# Identifying Duplicate Questions using Siamese LSTM Architecture

**Junwoong Yoon**
Department of Chemical Engineering
Carnegie Mellon University
junwoony@andrew.cmu.edu

**Ni Zhan**
Department of Chemical Engineering
Carnegie Mellon University
nzhan@andrew.cmu.edu

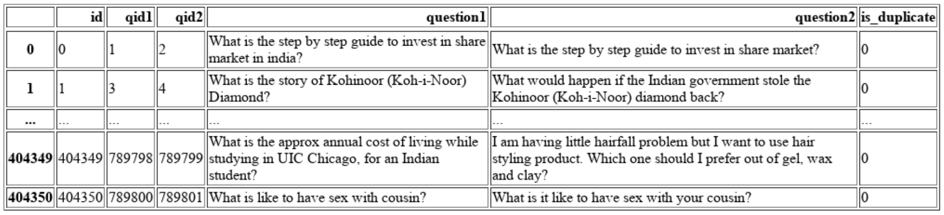
## Abstract

Identifying duplicate questions is an important problem for efficient knowledge sharing on sites such as Quora, and detecting semantic similarity is a research interest in natural language processing. This work uses word embeddings, siamese recurrent neural network structure, and distance metric to detect similarity between questions. The methods achieve a 0.830 validation accuracy on the Quora Question Pairs dataset. We test new methods, changing the preprocessing, word embedding, distance metric, and neural network structure. We find that effective preprocessing, word embeddings, and distance metrics significantly improve the model, while changing the neural network structure result in similar performance.

## 1 Introduction

The project is “Identifying Duplicate Questions”. On question and answer websites such as Quora, Stack Exchange, user-submitted questions with different wording could have the same meaning. These sites want to have only one instance of a logically distinct question for efficient knowledge sharing, which requires detection of questions with duplicate meaning. Paraphrase detection is a challenge in natural language processing because word meaning and sentence structure are difficult to capture and compare as data. Therefore, the goal of this project is to identify if two questions are semantically equivalent, that is if they are duplicated.

Questions having similar vocabulary could have different meaning based on sentence structure. Deep learning has been successful in addressing this challenge and capturing sentence semantics. We build on this work and experimented with different model structures using Quora’s "Question Pairs" dataset, shown in Fig. 1. Briefly, we used word2vec to represent the data, and recurrent neural networks to encode semantic meaning of questions. We used distance metrics between the vector representation of questions to determine if they are duplicates.



| | id | qid1 | qid2 | question1 | question2 | is_duplicate |
|---|---|---|---|---|---|---|
| **0** | 0 | 1 | 2 | What is the step by step guide to invest in share market in india? | What is the step by step guide to invest in share market? | 0 |
| **1** | 1 | 3 | 4 | What is the story of Kohinoor (Koh-i-Noor) Diamond? | What would happen if the Indian government stole the Kohinoor (Koh-i-Noor) diamond back? | 0 |
| **...** | ... | ... | ... | ... | ... | ... |
| **404349** | 404349 | 789798 | 789799 | What is the approx annual cost of living while studying in UIC Chicago, for an Indian student? | I am having little hairfall problem but I want to use hair styling product. Which one should I prefer out of gel, wax and clay? | 0 |
| **404350** | 404350 | 789800 | 789801 | What is like to have sex with cousin? | What is it like to have sex with your cousin? | 0 |

Figure 1: Quora "Question Pairs" dataset

Preprint. Work in progress.

## 2 Background and Related Works

Paraphrase detection is an important and common problem in natural language processing. Traditional machine learning techniques is one approach, as demonstrated by Dey et al. [1]. They used support vector machines (SVMs) with hand-picked features and extensive preprocessing (adjusting for named entities, synonyms, etc.), and the method performed well on SemEval-2015 dataset.

Various deep learning techniques have also been applied successfully. Bogdanova et al. used a convolutional neural network and word embeddings to represent questions as vectors, and measured distance between vectors [2]. They found their method more effective than Jaccard similarity and SVMs on StackExchange data.

One published work that used the same Quora dataset was from Wang et al. They used Long Short Term Memory Networks (LSTM) and matching time-step of embeddings between sentences [3]. Their model reached 0.88 accuracy on the Quora dataset.

This work builds upon the Siamese LSTM and Manhattan distance model used by Mueller et al. [4]. The "Siamese" architecture refers to encoding two input questions using the same LSTM network, as shown in Fig. 2. Siamese network is capable of performing similarity tasks and has been used for capturing semantic relatedness of sentences, and Mueller et. al. showed the LSTM successfully models complex semantics. This work focuses on optimizing the Siamese neural network model and pipeline.

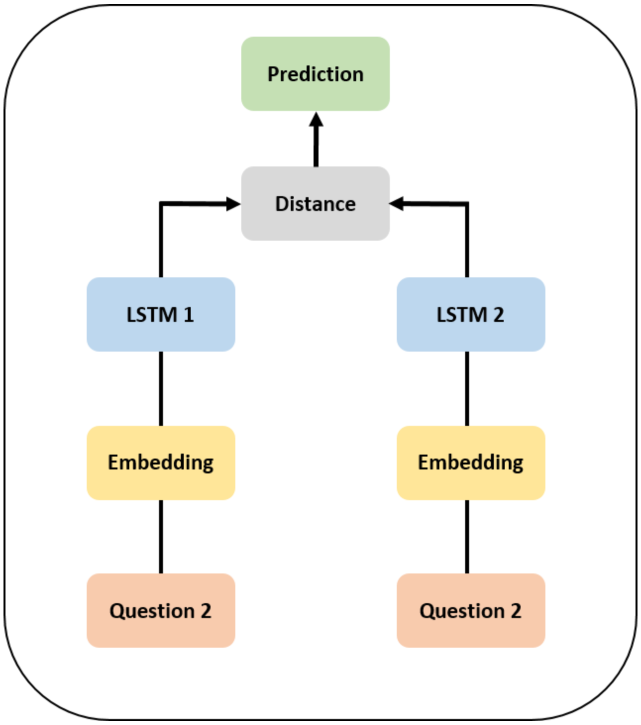


Figure 2: Architecture of our Siamese network. LSTM 1 and LSTM 2 share the same weights.

## 3 Dataset

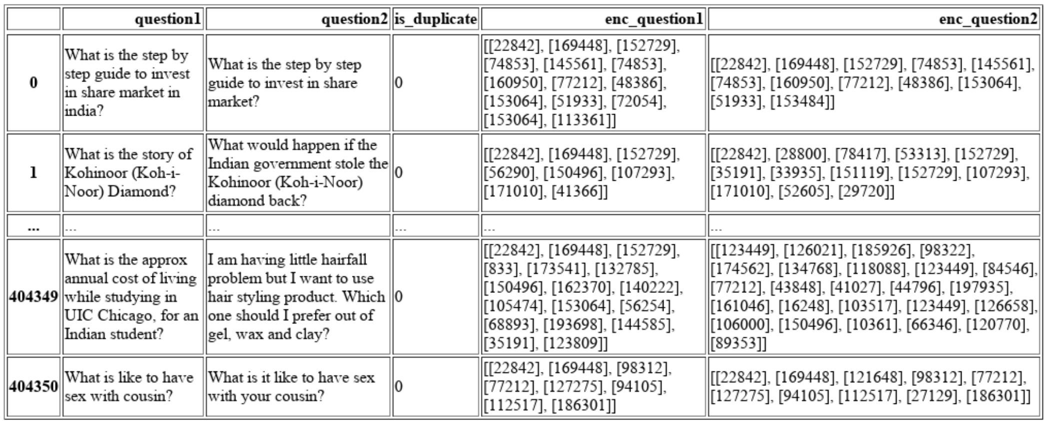
The dataset contains over 404k Quora question pairs. Each data point has two questions and a binary true/false label indicating if the questions are duplicates as shown in Fig. 1. The total number of unique questions in the training dataset are 790k and the ratio of duplicate question pairs is about 40%. The dataset had 84,000 unique words in its vocabulary.

# 4 Methods

## 4.1 Preprocessing

We converted the raw text data into mathematical representation that our model can read. First, we created a bag of words by storing unique words in the training data. We then made ‘word2idx’ dictionary where the keys are words (str) and values are the corresponding indices (int). Using the ‘word2idx’, we encoded all the questions with list of word indices as shown in Fig. 3. This simple preprocessing is implemented in our baseline model.

We also performed dataset "clean-up" because previous work indicated that preprocessing noisy language datasets is an important step. During clean up, we removed Natural Language Toolkit (NLTK) stop words, converted contractions into full words, and removed punctuation. To batch computations, we used fixed length of input questions. To achieve this, we padded the front of shorter sentences with zeroes.



| | question1 | question2 | is_duplicate | enc_question1 | enc_question2 |
|---|---|---|---|---|---|
| **0** | What is the step by step guide to invest in share market in india? | What is the step by step guide to invest in share market? | 0 | [[22842], [169448], [152729], [74853], [145561], [74853], [160950], [77212], [48386], [153064], [51933], [72054], [153064], [113361]] | [[22842], [169448], [152729], [74853], [145561], [74853], [160950], [77212], [48386], [153064], [51933], [153484]] |
| **1** | What is the story of Kohinoor (Koh-i-Noor) Diamond? | What would happen if the Indian government stole the Kohinoor (Koh-i-Noor) diamond back? | 0 | [[22842], [169448], [152729], [56290], [150496], [107293], [171010], [41366]] | [[22842], [28800], [78417], [53313], [152729], [35191], [33935], [151119], [152729], [107293], [171010], [52605], [29720]] |
| **...** | ... | ... | ... | ... | ... |
| **404349** | What is the approx annual cost of living while studying in UIC Chicago, for an Indian student? | I am having little hairfall problem but I want to use hair styling product. Which one should I prefer out of gel, wax and clay? | 0 | [[22842], [169448], [152729], [833], [173541], [132785], [150496], [162370], [140222], [105474], [153064], [56254], [68893], [193698], [144585], [35191], [123809]] | [[123449], [126021], [185926], [98322], [174562], [134768], [118088], [123449], [84546], [77212], [43848], [41027], [44796], [197935], [161046], [16248], [103517], [123449], [126658], [106000], [150496], [10361], [66346], [120770], [89353]] |
| **404350** | What is like to have sex with cousin? | What is it like to have sex with your cousin? | 0 | [[22842], [169448], [98312], [77212], [127275], [94105], [112517], [186301]] | [[22842], [169448], [121648], [98312], [77212], [127275], [94105], [112517], [27129], [186301]] |

Figure 3: Preprocessed data that converts sentences into vectors of indices of words

## 4.2 Embedding

To capture semantic meaning of words, we embedded each word as a vector using word2vec [5-7]. We used Google’s embedding of 300-dimensional vectors for 3 million words that was trained on Google News dataset containing 100 billion words. The literature indicates that word embeddings are an important aspect of natural language processing problems and transfer learning or pretraining with a dataset can improve performance. To test this, we pretrained word vectors using the Quora dataset itself. We used both continuous bag of words (cbow) and skip-gram algorithms to train different word vectors of 300-dimensions, and compared the resulting model perfomance using the three word embeddings (Google News, cbow, and skip-gram). After embedding, each question is represented as a matrix of word vectors.

## 4.3 Question Encoding

We implemented siamese network using a pair of the same LSTM using Keras. The matrices of questions were inputs to the LSTM, and the word vectors were fed sequentially to the LSTM. The LSTM captured relevant semantic information, and the vector representing the question was the final hidden state from the LSTM [4]. We used the LSTM structure of Hochreiter et al. [8], and tested different LSTM output dimension. We also wanted to examine how changing the recurrent neural network structure would affect performance, so we tested the Gated Recurrent Unit (GRU) structure as an alternative to the LSTM, using the GRU structure from Cho et al. [9].

### 4.4 Distance Measure

To compare similarity between the vector representation of questions, we calculated a distance estimate between the two vectors. The negative exponential of the distance then scales to $\hat{y} \in [0, 1]$, which is used to predict if the questions are duplicates. If $\hat{y}$ is above a cutoff, the questions were duplicates, and if $\hat{y}$ was below the cutoff, the questions were not duplicates.

$$\hat{y} = exp(-d(\mathbf{h_1}, \mathbf{h_2})) \tag{1}$$

The distance measure could make a difference because the vectors are part of a space encoded by the neural networks. Because of this, we tested different distance estimates including euclidean ($L_2$), cosine, and Manhattan distance.

$$d_{euc}(\mathbf{h_1}, \mathbf{h_2}) = \|\mathbf{h_1} - \mathbf{h_2}\| \tag{2}$$

$$d_{cos}(\mathbf{h_1}, \mathbf{h_2}) = \frac{\mathbf{h_1} \cdot \mathbf{h_2}}{\|\mathbf{h_1}\|\|\mathbf{h_2}\|} \tag{3}$$

$$d_{manhattan}(\mathbf{h_1}, \mathbf{h_2}) = \mid \mathbf{h_1} - \mathbf{h_2} \mid . \tag{4}$$

### 4.5 Training

We use the mean squared error as our loss function for all models,

$$L = \frac{1}{N}\|\hat{\mathbf{y}} - \mathbf{y}\|^2 \tag{5}$$

where $\hat{\mathbf{y}} \in [0, 1]$ and $\mathbf{y} \in \{0, 1\}$ denote the prediction and ground truth labels. We use different optimization methods and backpropagation to adjust network weights and compared the resulting model. With Adadelta, we used gradient clipping to avoid exploding gradient problem. The data was split into training (90%) and test sets (10%), and we used batch training with batch size of 64.

## 5 Experiments and Results

### 5.1 LSTM Output Dimension

We tested different number of hidden units, or output dimension of the LSTM, as shown in Fig. 4. The validation accuracy improved as number of hidden units increased from 10 to 50 units, and then decreased from 50 to 60 units. This indicates that around 50 dimensions can adequately capture the variation in the space of encoded questions, but higher dimension would increase model complexity and probability of overfitting. For the other experiments, we used 50 units.

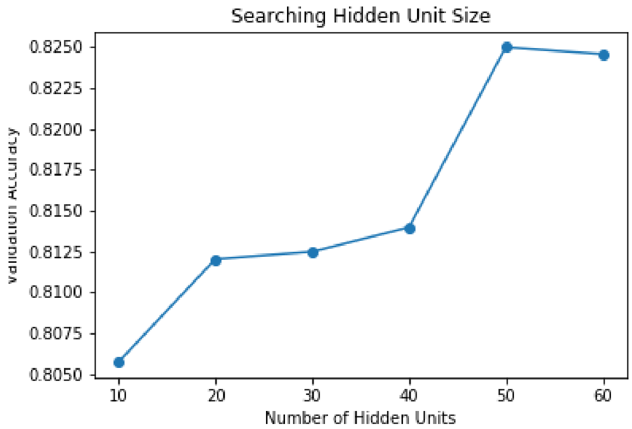


Figure 4: Graph of hidden unit size vs. accuracy for model using LSTM with "clean-up" preprocessing

### 5.2 Optimizer

Table 1 shows how accuracy changes with different optimizers using our baseline model. RMSProp, Adam, and Adadelta performed around the same level, and these three optimizers performed better than Stochastic Gradient Descent (SGD) by about 10% accuracy. RMSProp, Adam, and Adadelta change the parameters in a smoother way than SGD. Because of this, we used Adadelta for the remaining experiments.

Table 1: Comparison of Optimizers and the corresponding Validation Accuracies uinsg baseline model

| **Optimizer with Manhattan distance** | **Acc. (Val)** |
|---|---|
| SGD | 0.697 |
| RMSprop | 0.798 |
| Adam | 0.793 |
| Adadelta | **0.807** |

### 5.3 Distance Measures

According to Table 2, the best distance measure was Manhattan distance, and the worst distance measure was L2 distance. Manhattan distance outperformed L2 distance by almost 20% in accuracy, which makes the chosen distance metric very important for comparing similarity. It is non-obvious what the "natural" distance is in the space encoded by the LSTM. To improve on the Manhattan distance, there could be a neural network to calculate a distance measure between the question vectors. In other models, we continued to use Manhattan distance.

Table 2: Comparison of Distance Metrics and the corresponding Validation Accuracies using baseline model

| **Distance Metrics with Adadelta Optimizer** | **Acc. (Val)** |
|---|---|
| $L_2$ distance | 0.615 |
| Cosine distance | 0.632 |
| Manhattan distance | **0.807** |

### 5.4 Model Types

We show how different preprocessing, word-vector embedding, and models change performance. These are shown in Table 3. We use validation accuracy as the performance metric to compare methods or models. The baseline model is siamese LSTM, without clean-up, and Manhattan distance, and it reaches 0.807 validation accuracy. In all models, we stopped training after the validation accuracy converged, at least 10 epochs. Fig. 5 shows the training curve for the baseline model.

The siamese LSTM model did not perform as well as the LSTM and matching time-step method of Wang et al., which achieved an accuracy of 0.88 [3]. Their matching time-step method was able to capture more semantic meaning from questions as they were encoded. This is expected, as our method was simpler, and used the final hidden state from the LSTMs.

Table 3: Comparision of Models with different preprocessing and embeddings

| **Model** | **Acc. (Val)** |
|---|---|
| LSTM, no clean-up | 0.807 |
| LSTM, clean-up | **0.825** |
| LSTM, clean-up, cbow | 0.811 |
| LSTM, clean-up, skip-gram | **0.830** |
| GRU, clean-up | 0.821 |

### 5.5 Preprocessing and Clean Up

When clean-up is included in preprocessing, the validation accuracy improves by 3%. This is consistent with what Dey et al. found, that clean-up in preprocessing significantly improves model performance. From this result, further clean up would likely continue improvement over the current achieved accuracy.

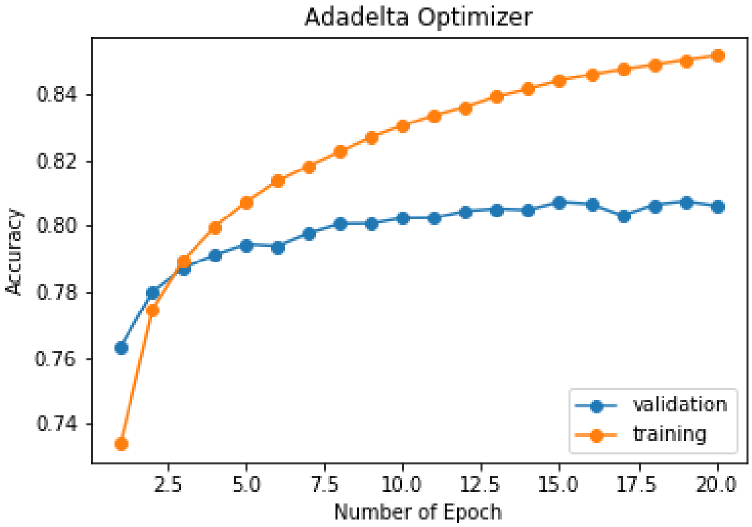


Figure 5: Learning curve showing convergence of the validation accuracy produced by Adadelta optimizer

### 5.6 LSTM Structure

Comparing LSTM and GRU structure, both models perform at the same level. This result is expected based on an empirical study by Jozefowicz et al. [10]. They found that LSTMs and GRUs perform around the same level, but the difference in performance also depends on the task. A non-obvious change in LSTM structure would be necessary to significantly improve the siamese network performance for detecting semantic similarity. We expect that changing the activation function or other minor changes to the LSTM structure would not significantly change model performance.

### 5.7 Word Embeddings

The word-vector embeddings make a significant difference on model performance. The cbow word embeddings on the dataset performed slightly worse than the Google News embeddings, but the skip-gram embeddings performed slightly better than Google News embeddings. This is remarkable considering that the embeddings trained on the dataset were trained on vocabulary of 84,000 words, while the Google News embeddings were trained on 100 billion words. Using word-vector embeddings that were trained with the dataset itself captured word meanings with context of the dataset, and can achieve good model performance. It is possible that these word embeddings would not generalize well to another task or dataset such as comparing sentence similarity from news article.

The result also indicates that skip-gram performs better than cbow algorithm for sparse vocabulary, also noted by Mikolov [5]. The cbow trains and predicts for words given context, while skip-gram trains and predicts for context given a word. This difference increases the training examples for skip-gram given a dataset, and makes skip-gram better at capturing specific meaning while cbow captures general word meaning. While training the word-vector embeddings, there are other hyperparameters such as "window", which changes the number of neighboring words to consider, and dimensionality of word vector which could change model performance. We used window of five and 300 dimension word-vectors. The best window likely depends on the average length of questions in the dataset. There is probably a range of word-vector dimensions that adequately capture word meaning, and model performance would probably not change if staying within this range. The word embeddings are an important part of the model and natural language processing, and there is room for further improvement and experimentation with regard to the word-vector embeddings.

## 6 Conclusion and Future Work

We used word embeddings, siamese LSTM and Manhattan distance to detect duplicate questions using the Quora questions dataset. The process performed well, achieving an accuracy of 0.830, indicating that the LSTMs effectively capture semantic meaning and question structure.

We changed aspects of preprocessing, embedding, and the model and found how they changed performance. We showed that changing the recurrent neural network structure from LSTM to GRU did not significantly change performance. We showed that preprocessing and clean up is an important step, and our clean up methods improve model accuracy by 3%. Word embeddings also make a difference, and further work could investigate other embedding methods. The distance metric is also an important part of the model, and choosing an effective distance can improve accuracy by 20%. In future work, a neural network could be used to optimize the distance metric.

For future work, augmenting the dataset with more examples, by generating non-equivalent sentences, could also improve model performance. In addition, we could add regularization term into our loss function in case the model overfits the training data.

## References

[1] Kuntal Dey, Ritvik Shrivastava, and Saroj Kaushik. A paraphrase and semantic similarity detection system for user generated short-text content on microblogs. In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 2880–2890, 2016.

[2] Dasha Bogdanova, Cicero dos Santos, Luciano Barbosa, and Bianca Zadrozny. Detecting semantically equivalent questions in online user forums. Proceedings of the Nineteenth Conference on Computational Natural Language Learning, 2015.

[3] Zhiguo Wang, Wael Hamza, and Radu Florian. Bilateral multiperspective matching for natural language sentences. CoRR, 2017.

[4] Jonas Mueller and Aditya Thyagarajan. Siamese recurrent architectures for learning sentence similarity. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, AAAI'16, pages 2786–2792. AAAI Press, 2016.

[5] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. CoRR, 2013.

[6] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems, pages 3111–3119, 2013.

[7] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 746–751, 2013.

[8] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. Neural computation, 9(8):1735–1780, 1997.

[9] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. CoRR, 2014.

[10] Rafal Jozefowicz, Wojciech Zaremba, and Ilya Sutskever. An empirical exploration of recurrent network architectures. In International Conference on Machine Learning, pages 2342–2350, 2015.