Case Study Report

Contact information

Full Name: **Ruslan Yermakov**

Email: **yermakovruslan@gmail.com**

Linkedin: ruslanyermakov

Code available at https://github.com/wingedRuslan/labtwin-test

Problem definition

Goal: implement a multi-label classification model that can classify a wiki_text into category[1-6]

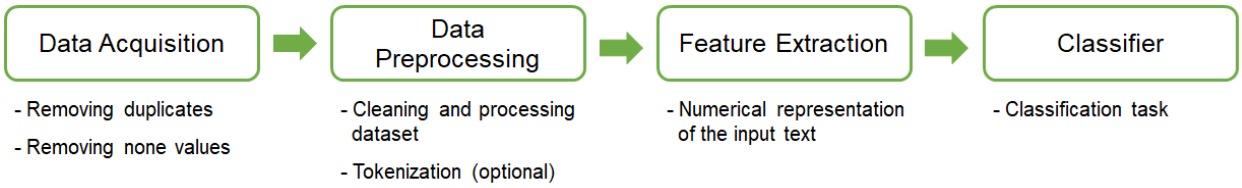
Assumption: “classify a term as scientific or non-scientific, and if it is scientific, also classify it into a scientific category.“ - Since each term is associated with the corresponding text in texts.pkl, to procedure to classify a term would be the following:

- Find the corresponding text for the term. Classify the text associated with a term into 1 of 6 labels.

Dataset: wikipedia domain data

Project implementation

General steps in the project



Each step in this pipeline is important and contributes towards achieving high performance in the text classification task. However, the key factor boosting performance is the Feature Extraction part. The numerical representation of the input text helps classifier to better distinguish between classes and thus make better decisions.

For the feature representation I used contextual representations by DistilBERT(smaller, faster, cheaper and lighter than BERT). Taking into account my limitation in computation power, I was not able to use BERT (or any other “complex” language model).

Preprocessing Phrase:

- Data_Exploration_Processing.ipynb

Steps like:

- lowercase text, term, labels
- check for / remove duplicates
- check for / remove outliers
- Check labels distribution (since the task is multi-label classification)

After preprocessing step - the size of the dataset: 8272

Class imbalance was detected. To address this problem, I undersampled the overrepresented class and oversampled the most underrepresented class.

Then I split the dataset into train/test datasets (~10%).

For more detail, please see the corresponding notebook!

Modelling:

- training_DistilBERT_multilabel.ipunb

Used Google Colab to train a `DistilBERTForSequenceClassification` model from Hugging Face library.

For more detail, please see the corresponding notebook!

Inference:

- predicting_labels_DistilBERT.ipynb

Performing predictions on the test_dataset.

Results:

```
Confusion Matrix

[[278   6   2   1   1   1]
 [  4  32  11   3   0   2]
 [  0   6  38   1   4   4]
 [  2   2   2  78   5   1]
 [  1   2   1   2  59   0]
 [  0   0   4   0   1  48]]
```

```
Accuracy: 0.89

Micro Precision: 0.89
Micro Recall: 0.89
Micro F1-score: 0.89

Macro Precision: 0.82
Macro Recall: 0.83
Macro F1-score: 0.82

Weighted Precision: 0.89
Weighted Recall: 0.89
Weighted F1-score: 0.89
```

According to the confusion matrix, most of the labels were predicted correctly. F1-score of 90% confirms that the developed approach achieved a great performance.

Potential Improvements to the work:

I would treat the developed approach as a baseline upon which further work needs to be carried out in order to improve the performance.

Data-Centric Part:

- preprocess text field (e.g. removing special characters, certain parts (sections) in text based on domain knowledge)
- check for duplicates via contextual embeddings by BERT other than straightforward string-wise check
- check that labels per texts are appropriate (avoid cases where label “drug” for text about a film)

Model-Centric Part:

- To perform better in current multi-classification task it might be better to leverage 2 classifiers, 1st - to separate "non_science" labels from "science" labels with the help of DistilBERT; 2nd - if the label is "science", classify text further into scientific labels by leveraging BioBERT
- Hyper parameters search for the better set of hyperparameters for training models