Capstone public summary

Comparison of Named Entity Recognition Pre-Trained Models Applied to Financial Documents

MSc Business Analytics 2021-22

Department of Economics and Business, Central European University

Aftab Alam **June 2022**

Introduction

This project focuses on comparing Named Entity Recognition across different NLP libraries, by applying pre-trained models to financial documents and analyzing performance based on accuracy, scalability, memory usage, ease of implementation, maintenance, etc. The client is interested in identifying the best performing model across the following labels: LOC (location), ORG (organisation) and PER (person). Different available pre-trained models from both the libraries were tested on a benchmark dataset and evaluated using different metrics to recommend a final model.

Dataset

For the purpose of this project, the dataset has been taken from Github. It was uploaded by 'juand-r' user. The dataset is generated using CoNLL2003 data and U.S. Security and Exchange Commission (SEC) filings. The dataset was created by Alvarado, Verspoor, and Baldwin for their paper 'Domain Adaptation of Named Entity Recognition to Support Credit Risk Assessment'. They used eight documents selected at random and sourced from SEC for manual annotation based on four named entity types provided in CoNLL-2003 dataset; PER, LOC, ORG, and MISC. They carried out the annotation using the Brat annotation tool.

The dataset is divided into two parts, train, and test. The training dataset contains tokens and labels from five out of the eight selected documents, whereas the remaining are in the test dataset. But for the purpose of this project's analysis, both datasets were combined, since the project uses pretrained models. This combining results in a larger dataset comprising of 1467 sentences with 54256 tokens.

Environments

The project aimed at using virtual environments to mimic production capabilities, however, due to resource constraints and other limitations, only free versions of the readily available virtual environment were considered. Databricks Community version was chosen along with the other free

virtual environment, Google Colab. The project initially tried leveraging both Databricks Community and Google Colab to run the models, however, due to encountering some issues with Java for Spark NLP, project continued with using Google Colab only.

Process

The dataset was run on a total of 9 different models, namely: Spark Onto-100, Spark Onto-300, Spark BERT, Spark NLU Onto, Spark NLU CoNLL, Spacy Small, Spacy Medium, Spacy Large, and Spacy Transformers. For Spark NLU models, implementation was the easiest since it is a Python library designed for easy implementation. The library had to be downloaded along with the relevant modules of Onto and CoNLL. The dataset was then run on it.

For Spacy models, implementation steps were a bit more than Spark NLU, however, a lot less than the other Spark models. Each Spacy model had to be downloaded and initiated before running the models on the dataset. Spark models had the most steps as the pipeline had to be built from start; initiating with document assembler, sentence detector, tokenizer, spell checker (optional), NER converter, embeddings, and calling the model.

Limitations & challenges

The project came across Java issues with Databricks and due to time limitation, it was decided to run the models on Google Colab. Another issue arose with Spark NLU models; when run on the full dataset, it gave a Java error. However, it was resolved by converting the text column of the dataset into a list before inputting it to the models.

Due to the unsatisfactory results of Spark Onto and Spark BERT models, client suggested training the Spark DL on train version of the dataset to see if it improved results on the test dataset, however, the results still remained unsatisfactory. One of the reasons being that the model was trained on a dataset that was extremely small compared to the pre-trained Spark models training dataset. Thus, it was decided with the client to not train our own models and use pre-trained models only.

The ideal dataset would have been one that was manually annotated and had maximum types of tags, for example, 18 tags that are mostly used by the current state of the art models. Additionally, the required dataset would have been a financial dataset as the final model would be used for financial documents. However, the project could not find the perfect dataset. It was successful only in finding a dataset that contained financial corpora, but the annotations were limited to four types, namely: MISC, PER, LOC, ORG.

The biggest challenge of the project was to understand how to compare the resulting entities from the models with the benchmark tags. Reason being that all the models used in the project except for Spark

BERT and Spark NLU CoNLL models, the resulting entities were tagged in an 18 labels format. Whereas the benchmark dataset was annotated with 4 tags as mentioned above. It was hence decided to create contingency tables for all the models to see how the 18 resulting entity labels of the models could be mapped onto the 4 entity labels in the benchmark dataset annotation. Ultimately, it was decided to drop the 'MISC' & 'O' tags, as the client was only interested in 'PER', 'ORG', and 'LOC'. Additionally, since 'ORG' from the benchmark dataset was mostly tagged as 'LAW' in the models, the client directed the project to map it onto 'ORG', hence mapping both 'ORG' and 'LAW' onto 'ORG'. Also, 'LOC' from the benchmark dataset was mostly tagged as 'GPE' by the models hence the client wanted both 'LOC' and 'GPE' to be mapped onto 'LOC'.

Results

To test the efficiency and scalability of the models, the project ran the models on test (small), train (medium), and combined (large) versions of the dataset while measuring the CPU processing time and memory consumption for the whole process of the model, starting from calling the libraries to saving the results as pandas data frames.

In terms of CPU processing time, Spacy small took the least amount at 1.73 seconds on the small dataset, however, when considered the scalability of the models, Spark models performed considerable better than Spacy models. For memory usage, Spark Onto-100 consumed the least memory on the small dataset, 62.21 MBs, however, from scalability perspective, Spark NLU, Spacy small, Spacy medium, and Spacy Large performed better.

Finally, since the client was interested in decreasing false negatives in the results, recall was chosen as the metric of accuracy. Recall was considered for each individual label to see which model could be used for each label. Recall for 'LOC' was highest for Spacy Transformers, Spark Onto-300 for 'ORG', and Spacy small for 'PER'. Additional to these recall scores, the support numbers for all the labels were also considered in making the final recommendation.

Recommendations

Considering all the factors for evaluation, the project recommended the client to use NLU Onto for 'ORG' label which has the third highest recall score at 97.3%, Spacy small for 'PER' which has the highest recall score at 82.5%, and NLU Onto for 'LOC' label which has the second highest recall score of 82.4%. Project also recommended the client to test the models on their own annotated data since the project Colab notebooks are available it just needs the new data and the size of the dataset used in the project is small.