Introduction To Text Mining And NLP (INF582)
News Articles Title Generation

École Polytechnique

March 2024

1 Description of the Challenge

The primary objective of the this challenge is to use and develop advanced natural language processing (NLP) models that can automatically generate compelling and informative titles for news articles. Participants are tasked with creating algorithms that can understand the essence of a given news article and succinctly capture its key points in a headline, thus enhancing reader engagement and information retrieval.

In today's fast-paced digital age, the sheer volume of news articles published daily can be overwhelming for readers. An effective headline serves as a crucial gateway that influences whether readers click to access the full article. Automated title generation not only streamlines the content creation process for publishers but also ensures that readers quickly grasp the core message of an article. This challenge aims to address the importance of improving the user experience, enabling efficient information consumption, and ultimately contributing to the evolution of news dissemination in the digital realm. Therefore, it is important to develop and implement generative methods to get automatic summaries and headlines for huge articles.

A typical solution pipeline for this challenge involves a multi-faceted approach. Participants may leverage pre-trained language models, such as transformers, to understand the context and semantics of the news articles. Extractive summarization techniques can be applied to identify key sentences or phrases within the content. Additionally, participants may explore abstractive summarization methods to generate concise and coherent titles that capture the essence of the article. Fine-tuning on a diverse dataset of news articles will be essential to ensure the model's adaptability to various topics and writing styles. The challenge encourages innovative techniques and novel architectures to push the boundaries of automated news title generation.

The challenge is hosted on Kaggle, a platform for predictive modelling on which companies, organizations and researchers post their data, and statisticians and data miners from all over the world compete to produce the best models. The challenge is available at the following link: `https://www.kaggle.com/competitions/inf582-news-articles-title-generation`. To participate in the challenge, use the following link: `https://www.kaggle.com/t/a1dadd21eb7c4a068b0317b37a71a0df`.

2 Dataset Description

As mentioned above, you will evaluate your methods on a list of news articles on various topics. The dataset contains various subjects news articles in the French language. You are given the following files (which are available along the baseline code within the Kaggle page).

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 you have to generate titles. This dataset is distributed equally between the public and private leaderboards on kaggle.

An example of how to prepare the results for your submission is provided in the given baseline code. **Please make sure that your submissions follow the format provided on the baseline (ID, titles) otherwise they would not be acceptable from Kaggle.**

3 Evaluation

The performance of your models will be assessed using the **ROUGE-L F-Score** metric. ROUGE is based on the proportion of n-gram overlap between the system-generated sentence and one or more reference sentences. Given a set S of reference sentences and a generated sentence g, ROUGE-n recall is computed as:

$$ROUGE - n = \frac{\sum_{s \in S} \sum_{gram_n \in S} C_{match}(gram_n, s, g)}{\sum_{s \in S} \sum_{gram_n \in S} C(gram_n, s)}$$

where n is the length of n-gram, $C(x, y)$ is the number of occurrences and $C_{match}h(x, y, z)$ is the maximum number of co-occurrences of x in y and z. ROUGE-n Precision is similar to the ROUGE-n Recall after replacing the denominator in the equation with the total number of tokens in the generated sentence instead of the reference sentences.
Another variant of ROUGE is ROUGE-L which considers the Longest Common Subsequence between the reference and the generated sentence. Thus ROUGE-L Recall and Precision can be written:

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

An example for the computation of this score is provided in the baseline code.

4 Provided Source Code

You are given a script written in Python that will help you get started with the challenge. The script contains two basic extractive summarization baselines:

- **Lead-1** is a baseline method used for generating titles in natural language processing tasks. In this approach, the first sentence of the input text is extracted and used as the title. The rationale behind Lead-1 is that the opening sentence of a piece of text often encapsulates the main idea or topic of the entire document. By using this sentence as the title, the baseline aims to produce a concise and relevant summary of the content. While Lead-1 is a simple and straightforward method, it may not always capture the most informative or engaging aspect of the text, particularly in cases where the main idea is not clearly expressed in the opening sentence.

- **EXT-ORACLE-1**, or Extractive Oracle-1, is a baseline technique used for generating titles by maximizing ROUGE-L scores. Ext-Oracle-1 leverages the original titles to identify the most salient information in the text, aiming to produce titles that are both informative and relevant. While this

method may yield high-quality titles, it requires having access to the original titles and may not be scalable for large datasets.

5 Useful Python Libraries

In this section, we briefly discuss some tools that can be useful in the challenge and you are encouraged to use.

- A very powerful machine learning library in Python: `scikit-learn`[1].
- A very popular deep learning library in Python is `PyTorch`[2]. The library provides a simple and user-friendly interface to build and train deep learning models.
- Since you will also deal with textual data, the Natural Language Toolkit (`NLTK`)[3] of Python can also be found useful.
- `Gensim`[4] is a Python library for unsupervised topic modeling and natural language processing, using modern statistical machine learning. The library provides all the necessary tools for learning word and document embeddings.
- `Hugging Face`[5] an immensely popular Python library providing pretrained models that are extraordinarily useful for a variety of natural language processing (NLP) tasks.

6 Rules and Details about the Submission of the Project

Rules. The following rules apply to this challenge: (i) one account is allowed per participant (ii) there is a limit in the size of each team (at most 3 members), (iii) privately sharing code outside of teams is not permitted, (iv) there is a limit in the number of submissions per day (at most 4 entries per day), (v) the use of external data is **not allowed** (except from word embeddings, e.g. BERT, GPT, BART, WordVec, etc..). For instance, you are not allowed to use external data to determine if a summary is generated by a machine or written by a human. (vi) your code must be **reproducible**.

Evaluation and Submission. Each team must fill this form before **21/03/2024**.
Note: without filling this form you will not be able to submit your files.
Your final evaluation for the project will be based on (1) the presentation you will give (**40**%), (2) on your position on the private leader-board and the score that will be achieved (**30**%), and (3) on your total approach to the problem and the quality of the report (**30**%). As part of the project, you have to submit the following:

- A 4-5 pages report, in which you should describe the approach and the methods that you used in the project. Since this is a real classification task, we are interested to know how you dealt with each part of the pipeline, e.g., how you created your representation, which features did you use, which classification algorithms did you use and why, the performance of your methods (loss, accuracy, and training time), approaches that finally did not work but are interesting, and, in general, whatever you think that is interesting to report.

[1] `http://scikit-learn.org/`
[2] `https://pytorch.org/`
[3] `http://www.nltk.org/`
[4] `https://radimrehurek.com/gensim/`
[5] `https://huggingface.co/`

- A directory with the code of your implementation (not the data, just the code).
- Create a `.zip` file containing the code and the report and submit it to Moodle.
- **Deadline: 31/03/2024 23:59**

Presentation: As mentioned above, you will be asked to present the approach you followed. Therefore, you will need to prepare some slides (using ppt or any other tool you like).
Date of presentation: TBA