MSDS 631: Final project

US Patent phrase to phrase matching

Chandan Nayak, Jaysen Shi

Project goal

Key challenges faced by patent attorneys - finding relevant patents in the filing process.

Semantic similarity in patent listings but scored with the context of the application

Television -> TV Set

Strong Material -> Steel. But does context matter? In materials sciences, steel is strong material but in clothing denim could be a strong material.

EDA - train dataset

id	anchor	target	context	score
28300faae81045eb	imaging axis	axis optical path	G02	0.50
e2e3ea9e64465308	adjacent laterally	successive circumferentially	B23	0.25
19e674aaa3af1519	stationary rod	rod cutting	G01	0.00
e3c8e3c5c1b80024	panel frame	window	F24	0.25
b2e9380e766656a8	materially less	substantially less	H01	0.75
0607387b760e28f2	illumination condition	light rays	G03	0.50
909b934b663e1279	photocleavable linker	linking bond	C12	0.25
6443aa0f9a94b73f	resilient metal	resilient metal strips	A01	0.50
1e4069569c578273	pre trip	trip	F25	0.50
c68854453f00df47	board id	sequence code	H05	0.25

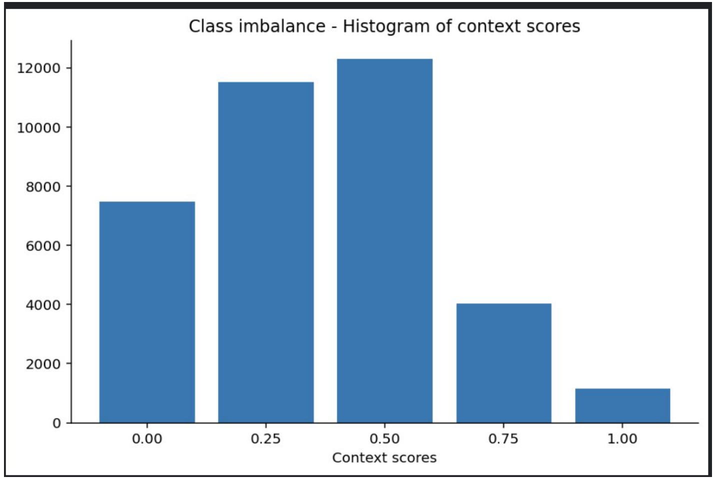
Train dataset has pair of phrases (an anchor and target) and a rating from 0 to 1 is provided based on context.

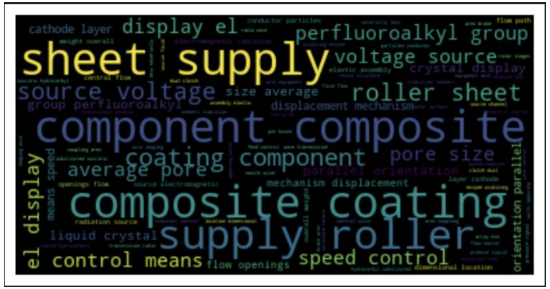
Total train samples: 36, 470

# of uniques values in ANCHOR column: 733

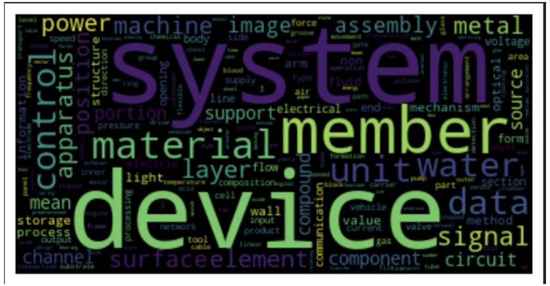
# of uniques values in TARGET column: 29,340

EDA - Imbalanced classes!





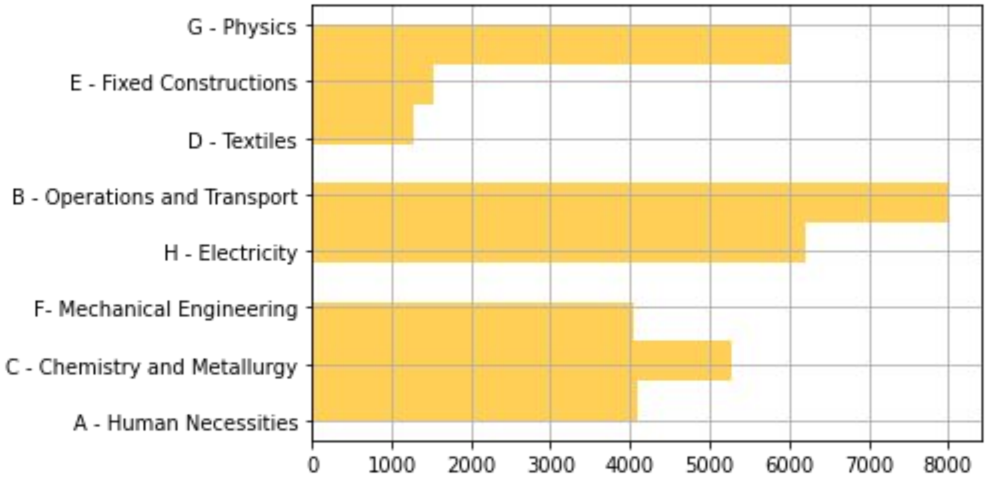
Word cloud for context



Word cloud for target

- Majority of context scores are <=0.5
- Same distribution across the eight different categories

EDA - train dataset (cont)



There are a total of eight major categories in the context field (denoted by first letter)

These are determined by Cooperative Patent Classication codes https://en.wikipedia.org/wiki/Cooperative_Patent_Classification

Initial approach - RNN based sequence model

Feature engineering: Convert the context codes to more rich text from the CPC code descriptions. This would give more text data to incorporate into our analysis.

id	anchor	target	context	score	code	title
37d61fd2272659b1	abatement	abatement of pollution	A47	2	A47	furniture domestic articles or appliances coff...
7b9652b17b68b7a4	abatement	act of abating	A47	3	A47	furniture domestic articles or appliances coff...
36d72442aefd8232	abatement	active catalyst	A47	1	A47	furniture domestic articles or appliances coff...
5296b0c19e1ce60e	abatement	eliminating process	A47	2	A47	furniture domestic articles or appliances coff...
54c1e3b9184cb5b6	abatement	forest region	A47	0	A47	furniture domestic articles or appliances coff...

Standard pipeline of text processing - stop word removal, lemmatization and stemming to create a dataloader class to feed into a GRU based sequence model.

Training input: anchor + target + title as an input sequence with the score converted to five classes.

Initial approach - RNN based sequence model (cont)

Model: 1 layer GRU network

Embedding size: 400

Hidden state size: 400

Learning rate: 0.01 with wd=5e-6 for 30 epochs, 0.0002 for 15 epochs

Loss function: Pearson loss

5-fold CV results:

```
[0.4757, 0.5077, 0.5216, 0.5379, 0.5105]
```

Initial approach - RNN based sequence model (cont)

What worked

- Learning rate annealing added ~ 0.01 to the score
- ReLU

What didn’t

- Stacking GRUs/RNNs
- BCELoss, MSELoss were harder to optimize

RNN-based models did **not** give us the results we were hoping for, so we had to come up with a better approach 😈😈😈

Final approach: patent BERT + 🤗

Google released a BERT model trained on corpus of US patents in 2020.

https://services.google.com/fh/files/blogs/bert_for_patents_white_paper.pdf

Key changes over traditional BERT - customised tokenization as patent language is meaningfully different from other text corpuses like Wikipedia. Example, BERT would have tokenized “prosthesis” as <pro><thes><is> but patent BERT uses the whole word as is. Researchers claimed that it improved the performance in masked word predictions.

But his was released as a tensorflow checkpoint and we found a pytorch checkpoint in Kaggle. The author did not give much details but it performed the best for us.

Final approach: patent BERT + 🤗

Training setup:

- GPU instance in Kaggle
- # Epochs: 5
- Batch size: 32
- Learning rate: 0.001
- Weight decay:0.01
- No changes made to transformer architecture
    - ● attention_probs_dropout_prob = 0.1
    - ● hidden_act: gelu
    - ● hidden_dropout_prob: 0.1
    - ● hidden_size: 1024
    - ● initializer_range: 0.02
    - ● intermediate_size: 4096
    - ● max_position_embeddings: 512
    - ● num_attention_heads: 16
    - ● num_hidden_layers: 24
    - ● max_seq_length: 512
    - ● max_predictions_per_seq:

Epoch	Training Loss	Validation Loss	Pearson
1	0.041600	0.024911	0.798570
2	0.024700	0.026484	0.832028
3	0.016900	0.021592	0.846081
4	0.011500	0.020069	0.850422
5	0.008500	0.019442	0.853039

Training results