Identifying Quora Question Pair Duplicates

Enzo Blindow, Tom Dop (Quora-The-Explorer)

1. The Problem

"Where can I learn to speak English fluently?"

"What resources are available to learn perfect English?"	"Where can I learn to speak English fluently in India?"
Duplicate	Not Duplicate
Structurally **different**, but semantically **similar**	Structurally **similar**, but semantically **different**

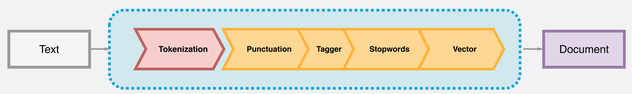
Our Ensemble Approach

Utilizing a combination of **LSTM**s (Long-Short-Term-Memory) and **NLP features**, feeding into various **classification** algorithms to predict duplicate questions.

2. Data

Binary classification problem with imbalanced classes

2.1 Preprocessing



2.2 Embeddings

3. Feature Engineering

Matching last character
questions ending in '?' more likely to be duplicate

Levenshtein Distance
number of edits required to equate strings

TFIDF
weighted differences

Shared Words

N-Grams
% difference in bi-grams, tri-grams, four-grams

Question Length

Topic Modelling
LDA generated topic similarity/dissimilarity

Shared Entites
"Why is Apple ORG so popular in India GRE ?"

Question Similarity
differences in context adjusted word vectors (SpaCy)

4. RNN Network with LSTMs

While traditional NLP approaches only tend to take **lexical differences** into account, LSTM cells can account for the **ordering** of words and may be better for measuring **meaning**.

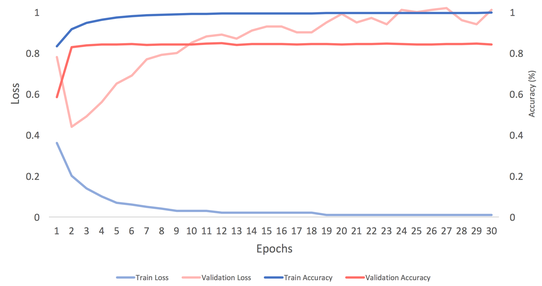
- Train / Crossvalidation Split: 20%
- Rebalancing of classes by **undersampling**
- **Siamese network** (two inputs) seem to perform well on semantic similarity tasks
- We tried both **subtracting** and **concatenating** the outputs from the two LSTM layers

4.1 Hyperparameter Tuning

	After 5 Epochs			
	Loss		Accuracy	
Description	Training	Validation	Training	Validation
Base	0.19	0.61	92.0%	80.3%
More Nodes	0.18	0.61	92.4%	81.9%
Bidirectional	0.15	0.68	93.8%	81.5%
Sigmoid	0.27	0.47	88.1%	80.9%
No dropout	0.11	0.53	95.6%	83.7%
Reweighted	0.33	0.59	88.1%	80.6%
Reweighted, Low dropout	0.17	0.88	94.4%	81.6%
Reweighted, No dropout	0.06	1.33	98.2%	82.2%
Subtract	**0.07**	**0.65**	**97.4%**	**84.1%**

Base Model 2 parallel LSTM layers (300 nodes), 2 Dense Layers (200 nodes), ReLu activation, 20% dropout

4.2 Model Performance



5. Classification Results

After combining outputs from LSTM and engineered features, we tried several different algorithms for classifying duplicate questions and compared results (Accuracy).

Neural Network	3 Dense Layers 200 nodes each, 20% dropout	**0.84515**
Logistic Regression		**0.84585**
Random Forest	max depth 2	**0.82639**
SVM	rbf kernel, c=1, gamma=1	(different set) **0.81001**

6. Summary

- Although we achieved a reasonable **final results (84.6% accuracy)**, our LSTM performance graph, suggests the model was **overfitting**
- While **accuracy** was used to judge the final result, **precision** and **recall** may have been a better measure of performance due to imbalanced classes
- Although we attempted to tune our model **hyperparameters**, we were limited by time and computational power and so were only able to test each model for **5 epochs**