chosen since this is a multi-class classification problem.
   - The model is set up for accuracy metrics.
5. **Training the Model:**
   - Train the model using the fit method on the training data (train_set, train_label) with validation on the validation set (val_set, val_label).
   - Early stopping is employed with a patience of 3 epochs to stop training if the validation loss does not improve.

The model is designed for text classification, utilizing pre-trained word embeddings to benefit from contextual information encoded in the GloVe vectors. The architecture includes an LSTM layer for sequential data processing and a Dense layer for the final classification.

5 . Evaluation

Measurement of Predictions

Precision, recall, accuracy and F1 score are common metrics used to evaluate the performance of multiclass classification models. Each metric has a different rationale and interpretation.

1. Precision is the ratio of true positives to the total number of predicted positives. It measures how accurate the model is in identifying the correct class for each instance. A high precision means that the model has a low rate of false positives, or misclassifying instances as belonging to a class when they do not.

$$\frac{1}{n}\sum_{i=1}^{n}\frac{TP_i}{TP_i + FP_i}$$

2**. Recall** is the ratio of true positives to the total number of actual positives. It measures how sensitive the model is in detecting all the instances that belong to a class. A high recall means that the model has a low rate of false negatives, or missing instances that should have been classified as belonging to a class.

$$\frac{1}{n}\sum_{i=1}^{n}\frac{TP_i}{TP_i + FN_i}$$

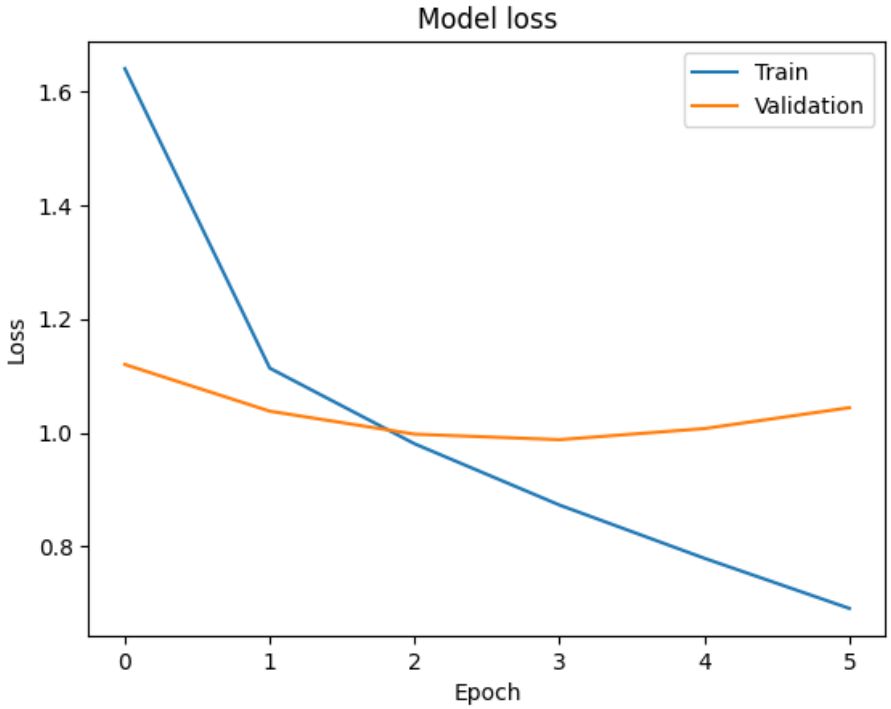
3. Accuracy is the ratio of correct predictions to the total number of predictions. It measures how well the model performs overall, regardless of the class distribution. A high accuracy means that the model has a low rate of errors, or misclassifying instances as belonging to any class.

$$\frac{\sum_{i=1}^{n} TP_i + TN_i}{\sum_{i=1}^{n}(TP_i + TN_i + FP_i + FN_i)}$$

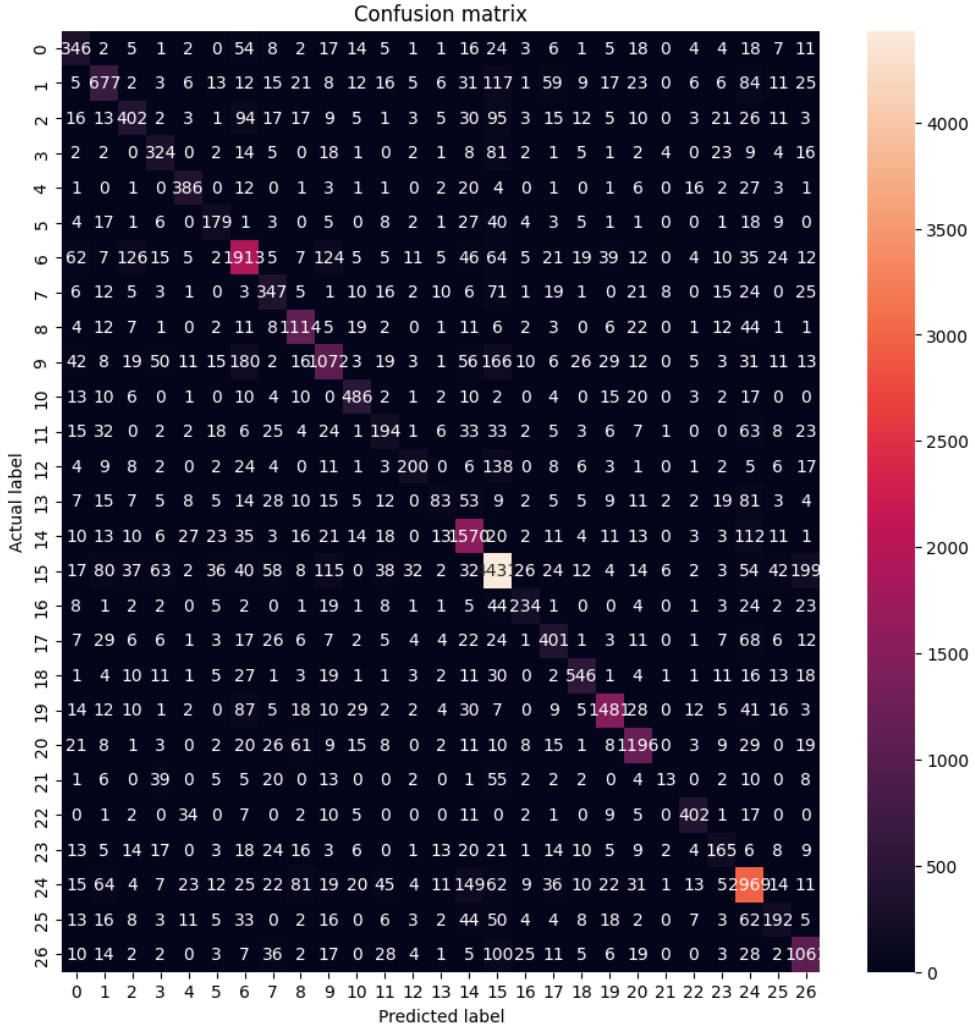
4. F1 score is the harmonic mean of precision and recall. It measures how balanced the model is in terms of both accuracy and sensitivity. A high F1 score means that the model has a good trade-off between precision and recall, or minimizing both false positives and false negatives.

$$\frac{1}{n}\sum_{i=1}^{n} 2 * \frac{Precision_i * Recall_i}{Precision_i + Recall_i}$$

1 . Result for DistilBERT :

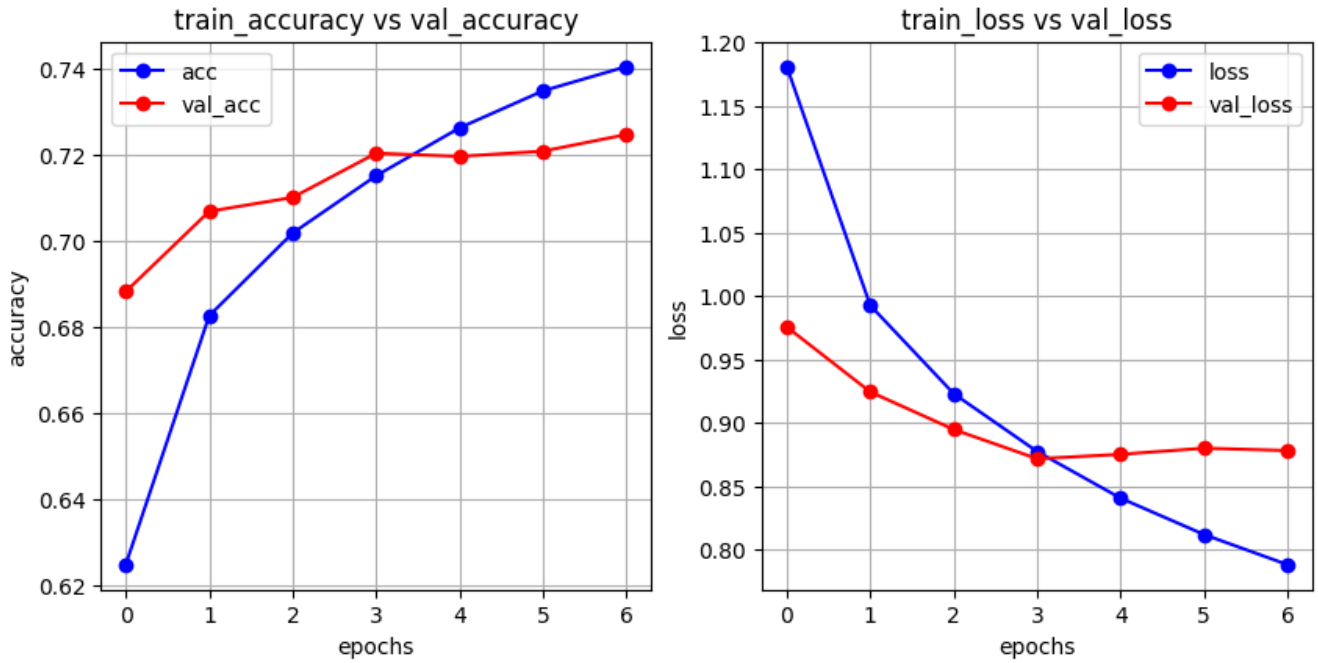


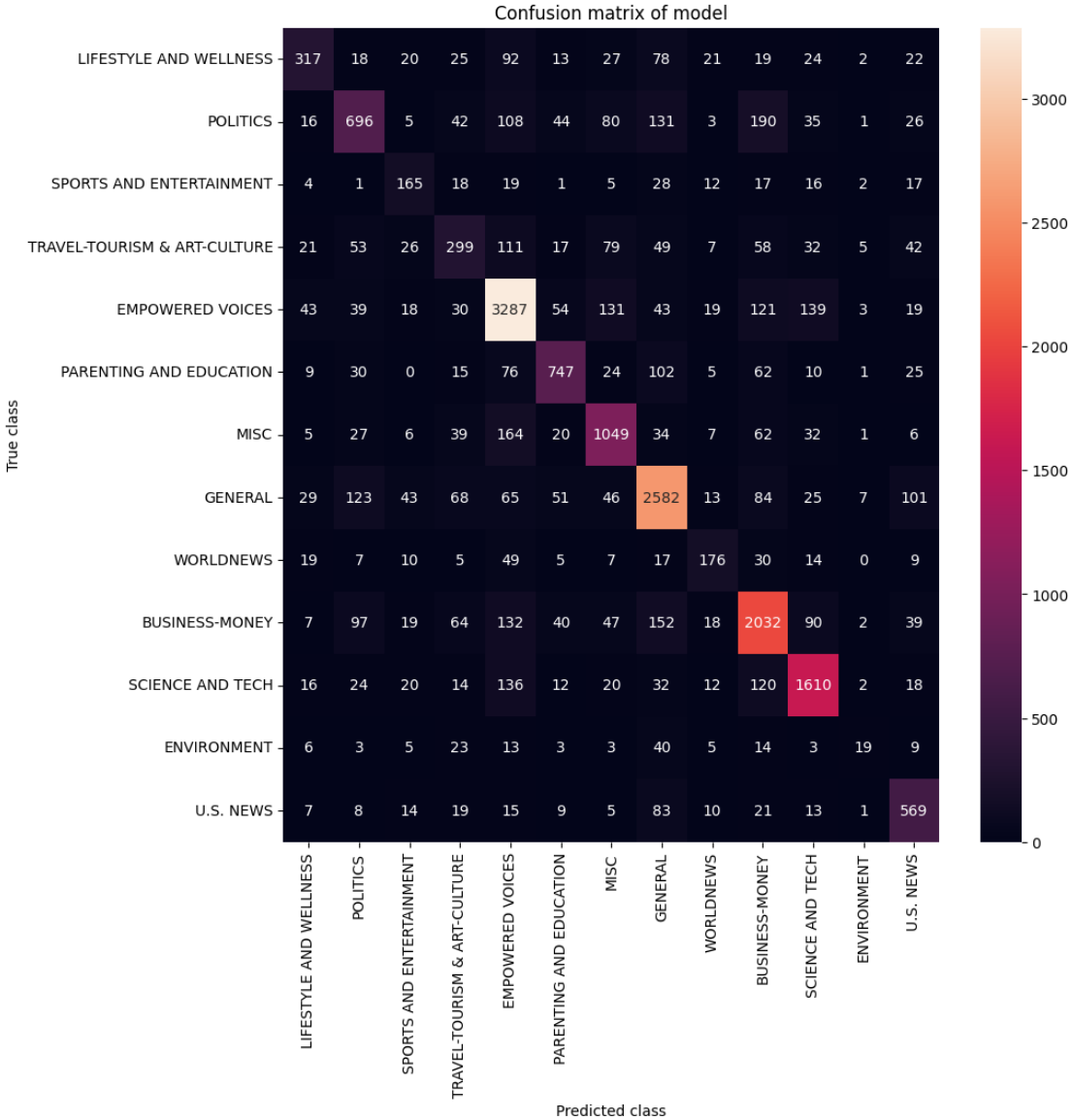
Plotting the training and validation loss



Confusion Matrix

2 . Result for LSTM Model:





confusion matrix LSTM

	DistilBERT	LSTM
Accuracy	**0.7101310932926054**	0.72
Precision	**0.7101310932926054**	
Recall	**0.7101310932926054**	
F1-score	**0.7101310932926054**	

Comparaison of the metrics for DistilBert &LSTM Models

6 . Implementation

7 . Conclusion