Project E13: LANGUAGE PROFICIENCY

Assess the language proficiency of 8th-12th grade English Language Learners

TEAM:
Priidik Meelo Västrik
Mihkel Tiks

Project repository: https://github.com/mmtiks/DSProject_2022

Task 2 - Business understanding

The goal of the competition is to improve learning conditions for English Language Learners. As English is the most studied language in the world, there is a lack of teachers and learners do not always get valuable feedback. The most affected skill is writing because these tasks take the most time to grade and thus teachers are not giving them to students enough. That is why this competition is about developing an algorithm to grade essays written by language learners. The results of the competition could help teachers with making grading essays easier and then students could practise writing exercises more than they are now. Also, grading essays would not be that subjective anymore and students could be more sure that they are graded fairly. The results will be scored using mean columnwise root mean squared error for all six aspects that the model has to grade. So the essays given to us are already correctly graded by humans and our model must grade as human-like as possible. In addition, the submissions are scored for efficiency where the runtime of the model is also taken into account. This category also only accepts submissions that do not have GPU enabled.

As we do not have prior experience with any data science competitions and natural language processing, we are not expecting to design an actually efficient and good model but more so just get an introduction into such topics and learn from the process. The competition allows us to use all available open-source material we can find. For example Hugging Face transformers and pretrained models such as DeBERTa. As the submission must be done as a Kaggle notebook, we can also use NVidia T4 x2 GPU for 30 hours a week provided for free by Kaggle. Also, according to competition rules our CPU and GPU runtimes must be below 9 hours and internet access must be disabled for the notebook.

After having looked into the problem and getting to know the system, our goal is to get an MCRMSE of 0.6 or lower on the private dataset.

Useful terminology:
neural network - a series of algorithms for recognizing relationships in a set of data through a process that mimics the human brain

pytorch - python-based scientific computing package for implementing neural networks

transformer - deep learning model that adopts the mechanism of self-attention

token - an instance of a sequence of characters that are grouped together as a useful

semantic unit

tokenizing - breaking text into units (tokens)

Hugging Face - a data science community platform that provides tools to build, train and deploy machine learning models based on open-source technologies

NLP (Natural Language Processing) - a branch of artificial intelligence concerned with giving computers the ability to understand text and spoken words

BERT (Bidirectional Encoder Representations from Transformers) - an open source machine learning framework for natural language processing

loss function - method of evaluation how well an algorithm models a dataset

Task 3 - Data understanding

We are given a training set with approximately 3900 argumentative essays and their grades for different aspects like cohesion, syntax, vocabulary, phraseology, grammar and conventions. The essays are written by 8th to 12th grade English Language Learners. The essays vary in size, some are only a few sentences, some contain even a thousand words. Although they are supposedly written by actual learners, some of them seem to be very random and contain unrecognisable syntax and words. This might just be because of the different level of the learners, as many of the essays just have typos in almost every word. Maybe the learners just have not paid attention to spelling and are not bothered by it or maybe some essays have been tampered with. We have also found that some of the essays contain lots of unnecessary whitespace which we will have removed for the model to be more efficient. In addition, the texts contained “newline” symbols and backslashes for apostrophes, which we would also have to move to improve the quality of tokenizing. A large majority of essays have about 500 symbols. The grades for each measure are given in increments of 0.5 from 1 to 5. Every prediction column values in the training dataset had a normal distribution which also represents the real-life distribution of grades normally. Also, the grades are correlated quite nicely and realistically - for example, if the syntax grade is 1 then the maximum grammar grade is 2. The training set does not contain any missing values so the datasets have good quality.

As we will use tokenizing we have to select a maximum token size and for faster training we may use only a certain length of the essays because processing very long essays might take up a lot of time. As over 90 % of the essays have less than a thousand symbols then we can only count the first 1000 symbols without losing much accuracy but gaining a lot of processing time.

We have used BERTopic to find some more used topics of the essays, or more like clusters. This process found only 40 topics, 20 of which had only a few instances, so most of the essays are about 20 topics. These topics were found by clustering the text by word/token count and so most of the topics are not very “logical”. For example, some of these are labelled “school hours day time”, “career young they

age” and such. From this we can get the most used words and find which words are frequently used together and thus maybe make some assumptions as to the vocabulary quality with this info.

Task 4 - Project plan

Understand tools, concepts and libraries:(time: both ~15 hours)

- PyTorch, Python, Pandas, HuggingFace…
- NLP, Neural Networks, Layers, Loss…
- Google Colab, Kaggle, GPUs…

Get familiar with the data: (time: both ~5 hours)

- Seek patterns, anomalies
- Evaluate data quality, balance etc…

Build a model and improve it: (time: both ~20 hours)

- Build prototype to fit general needs of the competition
- Find ways to improve model while staying in competition rules
- Construct one model using GPU(s)/TPU(s) and one without

Visualise and draw conclusions: (time: both ~5 hours)

- Statistics on data, best models, transformators, resources
- Plots, graphs, maps
- Final poster and presentation