Quora Question Pairs

Identify if two questions have the same intent

Agenda

1. Problem
2. Train & test data
3. Analyzing the data
4. Vectorizing the data
5. Extra feature selection
6. AI Models
    a. XGBoost
    b. Neural Network
7. Results

Problem

Given a pair of questions q1 and q2 we need to determine if they are duplicates of each other.

More formally:

Build a model that learns the function:
f(q1, q2) = 1 or 0

Train data

Question 1 - Question 2 - Answer

Question 3 - Question 4 - Answer

...

Question 400.904 - Question 400.905 - Answer

Test data

Question 1 - Question 2

Question 3 - Question 4

...

Question 2.000.108 - Question 2.000.109

Example

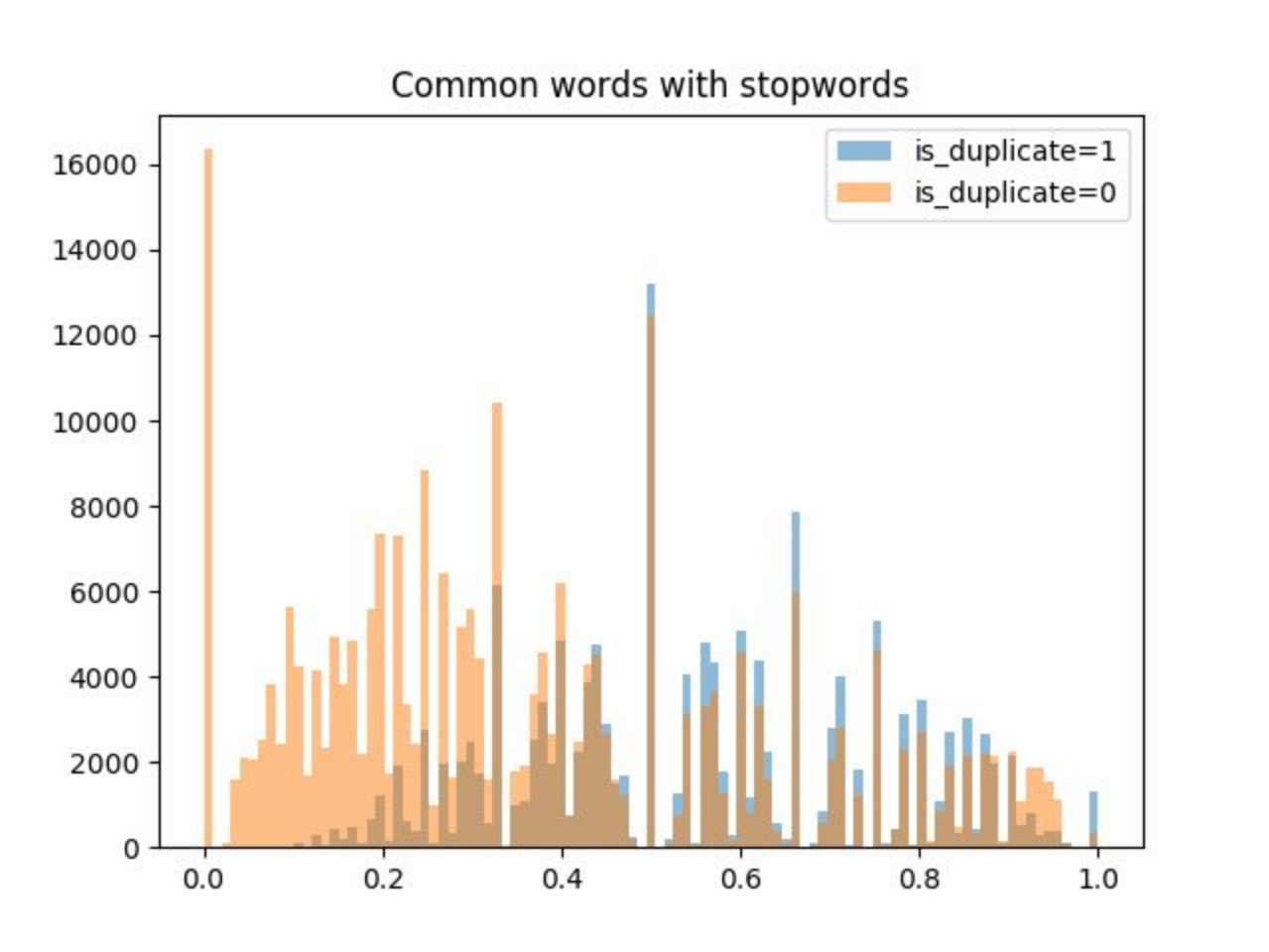
Could time travel ever be possible? - Will time travel ever be possible? - 1

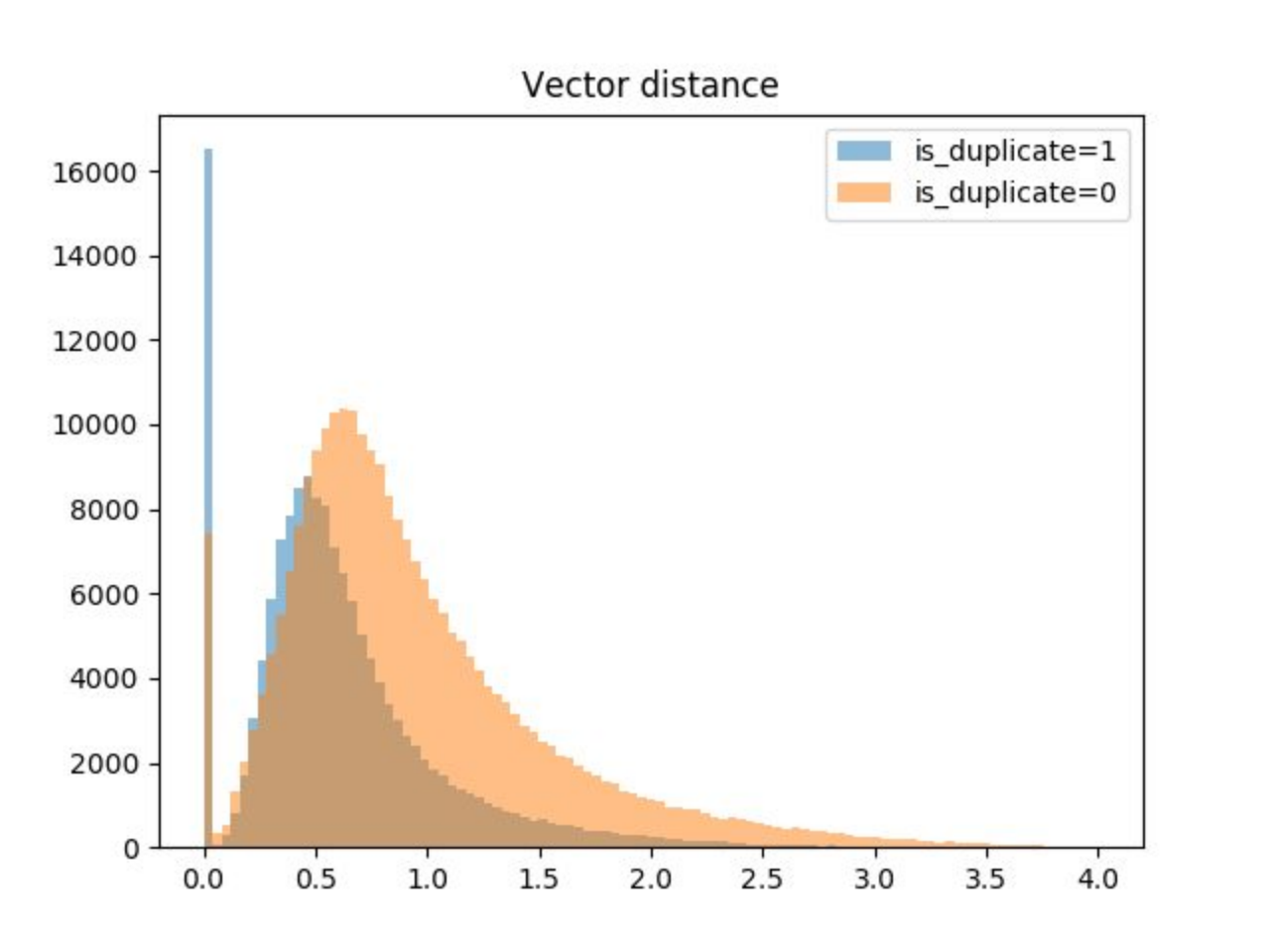
Why aren't blueberries blue? - Do rubber ducks quack? - 0

Analyzing the data

Needed to answer the question: How can a computer determine if two questions are duplicates?

What features makes a pair of questions more likely to be duplicates?


Common words with stopwords
is_duplicate=1
is_duplicate=0
16000
14000
12000
10000
8000
6000
4000
2000
0
0.0
0.2
0.4
0.6
0.8
1.0


Vector distance
is_duplicate=1
is_duplicate=0
16000
14000
12000
10000
8000
6000
4000
2000
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

Vectorizing

How do we perform calculations on strings?

Answer: By vectorizing it!

GloVe

Pre-trained vectors for English words.

Similar words placed closer in vector space, giving a sense of context.

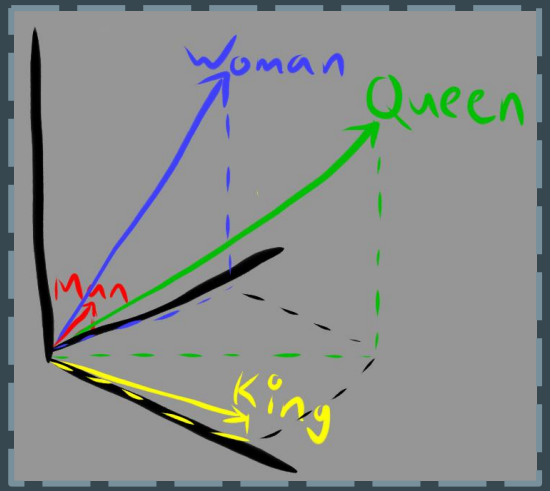
- GloVe 50d
- GloVe 100d
- GloVe 200d
- GloVe 300d

GloVe

King + Woman = Queen

glove(“King”) + glove(“Woman”) = glove(“Queen”)

[0.126, 0.043, ..., 0.321] + [0.421, 0.203, ..., 0.366] = [0.547, 0.246, ..., 0.687]



Extra Features

Basic Features:

- Length of question 1
- Length of question 2
- Length difference
- Nbr of words in question 1
- Nbr of words in question 2
- Number of common words
- ...

Distance Features (using GloVe vector space):

- Euclidian distance
- Manhattan distance
- Cosine distance
- Correlation distance
- Jaccard distance
- Chebyshev distance
- Hamming distance
- Canberra distance
- Braycurtis distance
- ...

Final vector

Adding everything together gives us a vector on following form:

[glove(Question 1), glove(Question 2), extra features]

= 115 dimensions

XGBoost

Stands for e**X**treme **G**radient **B**oosting
Gradient boosting is an approach which predicts the errors made by existing models and adds models until no improvements can be made

There are two main reasons for using XGBoost

- Execution speed
- Model performance

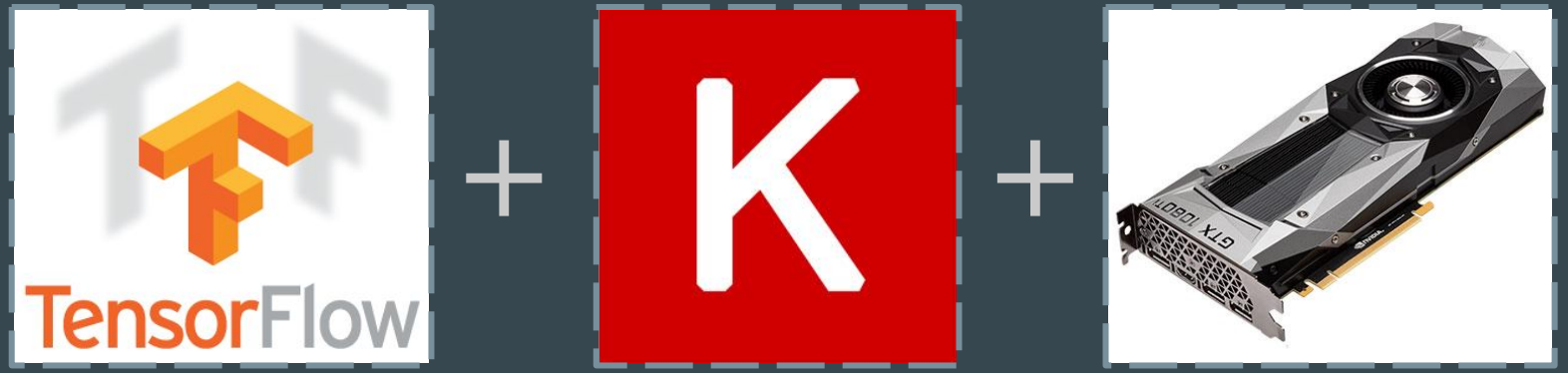
Have been shown to be the go-to algorithm for Kaggle competition winners

Result?

0.35660

Logarithmic loss

Neural Network



- Tensorflow - Open source machine learning library for python by Google
- Keras - Tensorflow API, additional abstraction layer.
- GPU acceleration support

Neural Network

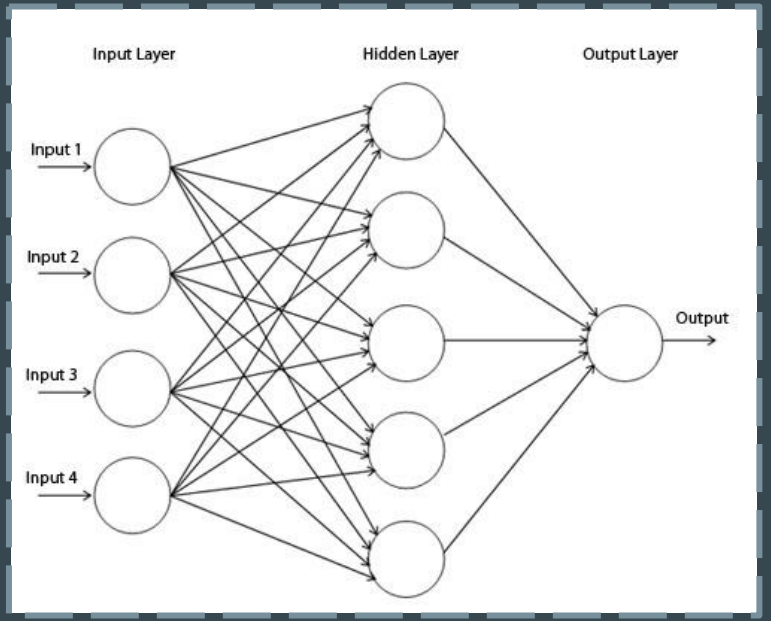
```
from keras.models import Sequential
from keras.layers import *

#Load data
x_train = np.load("../Data/LSTM_train_vector.npy")
y_train = np.load("../Data/labels.npy")
x_test = np.load("../Data/LSTM_test_vector.npy")

#Create LSTM neural network
model = Sequential()
model.add(LSTM(300, input_shape=(65,  115), activation='relu', return_sequences=True))
model.add(Dropout(0.5))
model.add(LSTM(300, input_shape=(65, 115), activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(1, activation='sigmoid'))
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])

#Train the model
model.fit(x_train, y_train, epochs=50, batch_size=1000)

#Test our trained model on the test set
predictions = model.predict(x_test)
```

Feed-Forward Neural Network

Input: GloVe vector, 115 neurons wide.

Weights: Edge weights between neurons updates automatically in the training phase.

Output: 1 neuron, value between 0 and 1.



Results

XGBoost: 0.35660

Feed-Forward Neural Network: 0.35354

1,257th place of 2,847 in Kaggle competition

Demonstration

Questions?