CS-4400 Final Project

Procedure

For the purposes of the project I opted to go for using a Sentence transformer to generate embeddings and then combine the outputs using tensor flow's lattice regression.

Data Analysis

The first step before starting the project was to see how each column in the dataset contributed to the labels in a general sense. I created histograms for each column and then used a threshold to see how good the labeling was and where it was centered. It gave me a good idea of which columns could be used effectively.

Model Training

Title model

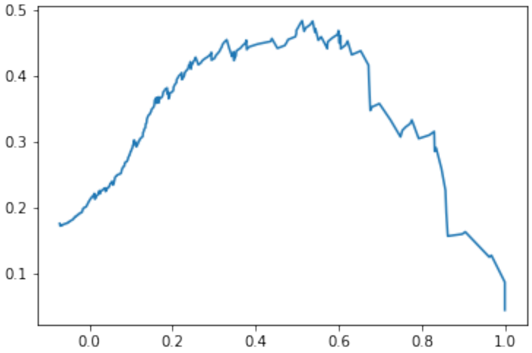
For training a model on the titles, I used the Sentence-transformers(sbert.net) library to get a good starting point. I used "paraphrase-distilroberta-base-v1" for my starting model weights and architecture and trained the model for 20 epochs. It gave me the following precision recall curves with the following statistics on the

Validation set:

Optimal Threshold 0.5121529698371887
Precision : 0.45098039215686275
Recall : 0.5227272727272727
F1 : 0.48421052631578954

On the test set:

True Positives : 22
False Positives : 37
True Negatives : 419
False Negatives : 22
Precision : 0.3728813559322034
Recall 0.5
F1 : 0.42718446601941745



F1-threshold curve

Category model

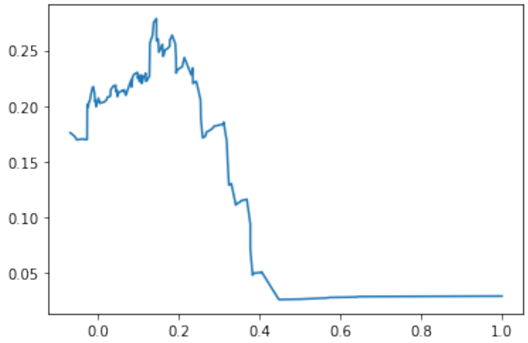
Similar to the title model, I trained another model using the same parameters on the categories in the dataset giving the following results on the

Validation Set:

Optimal Threshold 0.1443149596452713
Precision : 0.19298245614035087
Recall : 0.5
F1 : 0.27848101265822783

Test Set:
True Positives : 59
False Positives : 0
True Negatives : 0
False Negatives : 441
Precision : 1.0
Recall 0.118

F1 : 0.21109123434704832

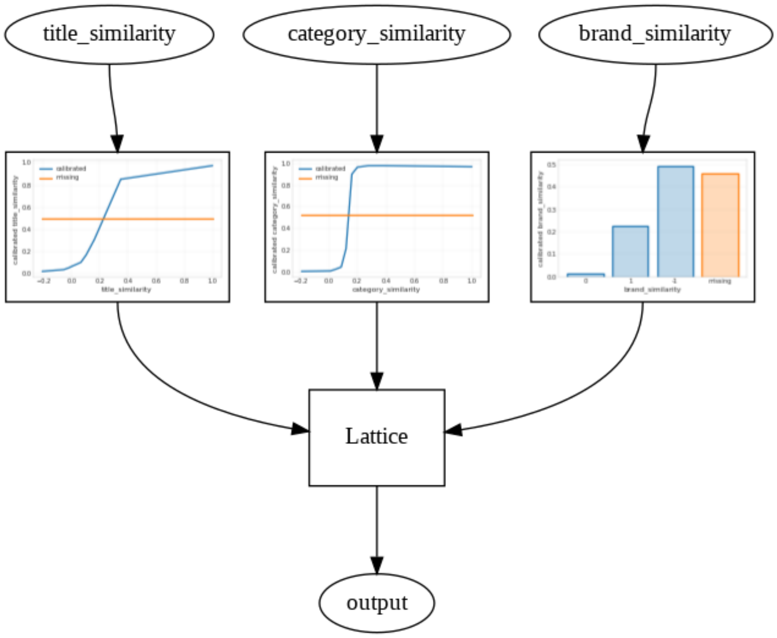


F1-threshold curve

To add to this I also calculated the brand similarity which was equal to if a given brand from ltable was equal to rtable.

Combining outputs from models

Since the individual models seemed to be giving a lower performance when used on a test set, I combined the outputs from the previous models and the brand similarity using a logistic regression model. For this I used tensor flows CannedClassifier which gave me an output using the similarities of the embeddings generated (using cosine similarity) in the previous model training. The combination looked something like the following generated image:



This gave me

Validation set:

```
Test Precision: 0.4307692348957062
Test Recall: 0.47457626461982727
Test F1 Score: 0.45161290252028835
```

Test set:

```
True Positives :  28
False Positives :  37
True Negatives :  404
False Negatives :  31
Precision :  0.4307692307692308
Recall  0.4745762711864407
F1 :  0.4516129032258065
```

Which as can be seen is a more stable output on a test set, leading me to believe the model at this point was not overfitting anymore.

Final output generation

Finally, the estimator and the trained models were used in conjunction to generate the final output on the dataset.

Before generating output, I used blocking on the brand to lessen the time of prediction given the time limit, which led me to predicting 2760 matches in the dataset.

The whole above procedure is stored as a google colab notebook and I would recommend viewing it there using the link provided below, however, as per the project requirements, I have uploaded the notebook as .ipynb file to a GitHub and put the link below.

Useful Links:

Dataset with ltable and rtable tuples along with similarity scores: https://drive.google.com/file/d/11AVSZvzHyK9cL4_Aa2aqr-slNtcSh2MK/view

Best title model: https://drive.google.com/drive/folders/106mX_VwXqpSgnyVcBhkQxnUh1WqD5dCC

Best Category model: https://drive.google.com/drive/folders/1-4ccOImMHnkYj_xTSY9dSvn1UgTJ70eT

Github link: https://github.com/Samyakkumar/CS4400_Final_project

Colab link: https://colab.research.google.com/drive/13EwfsTDocU4P2g523o7-PlvqHi0PnCOk?usp=sharing

Citations

1. **sbert.net**
2. **Scikit learn**
3. **Tensorflow**
4. **CS4400 sample solution**