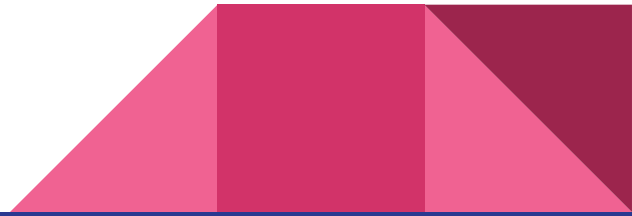
Document Ranking Using SBERT

Jenna Ingels, Robbe Lauwers

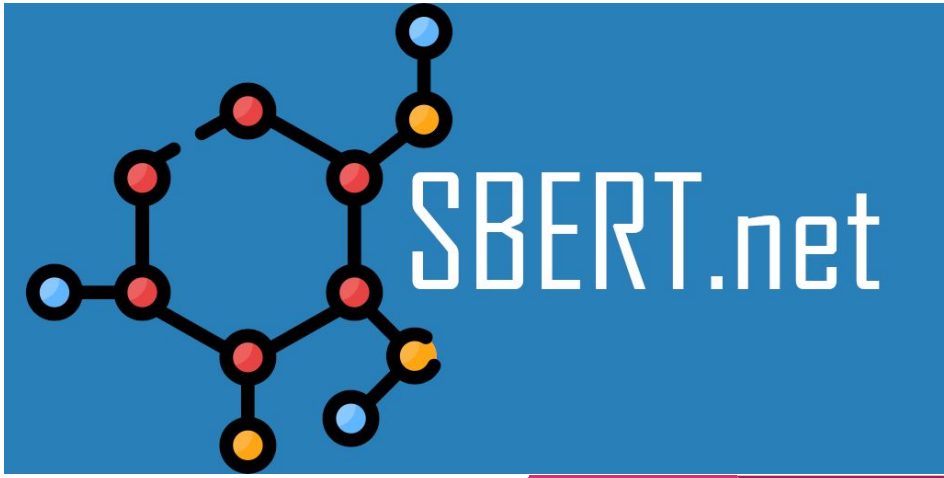
Background

Motivation

- Lucene Insufficient
- New Technology: Neural Networks

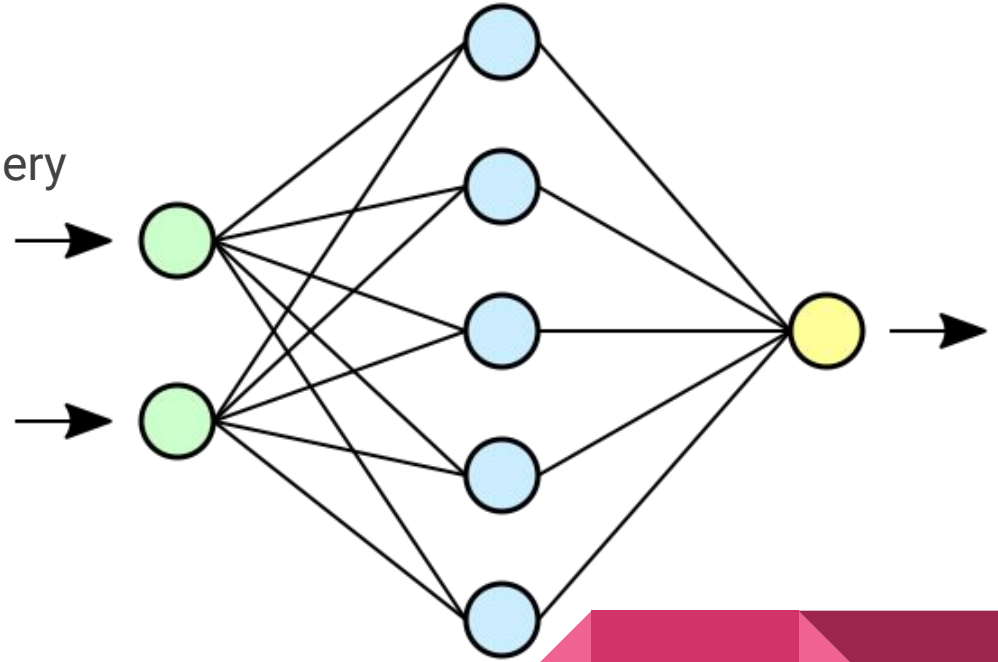


- Natural Language Processing framework
- Based on BERT by Google



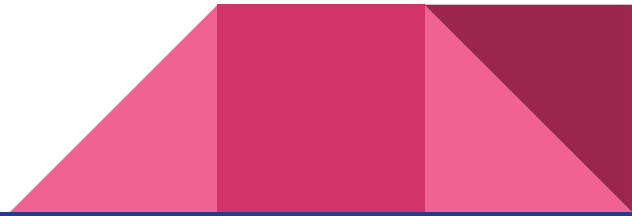
Overview

1. Teach Neural Network to rate files
2. Apply Neural Network to search query
3. ???
4. Profit



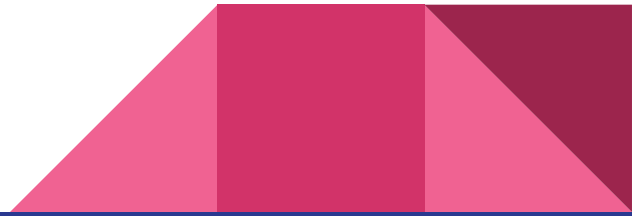
Teach Neural Network to rate files

- Import training data
- Initialise SBERT to a pre-trained model
- Train model on training data



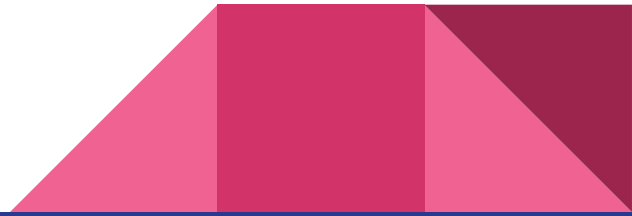
Apply Neural Network to search query

- Make a list of query-document pairs
- Let the Network score how much they match



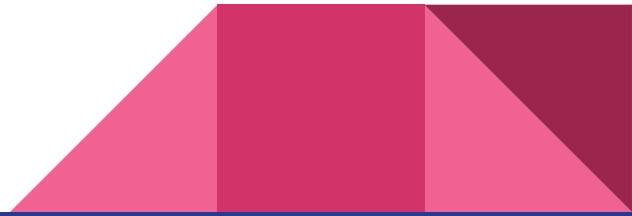
???

- The working of neural networks are poorly studied
- They are used for their results, not for their transparency



Profit

- Network finishes running
- Output is a list of scores
- Rate scores with development data

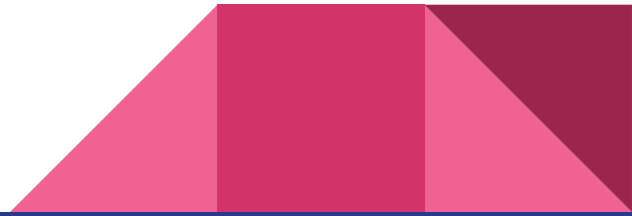


Implementation

Cross-Encoders

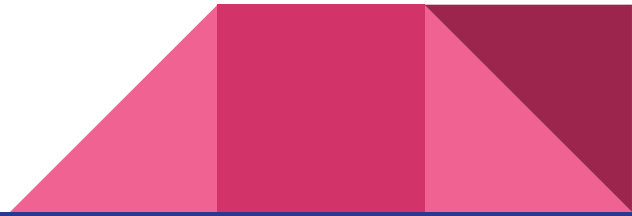
- High Performance
- Poor Efficiency

-> Use to score documents retrieved by Lucene



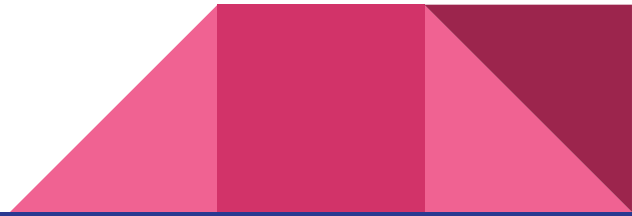
Models

- SBERT provides many pre-trained models
- Accuracy VS Speed
- Testing to find balance



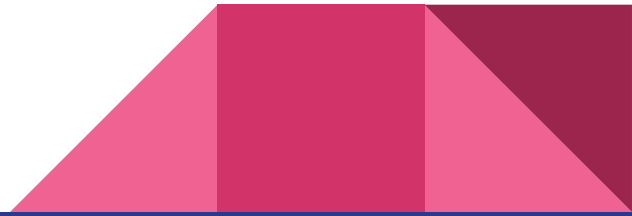
Rank Biased Overlap

- Compares ranked lists
- Emphasis on head of list
- Use to compare input and results



Precision/Recall

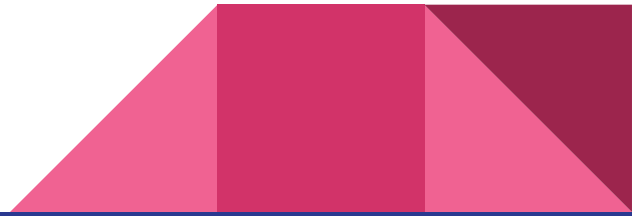
- Task was not only ranking, also predicting 0 or 1
- Test accuracy of neural network



Experiments

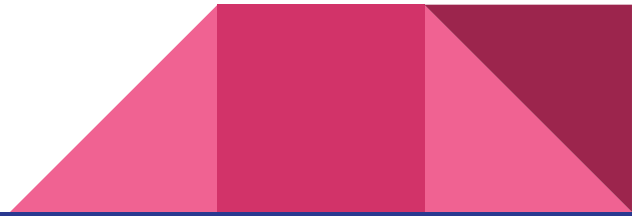
Method

- Choose pre-trained model
- Finetune on (part of) provided data
- Precision/recall



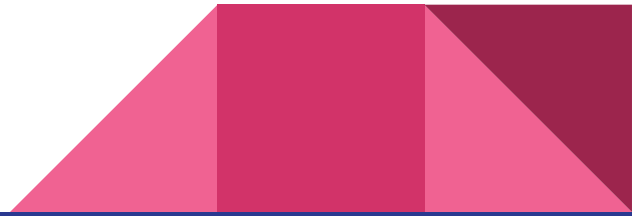
Hardware

- Google Colab
- NVIDIA Tesla K80
- Can compare training times
- Usage limits



Pre-trained models

- Different purposes
  - Duplicate questions
  - Information retrieval
- Different amount of levels
  - more: slower, more accurate

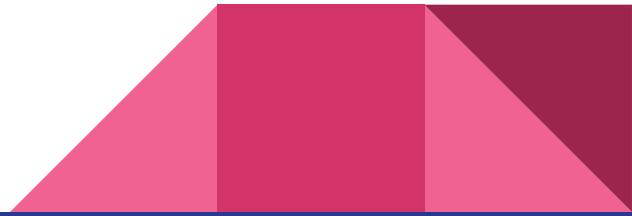


Pre-trained models

- Initially: MS MARCO TinyBERT
  - cross-encoder/ms-marco-TinyBERT-L-2-v2
  - Bing
  - Fast
  - Poor results
- SQuAD (QNLI)
  - cross-encoder/qnli-distilroberta-base
  - Wikipedia questions
  - Slower

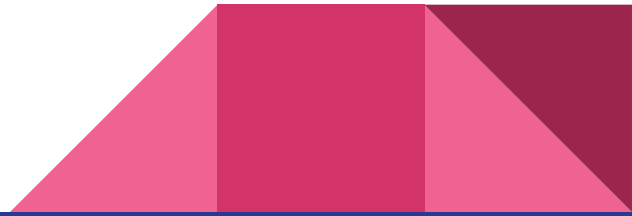
Epochs

- Iterations through dataset
- More epochs:
  - More accurate
  - 2x epochs, 2x time



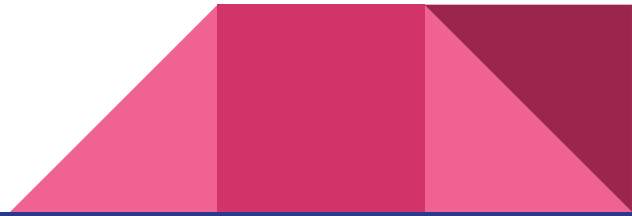
Maximum length

- Documents are large (~1000 words)
- Limit amount of tokens considered
- Adds some computing time
- Better precision, lower recall



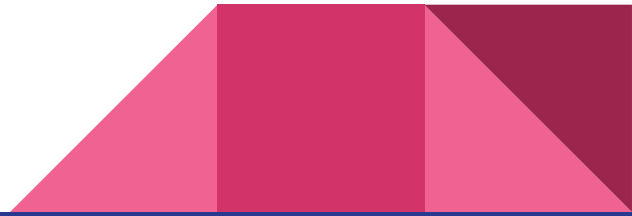
Speed

- MS MARCO TinyBERT
  - All data, 1 epoch, length 256
    - 03:18
  - All data, 1 epoch, length 512
    - 03:58
- SQuAD (QNLI)
  - All data, 1 epoch, length 512
    - 33:46



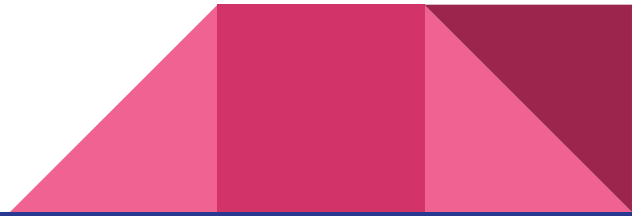
Evaluators

- Checks how accurate model is
- Dataset: 1 binary label
  - CEBinaryAccuracyEvaluator: poor recall
  - CEBinaryClassificationEvaluator: balanced
  - CERerankingEvaluator: similar to classification



Overfitting

- Training and precision/recall on same data
- Train on part of data, test on all
  - Significantly worse recall



Questions?