Sentence Splitter
Multilingual Natural Language Processing
Homework 2

Ercoli Fabio Massimo
802397

Della Porta Nicolò
1920468

1 Introduction

Quoting [Redaelli and Sprugnoli, 2024] "Sentence splitting, that is the segmentation of the raw input text into sentences, is a fundamental step in text processing". According to [Frohmann et al., 2024] the main challenges are:

- robustness to missing punctuation
- effective adaptability to new domains
- high efficiency

According to [Redaelli and Sprugnoli, 2024] we can add to the list:

- multilinguality

Because a sentence splitting that works well for English may not work well to split another language.

In this project we implemented two models for sentence splitting, using an Italian corpus as train and validation set. The first one is based on an embedding model, the second one is based on generative LLM. We want to analyze compare the two approaches. We want also to test the models out of domains.

2 Methodology

2.1 Embedding-based

We fine tuned a pretrained embedding model (multilingual or trained on Italian language) using the test and validation datasets provided by the homework guide.

The original dataset provides two large texts (one to use as train and the other as validation) together with the golden labels to mark the end of sentence (1) and all the rest of the words (0).

In order to make the datasets suitable for the training we had to apply some transormations. We needed to group words into sequences of tokens aligning the golden labels consistently. The number of tokens of each sequence must fit the max length of the embedding models, for instance 512.

We generated different datasets using different number of words to generate each sequence: 64, 128, 192, 256. Notice that the number of tokens for each sequences will be strictly grater, since for each word the tokenizer of the model will produce one or more tokens.

Before to use the datasets we still need to align (as mentioned above) the labels to the tokens. The alignment strategy we applied consists in keep 1 as the first token generated from a word having label 1, use 0 for all the other cases.

One aspect that we had to address was the fact that the label distribution is very unbalanced for this use case. Most the labels are 0s, and few 1s. We applied 2 different ideas.

First of all, we set the *load_best_model_at_end* as training argument to *true*, using *metric_for_best_model* to F1. This to implement an early stopping based on F1, and not on the accuracy (that would be misleading for such a unbalanced dataset).

Second of all, we override the loss function to weight much less (1/30) a miss-labeling on 0s then the ones on 1s. We called it a weighted trainer.

On the inference we don't need to align the label, since the labels are generated by the model.

2.2 LLM-based

Fine tune a LLM is usually proebitive or impossible using a modest hardware, such as a single GPU with 8 GB / 16 GB RAM. So we applied to techniques to reduce the amount of memory required (and also the time) to fine tune the models: LoRA (Low Rank Approximation) and the parameters quantization.

A special focus was give to create good prompts for both the training and inference phases. We first generate new datasets from the ones we used for Embedding-based solution in which for every sequence we have:

- *input_text*: here we simply concatenate all words of the sequence in a single text
- *output_text*: here we group all the words that belong to the same sentence in different lines

This is not a prompt yet. To create the final prompts we created dictionaries of roles (system / user / assistant) of contents (we called them: conversations) and we passed them to the LLM tokenizer in order to have those converted in the proper chat templates.

We used Supervised Fine-Tuning (TRL SFT) Trainer for train the model.

The fine tuned model is (as usually) pushed to the Hugging Face repository, so that it can be used for inference from any spot.

3 Experiments

3.1 F1 on different models - same domain

The performance goal we set is to maximize the F1 score on the validation set. The test set is unlabeled, so we cannot use it. Our expectation is that with a good result on the validation set can generalize nicely on the test set.

For the Embedding-based solution we tried the following base models:

- ModernBERT-base-ita (Dec 2024)
- Italian BERT XXL (most established)
- XLM-RoBERTa base
- XLM-RoBERTa large
- Italian ELECTRA

For the LLM-based solution we tried the following base models:

- LLaMA-3-8B Instruct
- Mixtral 8x7B Instruct
- Qwen3-4B Instruct

These tests are executed on the same domain, since all the datasets (train, validation and tests) come from the same novel: *I Promessi Sposi*.

3.2 Out of domain tests

We tried the same model on a different authors. For doing it we used the dataset from [Redaelli and Sprugnoli, 2024] in which different authors from the same century are used. The dataset has already the gold results annotated by the authors of the paper.

4 Results

In this section we want compare the two approaches embedded-based vs LLM-based. On all models that were tested. Highlighting:

- F1 metric - in context
- F1 metric - out of context
- memory usages (where it is possible)
- time required (where it is possible)

A Additional details/describe Appendix

References

Markus Frohmann, Igor Sterner, Ivan Vulić, Benjamin Minixhofer, and Markus Schedl. 2024. Segment any text: A universal approach for robust, efficient and adaptable sentence segmentation. *Preprint*, arXiv:2406.16678.

Arianna Redaelli and Rachele Sprugnoli. 2024. Is sentence splitting a solved task? experiments to the intersection between NLP and Italian linguistics. In *Proceedings of the 10th Italian Conference on Computational Linguistics (CLiC-it 2024)*, pages 813–820, Pisa, Italy. CEUR Workshop Proceedings.