Duplication Detection in Questions

-Deepak Munagala
-Purusharth Dwivedi
-Avneet Kaur

Problem Statement

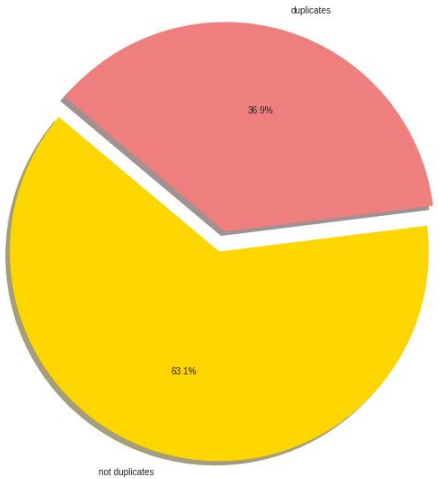
Given a pair of questions, the aim is design a model to be able to tell, whether they are semantically equivalent or not.

DATASET: Question pair dataset from Quora (404290, 6)

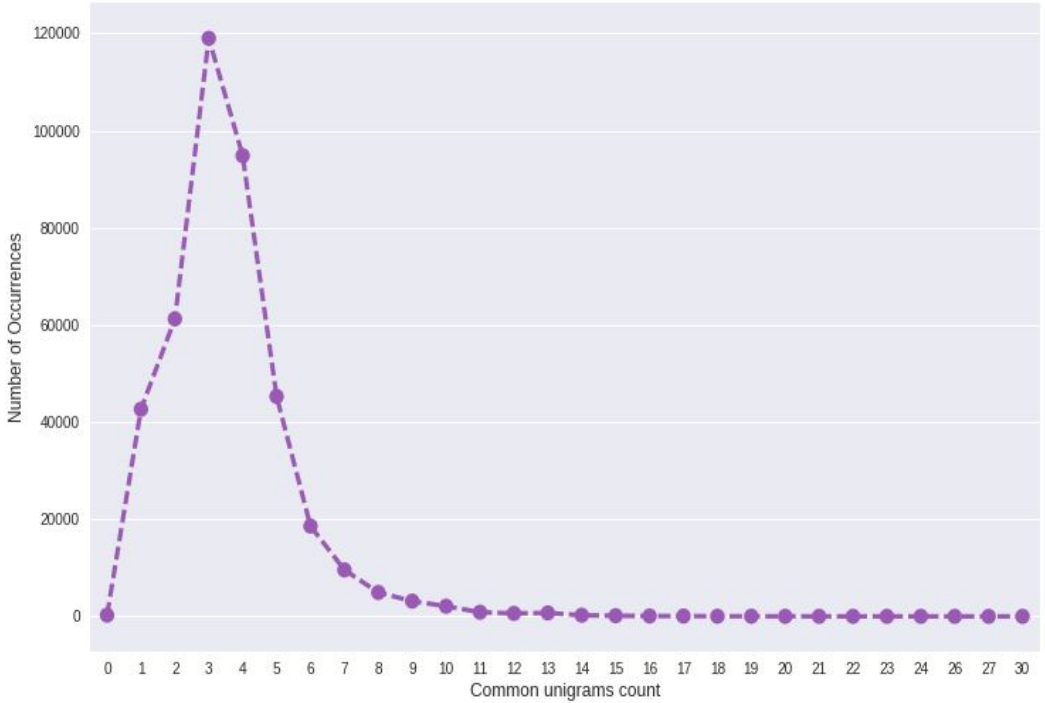
	id	qid1	qid2	question1	question2	is_duplicate
0	0	1	2	What is the step by step guide to invest in sh...	What is the step by step guide to invest in sh...	0
1	1	3	4	What is the story of Kohinoor (Koh-i-Noor) Dia...	What would happen if the Indian government sto...	0
2	2	5	6	How can I increase the speed of my internet co...	How can Internet speed be increased by hacking...	0
3	3	7	8	Why am I mentally very lonely? How can I solve...	Find the remainder when [math]23^{24}[/math] i...	0
4	4	9	10	Which one dissolve in water quikly sugar, salt...	Which fish would survive in salt water?	0

Explorartory Analysis

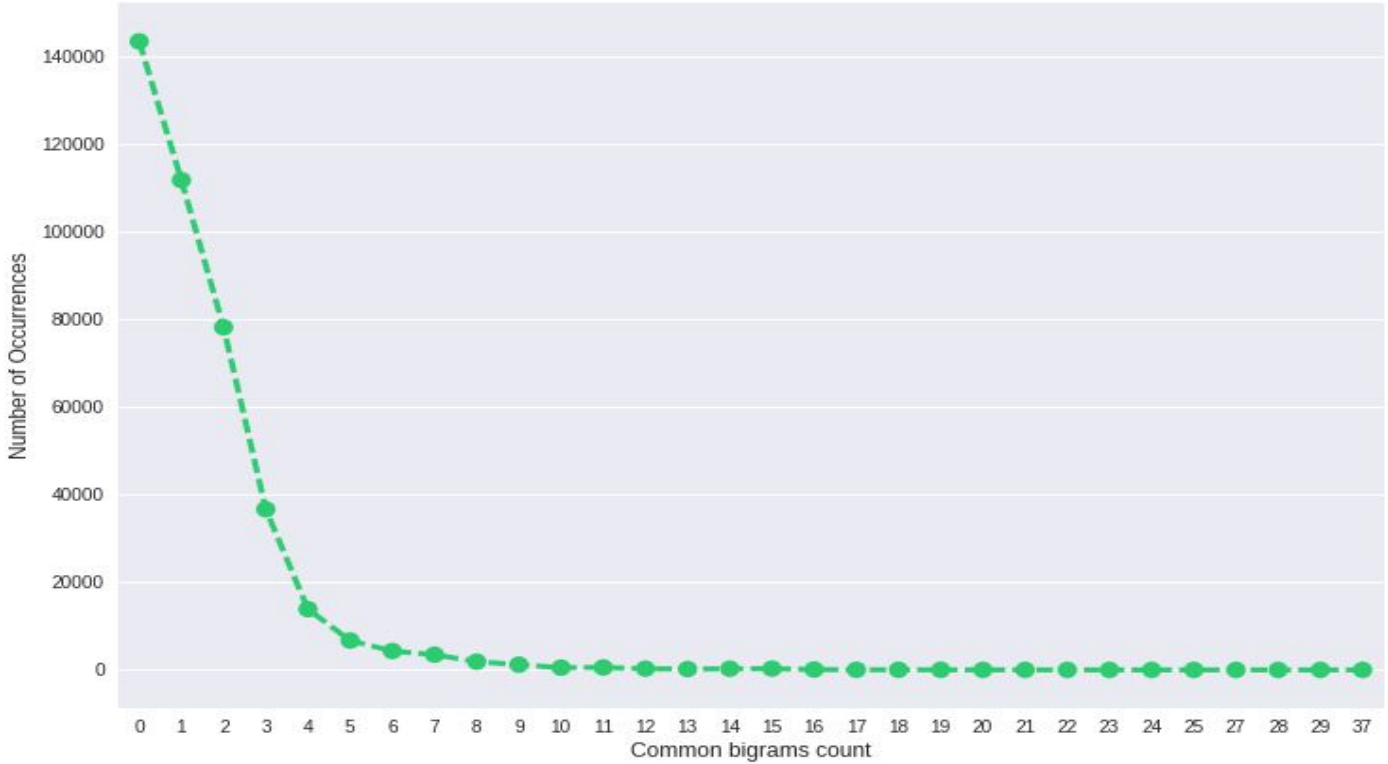
Duplicates/ Non Duplicates percentage



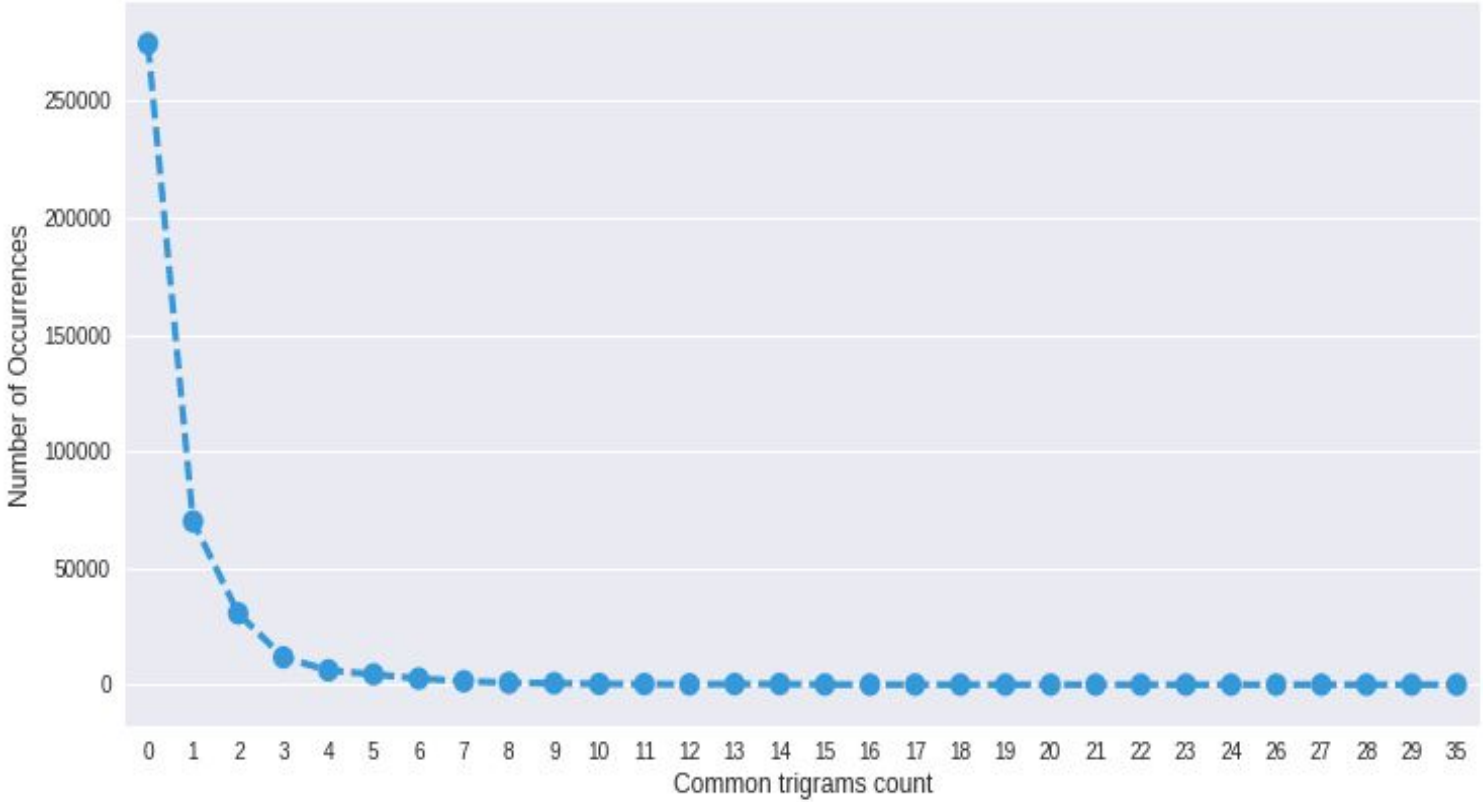
Common Unigrams



Common Bigrams



Common Trigrams



Methodology And Feature Extraction

Various textual features were extracted from the training data and fed into an XGBoost model for training.

The following measures were used:

→ **Semantic Similarity using WordNet corpus statistics.**
→ **Bag-Of-Words :**
    → **Word Overlap**: The number of shared words between both the questions normalised by the total number of the words in both questions. Unigram, Bigram, Trigram common ratios were calculated by finding the common unigrams, bigrams, and trigrams between a pair of questions
→ **Tf-Idf** : Tf-Idf representations of both the questions were calculated and cosine similarity was used as a measure.

WordToVec Model

- It takes as input a text corpus and outputs vector to each unique word
- Assigns vector of the size of vocabulary to each word
- Semantically similar words have almost same vectors
- We trained the model with Google news dataset (https://github.com/mmihaltz/word2vec-GoogleNews-vectors)

How we used WordToVec

WordToVec was used to get the vector representations of both the questions and following measures were calculated.
The vectors can be used to compute distance. Following measures were used-

- Word Mover’s distance,Cosine Similarity, City block distance(L1 norm), Euclidean distance, Canberra Distance, Bray Curtis Similarity

Features...

→ **Word Movers Distance:**
This distance between the two documents is the minimum cumulative distance that all words in document 1 need to travel to exactly match document 2. Given 2 sentences w1= a b c, and w2= d e f, WMD=Find min distance from a to d,e,f,Do same for b and c. Take sum of all these min distances.

→ **Cosine Similarity:**
Dot product of the two vectors divided by the product of the magnitude. (-1 to 1)

Features...

→ **Cityblock distance:**
Difference between the two vectors.(~manhattan distance)

→ **Euclidean distance:**
sum of squares of the individual differences between the elements of the vector.

→ **Canberra distance:**
Difference of the vectors, normalised by the sum of length of the vectors.

$$d(\mathbf{p}, \mathbf{q}) = \sum_{i=1}^{n} \frac{|p_i - q_i|}{|p_i| + |q_i|}$$

Semantic Similarity using Wordnet and Corpus Statistics

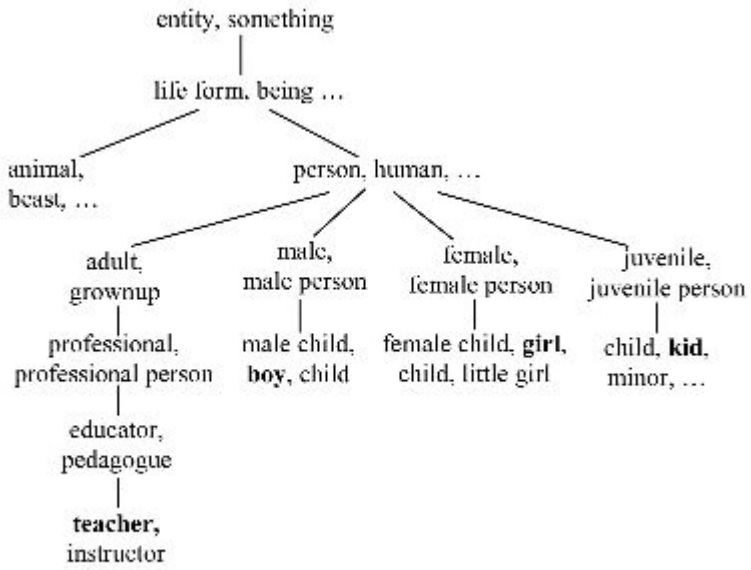


Fig. 2. Hierarchical semantic knowledge base.

WordNet is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into synonym sets .Using wordnet, we derive text similarity from **semantic** and **syntactic** information of both the sentences using the joint word set of two sentences. And computing the following measures:

- → word similarity
- → sentence similarity
- → word order similarity

And we use the above to compute the over all similarity using the formula.

S(T1,T2)=d*sentence_similarity+(1-d)word_order_similarity

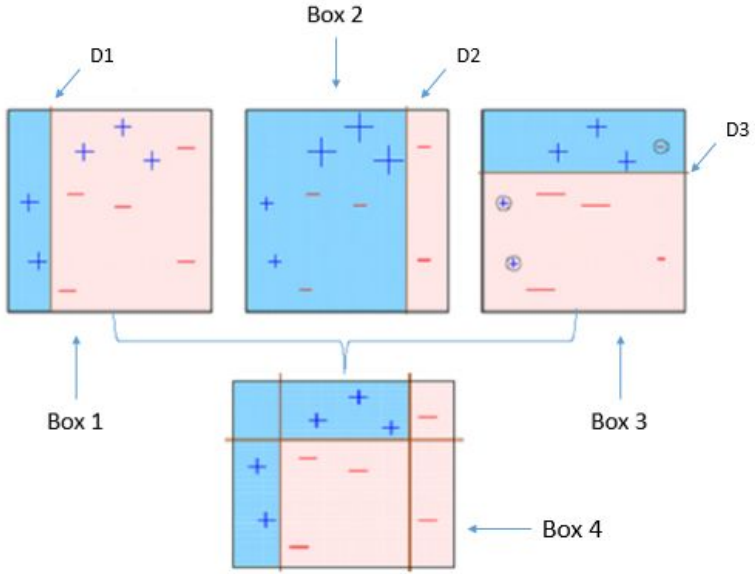
XGBoost: Extreme Gradient Boosting

→ converts a weak learner into a strong one

→ one of the most popular classifiers,based on Gradient descent method.The learning procedure consecutively fit new models to provide a more accurate estimate of the response variable

→ Aims at minimising loss function, in this case, log-loss.

$$logLoss=-1N\sum Ni=1(yi(logpi)+(1-yi)log(1-pi))$$



Results

We got a log-loss of 0.304, when we used only the mean, we got a log loss of 0.55, and when we used a combination of wmd and wordnet features, only we got a log loss of 0.356. With a combination of all the above we were able to minimise it to 0.30

References

https://www.analyticsvidhya.com/blog/2015/11/quick-introduction-boosting-algorithms-machine-learning/

http://ants.iis.sinica.edu.tw/3BkMJ9lTeWXTSrrvNoKNFDxRm3zFwRR/55/Sentence%20Similarity%20Based%20on%20Semantic%20Nets%20and%20corpus%20statistics.pdf