# DUPLICATE QUESTION DETECTION – A SEMANTIC LEARNING APPROACH

*There are no stupid questions… only duplicates!*

Anvay Govind Pandit, Harinath Sundararajhan, Sarwesh Krishnan, Shibin Tazhe Vettil

## INTRODUCTION

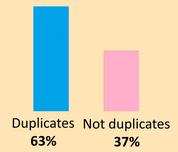
A **Duplicate Question** is a restatement of the first question using a different set of words. For example, "*What practical applications might evolve from the discovery of the Higgs Boson?*" and "*What are some practical benefits of discovery of the Higgs Boson?*" are duplicates.

A system to identify them will be particularly useful in social platforms like Quora, Stack Overflow, and Reddit. Duplicate questions provide a bad user experience as the answers get fragmented across different versions of the same question.

## DATASET

We utilized the Quora Question Pairs dataset for this project. Some key statistics:

- *Total number of question-pairs*: **404,290**
- *Total Number of Unique questions*: **537,933**
- *Train-test ratio*: **3:1**




## METHODOLOGY

In our quest to build the classifier system, we experimented on numerous approaches ranging from utilizing handcraft features to modelling deep learning representations.

**Handcrafted models:**

- *TF-IDF Character Bigram/Trigram Count + XgBoost*
- *LDA Topic Modelling + XgBoost*

**Deep learning approaches:**

- *Word Embedding + LSTM*
- *Char Embedding + LSTM*
- *Word Embedding + Bidirectional LSTM*

* In the above approaches the questions were represented with Glove 300D pre-trained word and char embeddings. These neural models had a Siamese structure intersecting on a custom layer with Manhattan Distance as the difference metric.

**Combined Approach:**

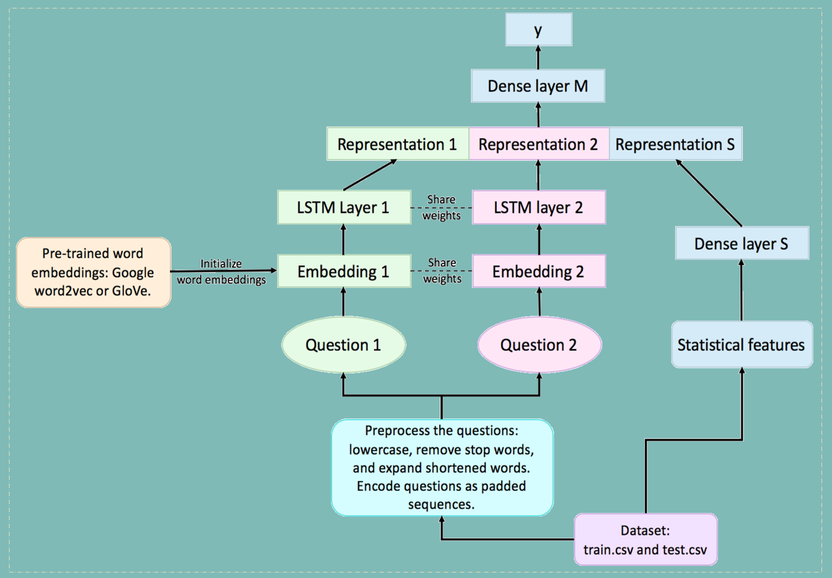
- *Word Embedding + Handcrafted TF-IDF Features + BiLSTM + Manhattan Distance*

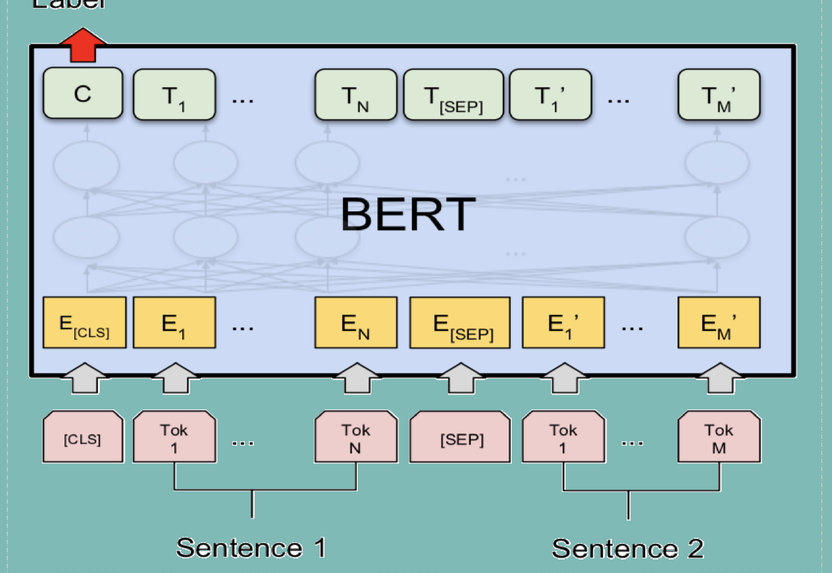
**Best Working Model:**

- *BERT Fine-Tuned Embeddings + Feed Forward NN*

**Other Approaches:**

- *TF-IDF Question Pair Similarity scores + SVM*
- *TF-IDF Question Pair Similarity Scores + XgBoost*
- *Word Embedding + Conv1D*
- *Char Embedding + Bidirectional LSTM*

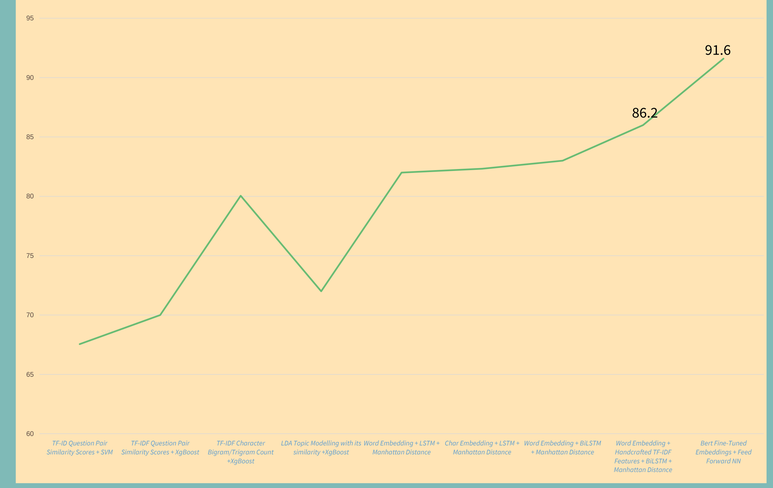



## REFERENCES

[1] **A Semantic Similarity Approach to Paraphrase Detection**
*Samuel Fernando and Mark Stevenson*
[2] **Distributed Representations of Words and Phrases and their Compositionality**
*Tomas Mikolov,Ilya Sutskever, Kai Chen, Greg Corrado and Jeffrey Dean*
[3] **Multi-Perspective Sentence Similarity Modelling with CNN's**
*Hua He,Kevin Gimpel and Jimmy Lin*
[4] **Learning Text Pair Similarity with Context-sensitive Autoencoders**
*Hadi Amiri,Philip Resnik,Jordan Boyd-Graber and Hal Daum´ III*

## CLASSIFICATION ACCURACY




## DATA PREPROCESSING

- TEXT EXPANSION
- PUNCTUATION PADDING
- STEMMING
- LEMMATIZATION
- SPELL AUTOCORRECT
- PUNCTUATION REMOVAL
- ABBREVIATION EXPANSION

## CHALLENGES

- Due to the model complexity of BERT, the inference and prediction of questions took a lot of time
- Due to the large amount of data corpus our training time was always on the higher side even on a GPU
- Data pre-processing was a challenge due because of abbreviations and text shortenings

## WHAT'S NEXT?

- Data Augmentation – flipping the question pairs
- Manually correct wrongly classified training samples
- Fine tune BERT – add LSTM instead of simple feed-forward structure