Natural Language Processing GLUE classifiers

Kolmakova Elizaveta
Innopolis University
Innopolis, Russia
e.kolmakova@innopolis.university

I. Introduction

Natural language processing as an area of machine learning is high-demanded for the industry and research field. For the current state of NLP approaches the machine can not deeply understand the semantics of language, while current NLP models can easily process the syntax of language. There are different syntax model tasks: Question Answering, Summarization, Text Classification, Text Generation, Translation, Sentence Similarity, Feature Extraction and other.

Text classification is one of the most applied tasks for machine learning. In general, classification is done withing the following pipeline: text preprocessing, text embedding, model architecture building and performance interpretation or model evaluation. Initially preprocessing is applied to make a sentence unique enough (by correcting it and filtering out outliers and least informative words) in order to give higher impact of its features. Text embedding is an open for research topic where many researchers contribute introducing their various models for text representation. Choice of representation function depends on the task and its restrictions or required performance: for some models word2vec is enough, for others more advanced models are needed. Another point of research is model architecture due to its diversity. The choice of the model depends on the task and chosen representation. For example, it is not beneficial to use Seq2Seq models for classification task because this is generative model. And the final important stage is performance evaluation. Assessment is vital basically for understanding how successful were made embedding and model architecture choices, but also for quality performance evaluation for deeper understanding as an attempt of making a model more interpretable.

The aim of this work* is to build and assess the performance of classifiers for three datasets from GLUE - CoLA, SST-2 and RTE. The report contains the following sections:

- **Introduction** - short description of an general approach for the study
- **Methodology** with description of applied approaches and experiments.
- **Analysis** with obtained results and its interpretation.
- and **Discussion and Conclusion**

*Link on GitHub: https://github.com/Elizaveta55/glue$_classifiers$

II. Methodology

This section is divided into three parts: datasets description, information about an approach and evaluation part.

A. Dataset

Three main datasets are used for the study: CoLA, SST-2 and RTE. Table I represents basic statistics about these dateseta: amount of instances, amount of instances per class and Top3 state-of-the-art models accuracy performance by [1]–[3]. Amount of instances per class is important to measure because there should be deep understanding of data including degree of imbalance. Imbalance might have significant impact on performance apart from main performance contributors which were listed above such as choice of representation function and model architecture. Controlling these indicators we try to ensure no other factors impact on performance and all of these factors are control variables. Only one factor can be an independent variable in order to measure its contribution to model performance.

Table I shows different bottlenecks of datasets: CoLA dataset is imbalanced, SST2 is a heavy dataset, RTE is not typical classification task because it has two sentences as an input therefore input can be represented differently. Following sections provides with experiments setups and metrics for evaluation for each dataset.

B. Approaches

1) CoLA: Three variables are selected to be investigated: representation function, model choice and balancing techniques.

1) **Representation function.** This part is responsible for mapping text features into vector representation. Two options were selected: doc2vec and BERT with attention mask. Doc2vec is selected because this model is an optimal trade-off between effectiveness and computational and/or memory consumption. In contrast with doc2vec BERT model is more advanced model with high performance but it resources-demanding.
2) **Model choice.** Generally, selected models could be divided into two groups: basic classifiers and recurrent classifier. Basic classifiers includes LogisticRegression, XGBClassifier, NearestCentroid, etc., recurrent classifier is a bidirectional LSTM.

Name	train instances	test instances	instances for class '0'	instances for class '1'	imbalanced ratio	Top1, acc / model	Top2, acc / model	Top3, acc / model
CoLA	8551	1043	2528	6023	42:100	86.4% /EFL	78% / FNet-Large	70.8% /T5-11B
SST2	67349	872	29780	37569	100:126	97.5% / SMART-RoBERTa Large	97.4% / T5-3B	97.4% /MUPPET Roberta Large
RTE	2490	277	1240	1250	1:1	93.2% /DeBERTa-1.5B	92.8% /MUPPET Roberta Large	92.5% /T5-11B

Table I: Datasets statistics and performance

3) **Balancing techniques:** undersampling and oversampling by SMOTE. These two techniques are one of the simpliest to apply to evaluate the performance. Undersampling is applied on majority class in order to make it equal to minority class. Oversampling is more advanced technique and includes multiple approaches from the simplest and effective like SMOTE to most relevant generative models to produce new samples.

2) SST2: SST2 is a large dataset with similar to CoLA approach, but it does not require an application of balancing techniques because the dataset is not significantly imbalanced. In terms of representation function application of BERT model with attention mask is not suggested because it is time-consuming - it would take approximately eight hours of calculations. Therefore we aim to evaluate performance of different classifiers and models.

1) **Representation function** is presented only by Doc2Vec and is not presented by BERT due to time limitation. For better performance it is suggestible to apply BERT representation as well because it is one of the most advanced model.
2) **Model choice** is presented within two groups: basic classifiers and recurrent classifier. Basic classifiers includes LogisticRegression, XGBClassifier, NearestCentroid, etc., recurrent classifier is a bidirectional LSTM. Here we applied different configurations of LSTM architecture but intentionally did not apply high variability of configurations because we aim to evaluate contribution of every added parameter and based on contribution make a conclusion about importance of added parameter. Explored parameters are: bidirectional or onedirectional LSTM, dropouts and one additional dense layer with sorftmax activation between LSTM and output layers.

3) RTE: Due to specific of input data the main aspect was selected to be considered - the approach of representation of two sentences in an effective manner. Generally, there are several aspects to consider:

1) **Sentences representation.** Several assumptions are investigated based on representation of each sentences: concatenation of features of representation of two sentences might be sufficient to train classifiers how to distinguish one class from another, subtraction of features of representation might be enough of averaged addition of features of representation might be enough to train classifiers how to distinguish one class from another. The choice of these assumptions is motivated with the attempt to find an optimal representation pattern which is easy to understand and not heavy to compute.
2) **Model choice:** basic classifiers and LSTM.

C. Evaluation

Due to specific of each dataset different metrics are suggested for the study: f1-score, AUC and confusion matrix. F1-score is very simple to calculate but at the same time it reflects performance considering imbalance as well as AUC. Confusion matrix reflects correctly and incorrectly classified instances within its classes.

III. Analysis

The table II represents metrics values for different datasets. Application of different sipmle classifiers resulted in deriving the most performed therefore in the table instead of "classifier" option the name of most performed classifier stated.

A. CoLA

ColA classification is complicated with data imbalanced, but despite of this f1-score were achieved up to 0.81.

1) **Representation function.** Based on AUC metric BERT model performed better then Doc2Vec. BERT model is more complex state-of-the-art representation model with better architecture to derive data features therefore it managed learn more data features. Moreover, BERT model is a pretrained model therefore it has better understanding of data distribution because it had learnt more features from other datasets. Doc2Vec was trained on CoLA dataset and could not benefit from previous features learning.
2) **Model choice.** LSTM model was not able to correctly classify different classes. Losses values showed LSTM was overfitted from the early epochs and only learnt to classify all instances as instances belonging to one class. At the same time basic classifiers were able to distinguish one class from another with better AUC values up to 0.62.
3) **Balancing techniques.** AUC metrics indicated there is no significant difference in performance of models with applied o balancing techniques. It emphasizes the idea of data complexity - simple balancing techniques were not able to handle the imbalance by deleting instances of majority classes or detecting and generating features of minority classes within the representation approaches applied in the study,

Dataset	Options			f1-score	AUC	True Positive	True Negative	False positive	False Negative
CoLa	BERT	Ridge ClassifierCV	No balancing	0.81	0.62	637	116	206	84
CoLa	BERT	LSTM	No balancing	0.81	0.66	721	0	0	322
CoLa	Doc2Vec	Quadratic Discriminant Analysis	No balancing	0.59	0.55	361	195	360	127
CoLa	Doc2Vec	LSTM	No balancing	0.81	0.5	721	0	0	322
CoLa	BERT	Linear Discriminant Analysis	Undersampling	0.65	0.6	406	207	315	115
CoLa	BERT	LSTM	Undersampling	0.81	0.5	721	0	0	322
CoLa	Doc2Vec	Passive Aggressive Classifier	Undersampling	0.62	0.49	414	131	307	191
CoLa	Doc2Vec	LSTM	Undersampling	0.81	0.5	721	0	0	322
CoLa	BERT	NuSVC	Oversampling	0.79	0.61	613	123	108	199
CoLa	BERT	LSTM	Oversampling	0.81	0.5	721	0	0	322
CoLa	Doc2Vec	Extra Trees Classifier	Oversampling	0.78	0.49	646	33	75	289
CoLa	Doc2Vec	LSTM	Oversampling	0.81	0.5	721	0	0	322
SST-2	Doc2Vec	NuSVC		0.70	0.58	424	88	20	340
SST-2	Doc2Vec	LSTM1		0.67	0.5	444	0	0	428
SST-2	Doc2Vec	LSTM2		0.67	0.5	444	0	0	428
SST-2	Doc2Vec	LSTM3		0.67	0.5	444	0	0	428
SST-2	Doc2Vec	LSTM4		0.67	0.5	444	0	0	428
SST-2	Doc2Vec	LSTM5		0.67	0.5	444	0	0	428
SST-2	Doc2Vec	LSTM6		0.67	0.5	444	0	0	428
RTE	BERT	Extra Trees Classifier	Representation Subtraction	0.61	0.58	92	71	54	60
RTE	BERT	LSTM	Representation Subtraction	0.69	0.5	146	0	0	131
RTE	BERT	NuSVC	Representation Addition	0.51	0.49	74	64	72	67
RTE	BERT	LSTM	Representation Addition	0.69	0.5	146	0	0	131
RTE	BERT	SVC	Representation Concatenation	0.55	0.53	80	69	66	62
RTE	BERT	LSTM	Representation Concatenation	0.69	0.5	146	0	0	131

Table II: Metrics values of all experiments

B. SST-2

SST-2 classification is complicated with amount of data instances, but despite of this f1-score were achieved up to 0.70.

1) **Model choice** LSTM model was not able to correctly classify different classes. Losses values showed LSTM was overfitted from the early epochs and only learnt to classify all instances as instances belonging to one class. No additional parameters improved the performance. It indicates simple bidirectional LSTM with output layer is not enough to train a classifier.

1) RTE: RTE classification is complicated with input data representation, but despite of this f1-score were achieved up to 0.61.

1) **Sentences representation** Assumptions about the representation of two sentences resulted in weak performance of models. More advanced approaches should be applied to train a classifier.
2) **Model choice.** LSTM model was not able to correctly classify different classes. Losses values showed LSTM was overfitted from the early epochs and only learnt to classify all instances as instances belonging to one class.

IV. Discussion and Conclusion

The aim of this study was to apply not complex approaches to train classifiers for different text datasets and some datasets performed well - CoLA, while others showed more advanced techniques and model is better to be applied to obtain better performance.

Nevertheless it always worth to try to apply some simple approaches in order to find optimal trade-off between simplicity, interpretability, effectiveness and resource-consumption.

References

[1] M. AI, "Paper with code: Cola," 2022.
[2] M. AI, "Paper with code: Rte," 2022.
[3] M. AI, "Paper with code: Sst-2," 2022.