Laurent GUIDDIR

NLP Project Report: Classification on PubMed 200k RCT Dataset

This report outlines the methodology, evaluation and result of our NLP sentence classification project on the Pubmed 200k RCT dataset.

The goal of this project was to use two different techniques to train models in order to classify a substantial amount of medical sentences from PubMed abstracts into five categories : BACKGROUND, METHODS, RESULTS, OBJECTIVE and CONCLUSIONS.

Moethodology:

In this dataset there is three file, train.txt to train our models, dev.txt to eventually fine tune our models and pre-test them, and finally test.txt to our selected models on data never seen before.

1 – Preprocessing :

The first, step is to pre-process our data by removing what is useless such as the articles Id's.

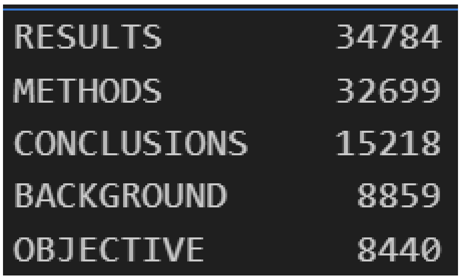
Therefore, for each file we extracted sentences and their respective labels.

We then proceeded to lemmatize each sentence for all three file using spacy en_core_sci_sm model(we decided to work on reasonable and manageable SUBSETs of the three file, due to computing power reasons).

With these newly obtained lemmatized sentences, we extracted the features using TF-IDF and unigram bag of word of TF-IDF to get a sparse matrix highlighting the relevance of each word for each sentence by comparing the Term-frequency (TF) of the words in a sentence relatives to their frequency in the whole corpus of sentences (IDF).

First method, Training our model using the features extracted with TF-IDF :

We then proceeded to train a model (model_balanced) with a logistic regression using the parameter class_weight set to balanced, to account for the fact that our dataset was not balanced, in order not create a biased model.



```
RESULTS        34784
METHODS        32699
CONCLUSIONS    15218
BACKGROUND      8859
OBJECTIVE       8440
```

The data distribution of our training subset, as we can see some classes are significantly more represented than other.

We also built a twos other model to compare the performances of our the model_balanced; by using the same technique but without the classe weighting parameter for the logistic regression, and another using the methods smote to oversample the underrepresented class instead of weighting them to account for the classes imbalance.

<u>Second methods, Training our model (“top_model”)using features extracted with PubMedBERT word embedding model :</u>

The PubMedBERT embedding model, is a state-of the art model trained on biomedical literature from pubmed and is therefore particularly interesting for our classification task.

Extracting embeddings for our seconds models:

We used PubMedBERT to extract the sentences embeddings on our lemmatized_sentences for our training, dev, and test set. But at second thought we remembered that we should have done in on full the sentences since PubMedBERT has been trained to extract meaning from full texts and the lemmatized version is less rich in meaning due to the process of lemmatization itself (such as lowercasing and tacking out the suffixes), even if the lemmatization was done with scipacy.

Then, we trained our model “**top_model**” using the embeddings, once again with a logistic regression.

And we used “top_model” to make predictions on our development set and our test.

As expected, we got significantly better result with this model even if he was trained on the lemmatized version of the sentences due the quality of the word embedding vector the model was trained on.

We also made a second model (“**fullSentence_top_model**”) with embedding extracted from the full sentences (i.e the non-lemmatized version) and got even better results.

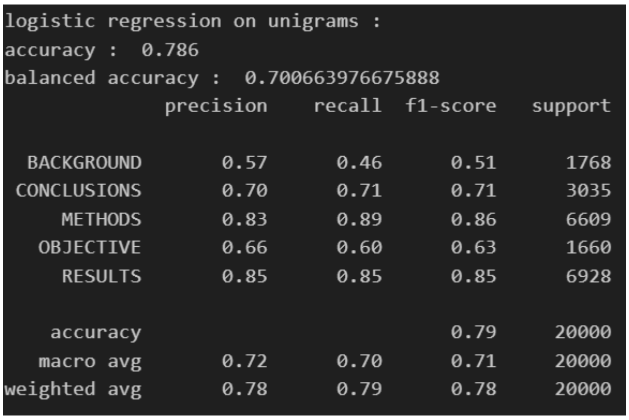
<u>RESULTS: For all models we used a logistic regression</u>

<u>Technique 1 :</u> (selected model -> “model”)

With the **first technique** using the scispacy lemmatiztion, unigram bag of words and TF-IDF feature extraction, we produced three models with their respective classification report bellow that we are going to discuss.

(*) F1-score : is a measure of the harmonic mean of precision and recall

(*2) : correctly classified sentences of the classes / number of sentences of the given classe

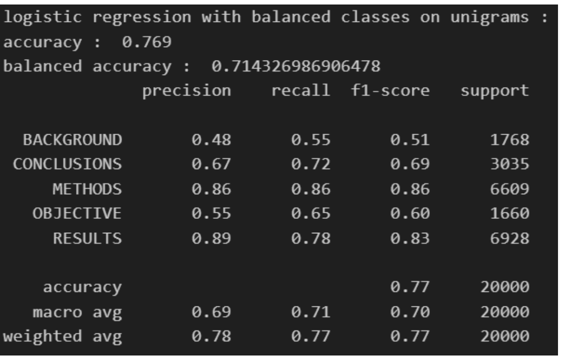


```
logistic regression on unigrams :
accuracy :  0.786
balanced accuracy :  0.700663976675888
              precision    recall  f1-score   support

  BACKGROUND       0.57      0.46      0.51      1768
 CONCLUSIONS       0.70      0.71      0.71      3035
     METHODS       0.83      0.89      0.86      6609
   OBJECTIVE       0.66      0.60      0.63      1660
     RESULTS       0.85      0.85      0.85      6928

    accuracy                           0.79     20000
   macro avg       0.72      0.70      0.71     20000
weighted avg       0.78      0.79      0.78     20000
```

Normal **model :** its our base model we can observe a decent accuracy so the model did good on overall, but with smaller class underperforming by all metric and lager class performing way better.



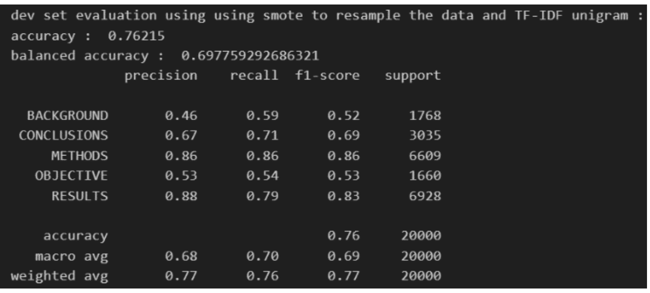
```
logistic regression with balanced classes on unigrams :
accuracy :  0.769
balanced accuracy :  0.714326986906478
              precision    recall  f1-score   support

  BACKGROUND       0.48      0.55      0.51      1768
 CONCLUSIONS       0.67      0.72      0.69      3035
     METHODS       0.86      0.86      0.86      6609
   OBJECTIVE       0.55      0.65      0.60      1660
     RESULTS       0.89      0.78      0.83      6928

    accuracy                           0.77     20000
   macro avg       0.69      0.71      0.70     20000
weighted avg       0.78      0.77      0.77     20000
```

model_balanced : For this model we had a better balanced accuracy, which is the mean of the recall (*2) of the differents classes. But more importantly we can see that the recall of underrepresented classes is significantly higher than the normal model, it was the goal of this model, reducing biases by weighing the classes thus penalizing misclassification of the smaller classes more than the bigger ones to achieve a less biased model. But still got an almost identical f1-score because the recall increase was offset by a precision decrease for the underrepresented classes.

Even though they both have near identical f1-score (0.71 vs 0.70) we can see that the non-balanced one still have a equal or better f1-score for each class so we can choose the normal model.

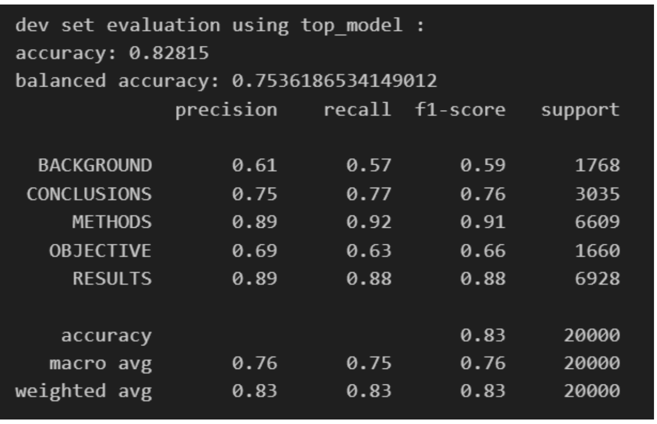


```
dev set evaluation using using smote to resample the data and TF-IDF unigram :
accuracy :  0.76215
balanced accuracy :  0.697759292686321
              precision    recall  f1-score   support

  BACKGROUND       0.46      0.59      0.52      1768
 CONCLUSIONS       0.67      0.71      0.69      3035
     METHODS       0.86      0.86      0.86      6609
   OBJECTIVE       0.53      0.54      0.53      1660
     RESULTS       0.88      0.79      0.83      6928

    accuracy                           0.76     20000
   macro avg       0.68      0.70      0.69     20000
weighted avg       0.77      0.76      0.77     20000
```

The **model_resampled :** was the worse performing model of the three, the artificial synthetic oversampling was probably not remotely as rich compared to what simple new and different sentences could have provided.



```
dev set evaluation using top_model :
accuracy: 0.82815
balanced accuracy: 0.7536186534149012
              precision    recall  f1-score   support

  BACKGROUND       0.61      0.57      0.59      1768
 CONCLUSIONS       0.75      0.77      0.76      3035
     METHODS       0.89      0.92      0.91      6609
   OBJECTIVE       0.69      0.63      0.66      1660
     RESULTS       0.89      0.88      0.88      6928

    accuracy                           0.83     20000
   macro avg       0.76      0.75      0.76     20000
weighted avg       0.83      0.83      0.83     20000
```

<u>Technique 2 :</u> (selected model -> "fullSentence_top_model")

The **top_model** : was trained on embedding made from the lemmatized version of the sentences by the PubMedBERT pre-trained model.

The result are good, but slightly less that our selected version (**fullSentence_top_model**), so lets jump to our selected model.

The **fullSentence_top_model :** was train on embedding made from the full sentences by the PubMedBERT pre-trained model.

As expected, we got significantly better result than the model using the previous technique (technic 1), since it was trained on embedding made bu PubMedBERT, which is a state of the art embedding model for medical texts. It also performed better than the **top_model** since it was trained on better embedding (made from the full sentences).

We also got identical result for all metrics across all the classes on the test set, so the model did not overfit the training data and did a pretty good job at generalizing.

Conclusion : For the first technique we selected the model “**model**” and for the second technique our best model was **full_sentence_top_model.** The second method was naturally gave us the best result since the model was train on extremely qualitive embeddings provided by PubMedBERT which is a state of the art embedding model trained on a huge amount of medical abtract and articles from PubMed.