Sequence Classification Models for English Linguistic Acceptability
Machine Learning for Natural Language Processing 2021

Oscar Bouvier
ENSAE Paris
oscar.bouvier@ensae.fr

Rayane Hanifi
ENSAE Paris
rayane.hanifi@ensae.fr

Abstract

In this project, we evaluate performances of various deep learning architectures on a hard Sequence Classification problem, which is identifying grammatical acceptability of English sentences.

1 Problem Framing

Identifying the grammatical acceptability of a sentence is a difficult task, even for humans, which nevertheless seems very useful, for example in automatic correction of sentences on a message server. Indeed, solving this task represents a challenge in NLP because it requires models to build deep understanding of the sentences grammatical structure with quite uninformative update signal (binary loss). Moreover, it does not rely on the use of a particular vocabulary as other standard NLP classification problems (Sentiment Analysis).

We study the performances of various NLP architectures on this task, using standard dataset called CoLA[1](Corpus of Linguistic Acceptability) which is composed of 10657 sentences from 23 linguistics publications, expertly annotated for acceptability.

We both evaluate models quantitatively (using different evaluation metrics), and qualitatively by comparing the prediction performances on specific grammatical sentences forms.

2 Experiments protocol

The CoLA dataset is composed of an unabalanced training set of 8551 sentences (71% correct, 29% uncorrect), an in-domain testing set of 527 sentences (mostly using formal language) and an out-of-domain testing set of 516 sentences (mostly using informal language) with similar distributions.

[1] https://nyu-mll.github.io/CoLA/

For the study, we build 3 binary classifiers : an *LSTM* classifier using bidirectional pooling architecture (figure 1) , an *ELMO* 1D Convolutional classifier (Peters et al., 2018) using pretrained embeddings and a *BERT* classifier (Devlin et al., 2018) using pretrained embeddings.

The *LSTM* classifier will be considered as a baseline in our project. Furthermore, it will allow us to compare the performance of models using *semantic embeddings* (Word2vec, FastTextn, GloVe), versus *contextual embeddings* (as in *ELMO* and *BERT*).

Indeed, traditional NLP models were initially restricted to semantic word embeddings such as GloVe or word2vec, where every single word was linked to a vector. Still, it makes it difficult to capture all meanings, especially as contexts changed. Thus, we believe that contextual embeddings, as in *BERT* model, are much more relevant as we are interested in the grammatical logical structures occurring between sentence's words, rather than in the meanings of the words used.

For the *LSTM* classifier, we performed corpus-trained embeddings (*word2vec* and *FastText*). Since the used corpus is small (5718 unique words), we also used pretrained embeddings (*Glove*) trained on large wikepedia database corpus, to perform fairly against other pretrained embedding models. For the preprocessing part, we used the *TreeWordBank* tokenizer to separate the word contractions (e.g "won't" $\rightarrow$ "wo", "n't") that we believe essential to the sentence structure understanding. Finally, we used designed *BERT* tokenizer for this particular model.

We will first evaluate the model performances quantitatively using 4 evaluation metrics : accuracy, F0 score, F1 score (to get precise insights on the model predictions on correct or uncorrect sentences) and the Matthews Correlation Coefficient. This coefficient (having value between -1

and +1) which expresses the correlation between the observed and predicted binary classifications, is generally regarded as a balanced measure which can be used even if the classes are of very different sizes, and seems therefore particularly suited for this problem.

The learning rate for every training is 5e-5 and has been chosen to provide consistent convergences. The number of hidden LSTM layers for LSTM classifier is 5 with 128 units. For the ELMO model, we use on top of embeddings a 1D convolutional layer with 16 out channels. For BERT, we used attention masks and a maximum sequence length of 64. We assess for convergence 30 epochs for LSTM and ELMO, 5 epochs for BERT.

The experiments are available on Colab [2] and on Github [3].

3 Results

The results of experimentations are listed in the table 1. For presentation purposes, we only keep the FastText embedding method for *LSTM* classifier, as other embeddings methods showed no signical changes in the results obtained. Firstly, it comes as no surprise that *BERT* model outperforms the *LSTM* and the *ELMO* models for all the metrics used, as presented in 1 in Appendix. Regarding metrics, *BERT* shows an accuracy of 0.81 in Training set compared to around 0.70 for the other two models. The most performance gain is in the Matthews Correlation Coefficient and F0 score demonstrating the model's ability to correctly predict the "incorrect" class (which can be considered as critic class in this framework).

The other two models has indeed poor performances in terms of accuracy as they're competitive with "dumb classifier" accuracy (around 0.7), systematically predicting the most frequent class. However, we can notice that ELMO model is more accurate when predicted "incorrect class" while LSTM predicts very rarely the "incorrect" class (being close to "dumb classifier"). The gain of performance with ELMO is not surprising as it is a more sophisticated model but it demonstrates mostly the importance of contextual embeddings in this approach, supported by the fact that the different choices of semantic embeddings in LSTM had no significant impact on performances.

[2] Colab link
[3] Github link

Thus, the BERT model outperforms largely the other two model, especially in terms of MCC, and converge much more faster (only 5 epochs for convergence versus more than 20). However, it still faces the same problem (in lower proportions) as incorrect sentences seem to be more difficult to predict for the *BERT* classifier (see table 6 and 7).

Qualitatively, even if *BERT* performs well, it fails to detect errors in sentences which are long and complex grammatically. Indeed, using table 8, we realized that average number of words per incorrectly predicted sentences is higher. It should be noted that this trend is more prominent for "incorrect"-labelled sentences, supporting the fact that classifier struggles more in predicting "incorrect"-labelled sentences. Moreover, it also highlights a potential minor classification bias, leading the classifier to consider "incorrect" sentences to be short.

Using POS-tagging (figures 2 and 3), we also realized that sentences with a lot of pronouns, especially verbal pronouns, particles, auxiliaries and coordinating conjunction are the most difficult sentences to handle for the *BERT*. Indeed, some verbs can be used with a large variety of pronouns in English which makes it difficult to understand if a sentence is correct or not. At the opposite, sentences with many nouns and determiners seem to be easier to handle.

4 Discussion/Conclusion

To conclude, this project enables to see the various advances in NLP universe, by applying several increasingly innovative architectures on a hard-classification task based on grammatical understanding.

Eventually, we observe that recent contextual embeddings models outperforms traditional NLP models such as LSTM. Moreover, it appears that the finetuned *BERT* model is, as expected, by far the best-performing model in our study. These results demonstrate once again the relevance of Transformer architectures in their ability to build efficient contextual embeddings wich are powerful tools to model the subtleties of grammatical and semantic structure of language.

Observing that POS tags are strongly related to the results, we believe that the use of POS tagging based preprocessing or embeddings could be useful to improve the performance of the models, and is thus a potential direction for future work.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *CoRR*, abs/1802.05365.

A List of tables

Model	Training set				In-Domain Testing set				Out-of-Domain Testing set			
	Acc.	**F0**	**F1**	**MCC**	**Acc.**	**F0**	**F1**	**MCC**	**Acc.**	**F0**	**F1**	**MCC**
LSTM	0.692	0.15	0.812	0.072	0.700	0.209	0.814	0.148	0.664	0.164	0.790	0.042
ELMO	0.708	0.344	0.812	0.202	0.709	0.439	0.804	0.261	0.666	0.343	0.776	0.144
BERT	0.812	0.0.638	0.873	0.518	0.840	0.710	0.890	0.610	0.796	0.606	0.862	0.498

Table 1: Experiment results

Table 2: LSTM Confusion matrix - In-Domain

Predicted / True	**Incorrect**	**Correct**
Incorrect	21	141
Correct	17	348

Table 3: LSTM Confusion matrix - Out-of-Domain

Predicted / True	**Incorrect**	**Correct**
Incorrect	17	145
Correct	28	326

Table 4: ELMO Confusion matrix - In-Domain

Predicted / True	**Incorrect**	**Correct**
Incorrect	60	102
Correct	51	314

Table 5: ELMO Confusion matrix - Out-of-Domain

Predicted / True	**Incorrect**	**Correct**
Incorrect	45	117
Correct	55	299

Table 6: BERT Confusion matrix - In-Domain

Predicted / True	**Incorrect**	**Correct**
Incorrect	103	59
Correct	25	340

Table 7: BERT Confusion matrix - Out-of-Domain

Predicted / True	**Incorrect**	**Correct**
Incorrect	81	81
Correct	24	330

Predicted / True	**Incorrect**	**Correct**
Incorrect	6.93	9.56
Correct	8.64	7.85

Table 8: Average number of words per BERT predicted sentences

B List of Figures

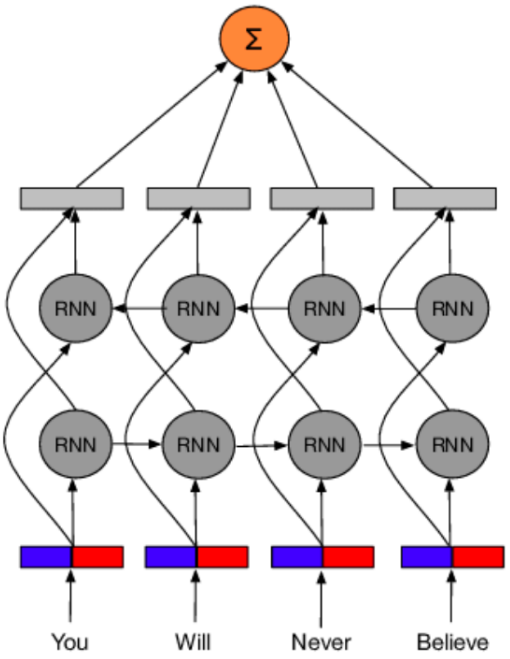

Figure 1: LSTM architecture

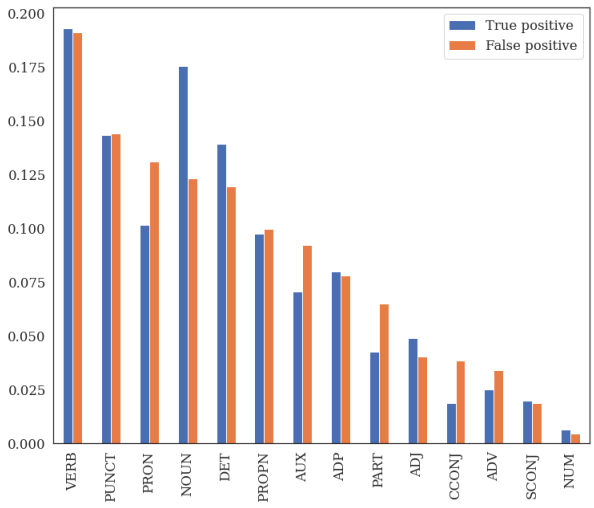

Figure 2: Word type distribution in positive BERT predictions (”Correct”)

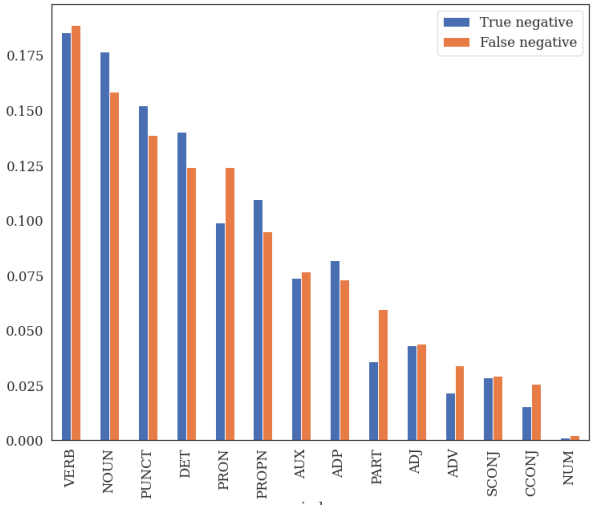

Figure 3: Word type distribution in negative BERT predictions (”Incorrect”)