Evaluating language knowledge of ELLs

Mihkel Tiks, Priidik Meelo Västrik

Institute of Computer Science, University of Tartu

Introduction

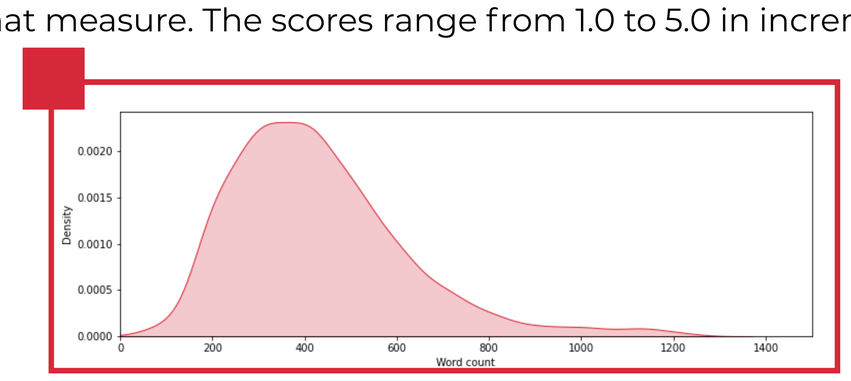
This project was put together as part of a competition with an aim to improve learning conditions for English Language Learners (ELLs). As English is the most studied language in the world, there is a lack of teachers and thus learners do not always get valuable feedback. This competition is about developing an algorithm to grade essays written by language learners. The results of the competition could help teachers by making grading essays easier and then students could practise writing exercises more than they are now. [1]

Data

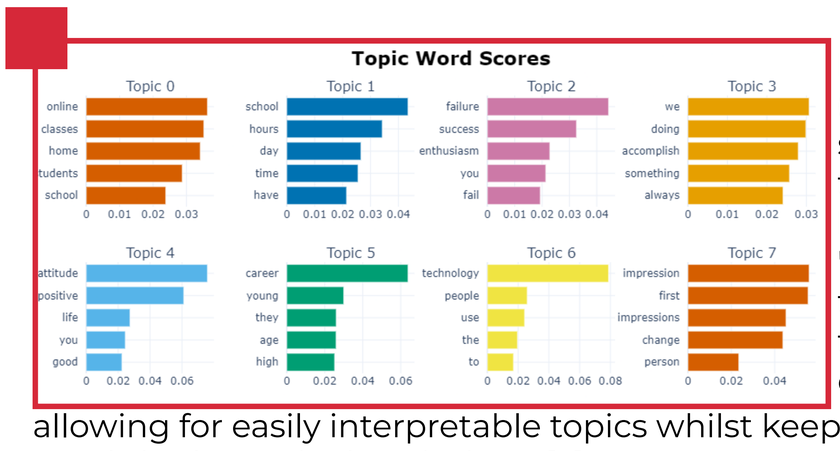
" The dataset comprises argumentative essays written by 8th-12th grade English Language Learners (ELLs). The essays have been scored according to six analytic measures: cohesion, syntax, vocabulary, phraseology, grammar, and conventions.

	text_id	full_text	cohesion	syntax	vocabulary	phraseology	grammar	conventions
0	0016926B079C	I think that students would benefit from learn...	3.5	3.5	3.0	3.0	4.0	3.0
1	0022683E9EA5	When a problem is a change you have to let it ...	2.5	2.5	3.0	2.0	2.0	2.5

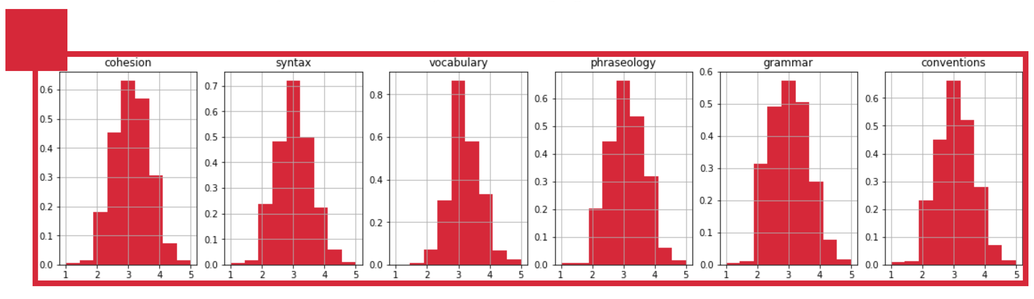
Each measure represents a component of proficiency in essay writing, with greater scores corresponding to greater proficiency in that measure. The scores range from 1.0 to 5.0 in increments of 0.5. "[1]



KDE plot of the essay lengths from the dataset.



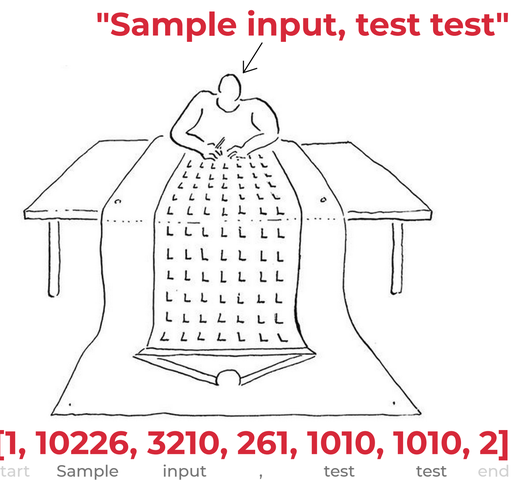
Most common (~200-300 samples each) topics found by BERTopic. 'BERTopic is a topic modeling that creates dense clusters allowing for easily interpretable topics whilst keeping important words in the topic descriptions.'[2]



Distributions of training dataset scores

Data preparation

'A tokenizer is in charge of preparing the inputs for a model. The [HuggingFace] library contains tokenizers for all the models'. [4]



The main component in data preparation in this project were the tokenizers. Since models need same format inputs when training and predicting, all texts that come out of the tokenizer need to be padded to a certain length. For example, if the maximum token length is 128 then the model will probably not be very accurate because it would only predict on that many words. On the other hand, large token lengths require more memory.

Results

The submissions were scored using MCRMSE (mean columnwise root mean square error) which on our best model ended up being 0.44 while the top of the competition ended up with around 0.43. Our performance likely could've been bettered some more with longer training, fine tuning and some other methods like ensembles. All of our models trained up to 10 hours.

Model	Score	Epochs	Tokens	Batch	Time (h)
deberta-v3-large	0.449285	9	1000	2	9.9
deberta-v3-base	0.453216	12	1000	2	3.4
deberta-v3-base	0.499741	12	100	2	1.3
distilbert-base-cased	0.503791	16	512	1	1.3
roberta-base	0.56192	6	512	2	0.7
distilroberta-base	0.576161	20	512	32	0.5
roberta-base	0.615119	10	50	2	0.7

Above are some of the better results from tests with different models (token - max sentence length, batch - batch size). These results were produced with kaggle GPUs. Early on already the deBERTa models performed much better than others so our main focus was on polishing parameters and improving that model.

As can be seen though - lighter distil and roberta models with less parameters and vocabulary size (roberta-base - 50k, deberta-v3-large - 128k) also perform reasonably well. Another interesting takeaway would be the accuracy change in reducing token size. Deberta, when evaluated with a sentence length of 100, 10 times smaller than the original, still outperforms lighter models and predicts with an error of 0.5. The same can be said for the roberta-base model with 50 tokens predicting only by roughly 50 words with only a 0.6 error rate. So perhaps in some aspects of essays, much of their quality can already be quite accurately predicted considering only a small sample of the text.

Tools

We used PyTorch Lightning to construct neural networks with the most important help being the HuggingFace library's pretrained transformers and models. 'Pretrained models when trained on the large corpus can learn universal language representations, which can be beneficial for downstream NLP tasks and can avoid training a new model from scratch.' [3] We experimented with several Bidirectional Encoder Representations from Transformers, BERT for short, which have become very popular in the recent years.

Resources

Figures: SANAA, Le Corbusier, Leon Krier
1. https://www.kaggle.com/competitions/feedback-prize-english-language-learning
2. https://maartengr.github.io/BERTopic/index.html
3 https://vitalflux.com/nlp-pre-trained-models-explained-with-examples/
4. https://huggingface.co/docs/transformers/main_classes/tokenizer