INF582 - NEWS ARTICLE TITLE GENERATION

March 31, 2024

Eyal Benaroche, Thibaut de Saivre, Sakula Hys
Kaggle names: Shaamallow, ThibautX2021, Sakula HYS
Team name: Skibidi Overlords

1 Introduction

In this challenge, we are tasked to leverage innovative NLP techniques to decode news articles and condense their main point into a headline. We were allowed to use pretrained models but no external data to develop an algorithm capable of grasping the context and meaning of a wide range of news articles in French. We have explored the dataset, tried to assess its diversity and generate comprehensive, detailed and precise titles for the articles.

Central to this challenge, the pertinence of the titles are measured by the *ROUGE-L F-Sccore* metric. This metric assesses the similarity between the headline generated by our models and the reference titles according to the following formula (with LSC meaning Longest Common Subsequences) :

$$ROUGE - L_{RECALL} = \frac{\sum_{s \in \mathcal{S}} LCS(s,g)}{\sum_{s \in \mathcal{S}} |s|}; ROUGE - L_{PRECISION} = \frac{\sum_{s \in \mathcal{S}} LCS(s,g)}{\sum_{g \in \}} |g|}$$

$$ROUGE - L_{F-Score} = \frac{ROUGE - L_{PRECISION} \times ROUGE - L_{RECALL}}{ROUGE - L_{PRECISION} + ROUGE - L_{RECALL}}$$

1.1 Dataset Analysis

The given dataset is composed of

1. **train.csv**: This file contains 30659 news articles from various topics (in the field text of the csv file) and titles (in the field titles of the csv file).
2. **validation.csv**: This file contains 1500 news articles from various topics (in the field text of the csv file) and titles (in the field titles of the csv file).
3. **test_text.csv**: This file contains 1500 News Articles in total for which we have to generate titles and only half is used for the public leaderboard.

1.1.1 Text Analysis

We decided to analyze the dataset according to the different topics discussed in the articles. For that purpose, we ran a topic detection model to generate labels for each article and differentiate the performance for title generation on each topic. We decided to use an off-the-shelf model `lincoln/flaubert-mlsum-topic-classification` model trained on the ML-SUM Dataset [4] and fine-tuned on press articles.

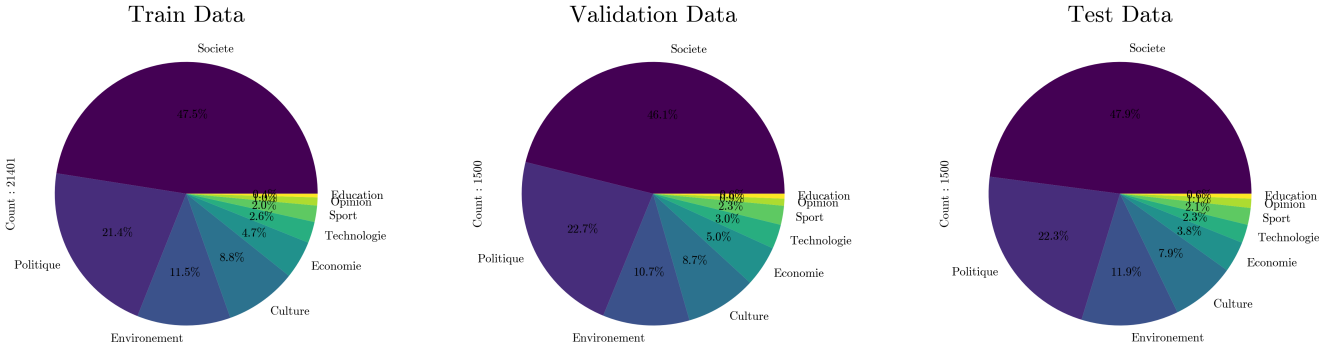

Figure 1: Labels distribution in the dataset

We can see in the label distribution that the very generic label "society" makes up for about half the articles' topics, with the second most present label being politics. This means that the model is trained on a diverse dataset of articles dealing with various subtopics. Since the test and validation dataset have a very similar label distribution, the training is fit for the model to handle diverse article and therefore should be able to generalize.

Looking at the PCA below, based on the labels embeddings, using both the `google-t5/t5-small` and `lincoln/flaubert-mlsum-topic-classification` tokenizer), we realize the embeddings do not account for the classification we have presented just before.

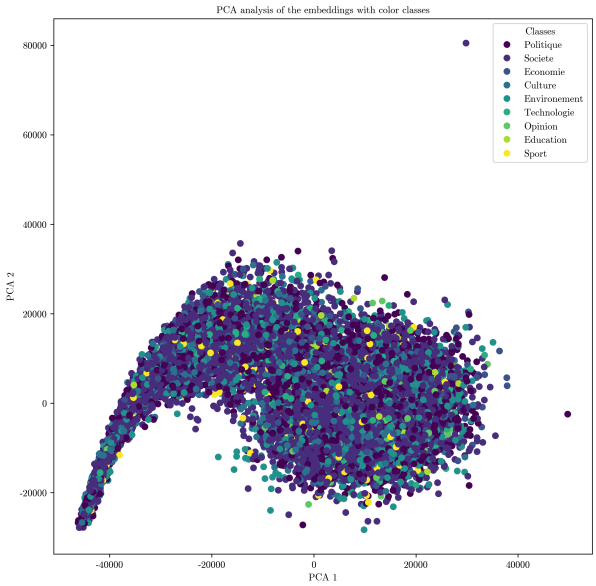

Figure 2: PCA for T5 embeddings

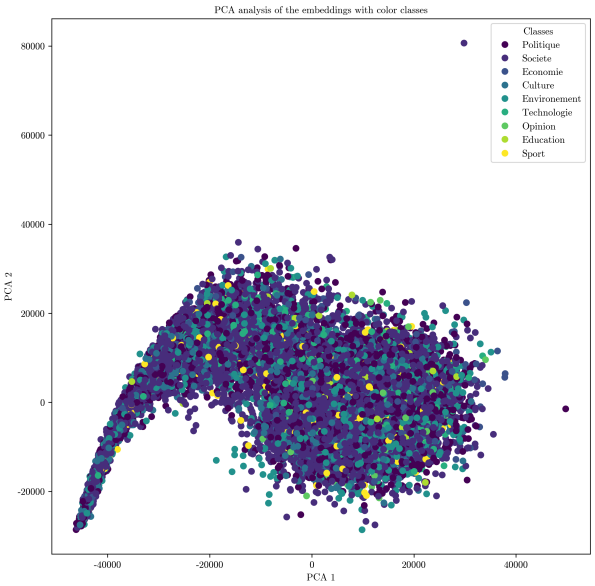

Figure 3: PCA for Flaubert embeddings

Therefore, we can use a simple polar representation to see the size of each article (distance from the center) and observe the quality of the generated titles later on (the angle is computed using the label and adding some jitter to avoid overlap and have a better representation, the distance is log based to have better coverage and visualization).

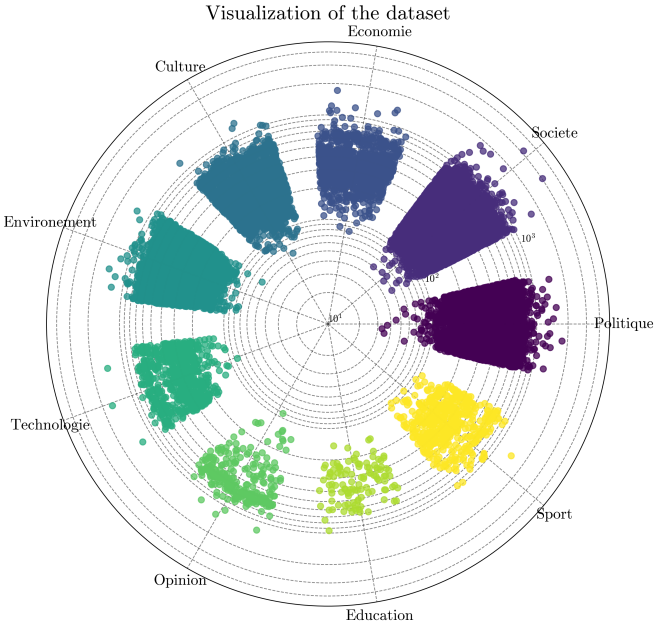

Figure 4: Size distribution for each label

1.1.2 Titles Analysis

For this part, a quick look at some samples of the training data shows that most titles aren't really titles. They are instead an extract of the original article and most often at the beginning of the said article. Contrary to a real title, they are often quite long and don't really act as an incentive to attract potential readers, but instead a sentence that represents the article, either completely extracted from the original article or a summarization sentence. (Here are some examples that motivate this answer id 1463, id 12117)

Since we had some labels for our articles, we tried to check whether our model gave better results on specific topics. Therefore, we ran the model on the validation dataset to avoid any overfitting on the training dataset.

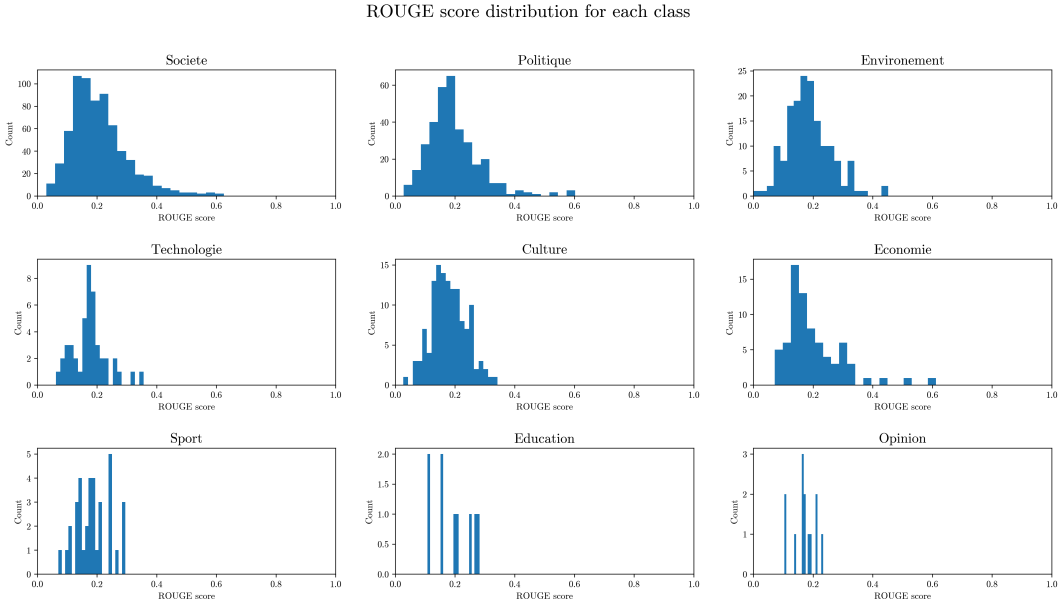

Figure 5: Rouge Distribution for the different topics

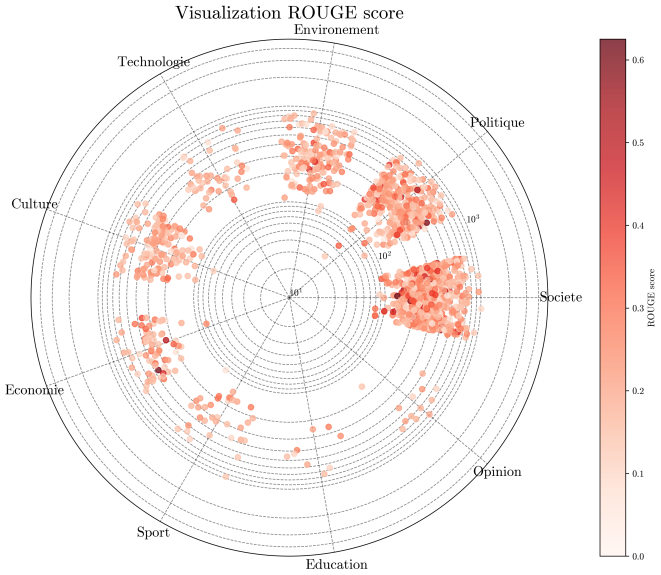

Figure 6: Rouge Score for each topic and size

2 Models

For generating new titles, we have decided to use an encoder-decoder architecture as it is one of the main stream methods for NLP task nowadays. We have used a Bart [1] and a T-5 [2] pretrained model for this specific task, following the *Sequence 2 Sequence* format advocated in this paper [2] in the context of summarization (as detailed earlier, the *"Title"* should be a description of the article hence the summarization approach).

2.1 Tokenizer

Each model from the HuggingFace library comes with its own pretrained tokenizer. We completed the tokenization process with the following settings:

- Strip input strings, replace all whitespace-like characters by whitespaces.
- Truncate or pad inputs in order to match the model input size.
- Extract input token ids, attention masks and target ids into the working dataset.

2.2 Encoder-Decoder Models

We first tested some of the pretrained models tagged for summarization in the French language in order to work with the one that gave the best results without fine-tuning.

The following models gave the best average Rouge-L scores on the validation dataset before fine-tuning:

1. `plguillou/t5-base-fr-sum-cnndm`: 0.20781614162134016
2. `mrm8488/camembert2camembert_shared-finetuned-french-summarization`: 0.20582048179444393
3. `csebuetnlp/mT5_multilingual_XLSum`: 0.20527144694133828
4. `lincoln/mbart-mlsum-automatic-summarization`: 0.20474941262252572
5. `csebuetnlp/mT5_m2m_crossSum`: 0.20217067318519055

We also tried more generic, foundational, models tagged for summarization among other task and not trained on a French Dataset but a mixture of multiple languages :

1. `facebook/bart-base`: 0.128
2. `google-t5/t5-small`: 0.132

We chose to go-on with the fine-tuning phase starting with the `google-t5/t5-small` pretrained model.

2.2.1 Training and Testing settings

We ran the transformer models on GPUs with adapted batch sizes in order to speed-up the processing. We used the `fp16` parameter to train the models with 16-bit precision floating point numbers in order to save memory.

This fine-tuning made the Rouge-L score of a simple `t5-small` model instance go from 0.132 up to 0.187 (and 0.128 to 0.184 for the `bart-base`). However, we were not able to run the fine-tuning in reasonable times on larger models like `plguillou/t5-base-fr-sum-cnndm`.

3 Best sentence extraction

The last method we tested was to choose as a paragraph title the sentence that best summarized it. Our strategy is as follows:

1. Compute embeddings for each paragraph sentence and each paragraph.
2. Concatenate sentence embeddings with their paragraph's one in order to add context.
3. Compute the Rouge-L score of each sentence relative to their paragraph target.
4. Train a regression or NN model to try and predict the Rouge-L score of sentences, given their contextualized embedding.
5. Use a greedy strategy and choose as a title the sentence with the best predicted Rouge-L score.

The results are showcased in the `score_regression.ipynb` jupyter notebook.

4 Further Improvements

As we generate a title, independently of the *Rouge-L* score, we have some concerns about the validity of the generated title, especially in terms of Journalistic objectivity. Indeed, it would be quite unfortunate if our generated titles were to introduce bias or misrepresent the content of the articles. Therefore, we can use the following methods to make sure the generated title is aligned with the original article.

1. **Sentiment Analysis** : To assess the bias of a title, we can use 0-shot classifier for sentiment analysis using a transformer based architecture. Using typical questions to determine the political alignment of an article, we could verify the alignment of the Title is close enough to the alignment of the article that would be the ground truth.
   For our dataset, we could use `lxyuan/distilbert-base-multilingual-cased-sentiments-student`
2. **BERT Score** : Bert Score is another metric that can be used just like Rouge to assess the quality of a generated answer. It can be used to further refine our training process, as we need the reference (again, just like rouge) to judge the quality of the generated content. Further details in this paper [5].
3. **Quest Eval** : This time, we can assess on the fly the quality of the generated summary using an other model trained to generate questions and retrieve answer from whatever textual input. Indeed, QuestEval [3] has been proposed as another method to assess on the fly the quality of a summary as current metrics such as ROUGE are known to be limited and to correlate poorly with human judgments
4. **NER** : Name Entity Recognition is a nice method to make sure the relevant information is inside the title. By adding a NER model during the generation process, we could make sure our *"Title"* is complete enough with the relevant information such as characters or locations.
5. **Further Training** : If we had more computing power, it would have been great to train the already refined models, trained on French dataset, for our specific task as they already gave better results than the foundational models, even when we trained those on the dataset.

Finally, while it's probably the best way to ensure we have high-quality results, the generated titles should be check-up by a human, making sure we have a proper title. The general feeling we got while checking samples from the generated titles are that they feel consistent and human-written but not necessarily the best title. Still, as said earlier in the 1.1.2 Title Analysis section, the expected titles are more a summary of the article than a real title.

Our code is available here as it's bigger than moodle maximum submission size.

References

[1] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, 2019.

[2] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2023.

[3] Thomas Scialom, Paul-Alexis Dray, Patrick Gallinari, Sylvain Lamprier, Benjamin Piwowarski, Jacopo Staiano, and Alex Wang. Questeval: Summarization asks for fact-based evaluation, 2021.

[4] Thomas Scialom, Paul-Alexis Dray, Sylvain Lamprier, Benjamin Piwowarski, and Jacopo Staiano. Mlsum: The multilingual summarization corpus. 2020.

[5] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert, 2020.