MLDM Project Intermediate status report

Kuchinski Valeria, Rakhimzhanov Nurlan

November 2022

Dataset presents pairs of phrases (an anchor and a target phrase). The task is to rate how similar they are on a scale from 0 (not at all similar) to 1 (identical in meaning).

Score meanings

The scores are in the 0-1 range with increments of 0.25 with the following meanings:

1.0 - Very close match. This is typically an exact match except possibly for differences in conjugation, quantity (e.g. singular vs. plural), and addition or removal of stopwords (e.g. “the”, “and”, “or”).

0.75 - Close synonym, e.g. “mobile phone” vs. “cellphone”. This also includes abbreviations, e.g. "TCP"→ "transmission control protocol".

0.5 - Synonyms which don’t have the same meaning (same function, same properties). This includes broad-narrow (hyponym) and narrow-broad (hypernym) matches.

0.25 - Somewhat related, e.g. the two phrases are in the same high level domain but are not synonyms. This also includes antonyms.

0.0 - Unrelated.

1 Classical ML

In the first section we provide the problem description, data analysis and its normalization. To understand the similarity between the anchor and target, first, we begin with baseline methods like Jacard coefficient,

Euclidean and Cosine distances. The calculations were based on BOW and TF-IDF vectors. The results obtained do not show excellent semantic similarity between anchor and target because either bow and tf-idf vectors do not reflect semantic meaning of words.

Next, we decided to look at the project problem from the other side. Let us consider the following problem:

In dataset the score has 5 "levels"of similarity. We can interpret the semantic similarity problem as a multi classification task, where score is the target and features are distances calculated based on bow vectors of anchor and target. However, the results are very bad. Because as we mentioned earlier bow vectors do not reflect the context of phrases. BoW also doesn't consider the semantic meaning of words. Furthermore, there are a lot of duplicates in data, so the results of classification are bad. We decided not to implement the logistic classification, because tf-idf vectors still do not represent the semantic meaning of words. So the results would be more or less the same as in the previous section (bow).

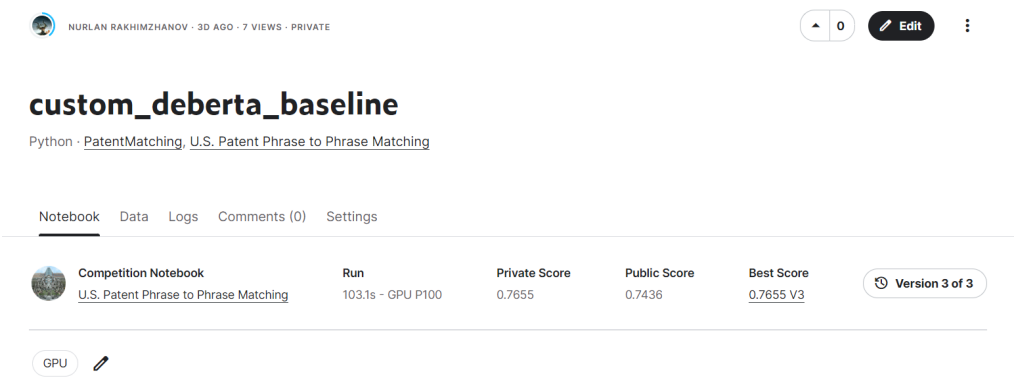
Vectors which can reflect the semantic meaning of words are embeddings. To obtain embeddings we used the FastText model. To get the embedding for a phrase we obtained embedding for each word in the phrase separately and then calculated the mean embedding. The results are better than calculated distances based on bow and tf-idf vectors, however the results obtained are not excellent, because here some examples of targets are very unusual. So it would be better to take into account the context of the patent's name. This information will be included in the next section where we use neural networks.

2 Fine-Tuning NN

For fine-tuning we used DeBERTa-v3-small model. Since right now deberta is one of the best pretrained models, because of the use of special kind of attention and ELECTRA-style pretraining. Main results is pipelines for fine-tuning deberta and inference in kaggle environment (since we chose notebook competition)

What was implemented:

- Found and downloaded dataset of decodings of "Context"column, which adds more information for model
- Used HuggingFace AutoTokenizers and AutoModels with torch datasets and dataloaders
- Changed task from binary classification, to multiclass
- Final model consisted of pretrained deberta, attentive pooling and fully-connected layer, with crossEntropy loss and SGD optimizer
- Trained for 3 epochs
- Figured out how to load notebooks instead of just predictions on kaggle, especially when you can't load pretrained model from internet



3 Future Plans

- Experiment with another uses of embeddings
- Full train implemented model
- Change target to binary
- Learn to use HuggingFace Trainer Api
- Experiment with different acrhitectures

Note: Since we managed to understand main things about pipelines, experimenting with things above is not a big work, just time for training.