Background

- Team Name: EPICURUS
- Private Leaderboard Score (Main Track): 0.7645
- Private Leaderboard Place (Main Track): 1st
- Private Leaderboard Score (Efficiency Track): -0.929755
- Private Leaderboard Place (Efficiency Track): 1st
- Name: Ahmet Erdem
- Location: Istanbul, Turkey
- Email: ahmeterd4@gmail.com
- I have BSc degree in Computer Engineering from Bogazici University and MSc in Artificial Intelligence from KU Leuven.
- I have 7 years of Data Science experience and I had joined many Kaggle competitions before.
- The competition problem is very similar to what I am currently doing for my mobile app, Epicurus. Therefore I wanted to join this competition.
- I think I have spent like 100 hours in total.

Solution

Pipeline:
Candidate Selection (Retriever methods) -> Feature Engineering -> Lightgbm -> Postprocessing

Validation Scheme:

- 1 fold validation
- All source topics and random 67% of the other topics are selected for the training set. The rest are validation topics.
- The contents which only match with validation topics are excluded from the training set. For evaluation, validation topics are matched with all the contents and competition metric is calculated.
- While training lightgbm model on the candidates, group4fold on topic_id is used on the validation set. Evaluation is done on the whole validation set afterwards.

At the end of the competition, I had 0.764 validation score and 0.727 LB. While it is a big gap, improvements in my validation score were almost always correlated with LB. And I got my validation score as my Private LB score, which I didnt expect.

Efficiency model got 0.718 validation, 0.688 Public LB, 0.740 Private LB and around 20 minutes CPU run-time.

Topic/Content Representation

Each topic is represented as a text using its title, its description and its ancestor titles up to 3 parents above in the tree. Example:

'Triangles and polygons @ Space, shape and measurement @ Form 1 @ Malawi Mathematics Syllabus | Learning outcomes: students must be able to solve problems involving angles, triangles and polygons including: types of triangles, calculate the interior and exterior angles of a triangle, different types of polygons, interior angles and sides of a convex polygon, the size and exterior angle of any convex polygon.'

Each content is represented as a text using its title, its kind and its description (its text if it doesn't have a description). Example:

'Compare multi-digit numbers | exercise | Use your place value skills to practice comparing whole numbers.'

Candidate Selection

TFIDF
Char 4gram TFIDF sparse vectors are created for each language and matched with sparse_dot_topn, which is a package I co-authered (https://github.com/ing-bank/sparse_dot_topn) It works very fast and memory efficient. For each topic, top 20 matches above 1% cosine similarity are retrieved.

Transformer Models
I used paraphrase-multilingual-MiniLM-L12-v2 for efficiency track and ensemble of bert-base-multilingual-uncased, paraphrase-multilingual-mpnet-base-v2 (it is actually a xlm-roberta-base) and xlm-roberta-large for the actual competition.

- Sequence length: 64. But only the first half of the output is mean pooled for the representation vector. Last half is only fed for context. This worked the best for me.
- Arcface training: Training contents are used as classes. Therefore topics have multiple classes and l1-normalized target vectors. The margin starts with 0.1 and increases linearly to 0.5 at the end of 22 epochs. First 2 and last 2 epochs have significantly lower LR. Arcface class centers are initialized with content vectors extracted from pretrained models. Each epoch takes around 4 minutes to 18 minutes depending on the size of the model.
- Ensemble method: Concatenation after l2 normalization

Models are re-trained with whole data for submission at the end.

Top 20 matches within the same language contents are retrieved.

In addition, for each topic, its closest train set topic is found and its content matches are retrieved as second degree matches.

Matches from Same Title Topics

For each topic, train set topics with the same title are found and their matched contents are retrieved.

Matches from Same Representation Text Topics

For each topic, train set topics with the same representation text are found and their matched contents are retrieved.

Matches from Same Parent Topics

For each topic, train set topics with the same parent are found and their matched contents are retrieved.

All retrieved topic-content pairs are outer joined.

Feature Engineering

- tfidf match score
- tfidf match score max by topic id
- tfidf match score min by topic id
- vector cosine distance
- vector cosine distance max by topic id
- vector cosine distance min by topic id
- topic title length
- topic description length
- content title length
- content description length
- content text length
- content same title match count
- content same title match count mean over topic id
- content same representation text match count
- content same representation text match count mean over topic id
- content same parent match count
- content same parent match count mean over topic id
- topic language
- topic category
- topic level
- content kind
- same chapter (number extracted from the text)
- starts same
- is content train
- content max train score
- topic max train score
- is content second degree match

Lightgbm Model

- Hit or miss classification problem
- Overweight hit (minority) class
- Monotonic constraint and 2x feature contribution on most important feature: vector cosine distance
- 2 diverse lightgbms: Excluded features which will potentially have different distribution on real test set in one of the models, vector cosine distance min by topic id. Also used slightly different parameters and kfold seed.

Postprocess

Postprocessing was very important. Using relative probabilities (gaps with highest matches) and using different conditions for train and test set contents were the key. While matching train set contents was like a classification problem, matching test set contents was like an assignment problem.

Topic-content pairs are included if they have one of the conditions below:

- Content has the best matching probability among other contents for the given topic.
- Content is among the train set contents and has above 5% probability and has less than 25% gap with the highest matching probability in the given topic.
- Content is among the test set contents and has less than 5% gap with the highest matching probability in the given topic.
- Content is among the test set contents and the topic is its best match and its total gap* is less than 55%.