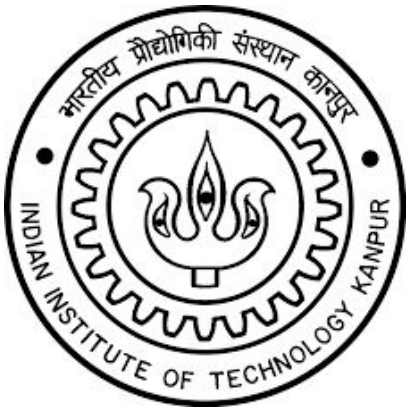
Quora Question Duplication

Natural Language Processing (CS671A)

Instructor: Harish Karnick



Group 17

Shivam Yadav - 14655
Abhishek Maher - 14021
Sachin K Salim - 14575
Raushan Joshi - 14537
Jatin Gupta - 14280
Rishav Raj - 150587

1. Problem Statement

Given a pair of questions, our aim is to detect whether they are duplicate or have exactly same semantics. This is useful in question-and-answer applications like Quora since this lets it deduplicate its database by merging the threads in one of the duplicate question with the other. Our model *M* models a function *f* that takes two input questions q_1 and q_2. It then returns *1* if the questions are semantically same and *0* otherwise.

2. Expected Challenges

Two questions could have most of the words common but still have different meaning. For instance,

1. What are natural numbers?
2. What is the smallest natural number?

These both questions have different meanings and generate different answers but they've a very similar sentence structure and use same words. This could result in false positives.

Conversely, two questions could use different words and have dissimilar structure but still mean the same.

1. Is watching too much television bad?
2. Should I not spend excess time in front of a TV?

Our model should tackle these challenges by learning the patterns in the questions asked.

3. Dataset

Quora has officially released a public dataset of Question Pairs which consists of over 400K lines of potential question duplicate pairs. Each row of the dataset contains a query id, respective question ids, the question pair and a binary value to indicate if the question pair is duplicate. It consists of about 250K negative (Non-duplicate) and about 150K positive (duplicate) instances. On analyzing the dataset, it is found that it contain shortest question of 1 character long and longest question of about 1100 characters long. Thus we remove such extreme data points.

Dataset link - http://qim.ec.quoracdn.net/quora_duplicate_questions.tsv

We may also use the test and train split dataset readily available on Kaggle.

Link - https://www.kaggle.com/c/quora-question-pairs/data

4. Previous Work

Various methods have been proposed to solve this problem. The Quora Question Duplication paper [3] by Elkhan Dadashov, Sukolsak Sakshuwong and Katherine Yu explored this problem using LSTM networks. It used a Siamese architecture where the representation of both sentences were learnt from an LSTM. Another approach mentioned in the paper was a sequence-to-sequence model that uses two LSTMs with separate parameters where the last state of the first LSTM is passed to the second. To initialise the word embedding, the paper

used the new model for word representation called GloVe[2], for Global Vectors, because in this model the global corpus statistics are captured directly by the model. GloVe pre-trained word vector with 300-dimensional vectors and initialized training words not contained in GloVe to small random vector[*]. According to the results given in the paper[2], GloVe is a new global log-bilinear regression model for the unsupervised learning of word representations that outperforms other models on word analogy, word similarity, and named entity recognition tasks.

Initially, Quora used random forests on extracted features to perform the classification task. However, in a Semantic Question Matching with Deep Learning blog at Engineering at Quora, the author speaks about three different experimental approaches to solve the problem. Two involve LSTM networks to perform classification after either concatenating the embeddings or extracting the Euclidean distance and angle as features from them and the third was an attention based approach that combined neural network attention with token alignment.

An attention-based approach has also been used. It combined neural network attention with token alignment, commonly used in machine translation. The most prominent advantage of this approach, relative to other attention-based approaches, was the small number of parameters. This model represents each token from the question with a word embedding. Training was done on example sentences using different scoring functions: additive scoring, bilinear scoring and bilinear scoring adding the original embeddings.

Performance of above mentioned models

Model	Precision	Recall	F1	Accuracy
Siamese with LSTM	73.0	86.8	79.3	83.2
Seq2Seq LSTM with Attention	70.2	83.7	76.4	80.8
LSTM with concatenation	88	86	87	87
LSTM with distance and angle	83	94	88	87
Decomposable attention	81	95	87	86

5. Proposed method

5.1. Embeddings

We plan to use Word2Vec for embedding the words. Word2Vec is a general term used for similar algorithms that embed words into a vector space with 300 dimensions in general. These vectors capture semantics and even analogies between different words. We are also considering the use of GLoVe pre-trained word vectors to initialize our word embeddings. GLoVe learns by constructing a co-occurrence matrix (words X context) that basically counts how frequently a word appears in a context. We would also try to construct our own word embeddings using Quora's text corpus to achieve better results.

The next step is to combine these word vectors to form a representation for the whole sentence. A simple method would be to take the arithmetic mean of the vector representation of

the contained words. We apply weighted average of word vectors by using TF-IDF scores to enhance mean vector representation.

5.2 Siamese LSTM Approach

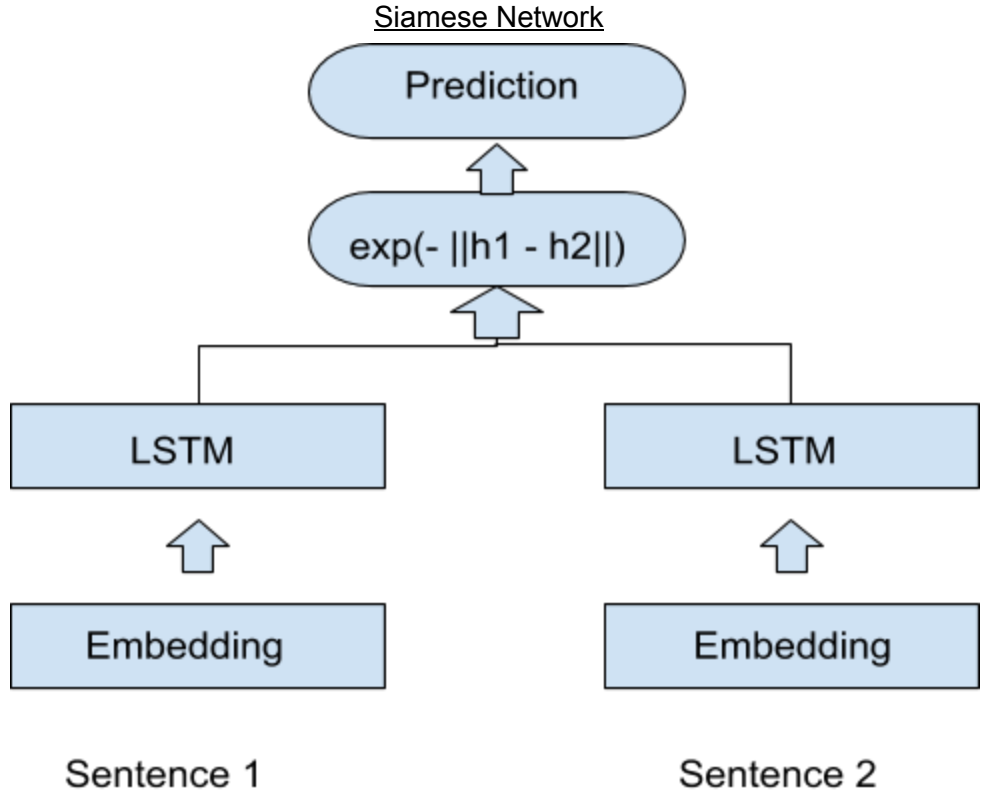
Siamese network[1] is a method to calculate semantic relatedness score between two sentences. It takes two sentences as inputs and projects data into a space in which similar items are contracted and dissimilar ones are dispersed over a learning space.

A single LSTM is used to produce output vectors for each question in the pair at the sentence level from the word-level embeddings for the first *L* words in each sentence where *L* is the maximum sentence length. This two vectors are then taken and their Hadamard product computed to form a single feature vector. This is then fed to a dense layer to produce the final classification.

The LSTM cells are implemented with standard equations for the gates where x(t) are the word embeddings at the t-th word:

$$
\begin{aligned}
i_t &= \sigma(W^{(i)}x_t + U^{(i)}h_{t-1}) \\
f_t &= \sigma(W^{(f)}x_t + U^{(f)}h_{t-1}) \\
o_t &= \sigma(W^{(o)}x_t + U^{(o)}h_{t-1}) \\
\tilde{c}_t &= \tanh(W^{(c)}x_t + U^{(c)}h_{t-1}) \\
c_t &= f_t \circ c_{t-1} + i_t \circ \tilde{c}_t \\
h_t &= o_t \circ \tanh(c_t).
\end{aligned}
$$

Here $\sigma(x)$ is the sigmoid function, h_0 is initialized to be zero, and the parameters $W^{(i)}, W^{(f)}, W^{(o)}, W^{(c)} \in \mathbb{R}^{H \times k}$ and $U^{(i)}, U^{(f)}, U^{(o)}, U^{(c)} \in \mathbb{R}^{H \times H}$, where k is the embedding size.



Losses to be considered will be both cross-entropy loss and margin-based loss.

References

[1] Jonas Mueller and Aditya Thyagarajan. Siamese Recurrent Architectures for Learning Sentence Similarity. In AAAI, 2016.

[2] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. "GloVe: Global Vectors for Word Representation," in Proceedings of the 2014 Conference on Empirical Methods In Natural Language Processing (EMNLP 2014), October 2014.

[3] E Dadashov, S Sakshuwong, K Yu. Quora Question Duplication

[4] Shuohang Wang and Jing Jiang. “Learning Natural Language Inference with LSTM” in Proceedings of NAACL, 2016.

[5] Eren Golge. Duplicate Question Detection with Deep Learning on Quora Dataset