# Quora Question Pairs

**Haresh Chudgar, Neha Yadav, Arunima Chaudhary**
College of Information and Computer Science
University of Massachusetts, Amherst
{hchudgar,nyadav,arunimachaud}@cs.umass.edu

## Abstract

Predicting whether two sentences are a paraphrase of one another is an active research problem with many real life applications. One such application is minimizing duplicate questions and answers on question answering websites such as Quora or Yahoo! Answers. We used the Quora Question pairs dataset (Iyer et al., 2017) which consists of question pairs tagged as duplicate and not duplicate. Among the methods tried, the best accuracy achieved was 82%.

## 1 Introduction

The task of paraphrase detection involves given two sentences, detecting whether one statement has the same meaning as another. An important application of this is in question answering websites such as Quora where duplicate questions leads to a bad user experience and fragmented answers. We used three methods to predict if a question is duplicate of the other or not. The first method determines if the pair is duplicate by comparing the nouns and verbs, the second method sums up the word embeddings of individual words of the question as a question embedding and uses a dense network for the prediction, and the third uses a recurrent nets output as a question embedding instead. Surprisingly the simple method of summing up the individual embeddings resulted in a better accuracy. In later sections we present in detail each method and analyze their results. Briefly, section 2 presents the current research which can be applied to this problem, section 3 analyses the Quora dataset, section 4, 5 and 6 present each of the methods we used, section 7 discusses the experiments and results we obtained from each method and lastly section 8 concludes the paper with ideas we have in mind but did not find time to test them.

## 2 Related work

In this section we will describe the relevant work that has been done in problems similar to our problem statement. In (Wang et al., 2016), the word representation is depicted by Mapping each word in both sentences to a vector using Word2Vec (Mikolov et al., 2013). The semantic decomposition is performed by finding the closest matching phrase in sentence 2 for each word in sentence 1. A phrase in sentence 2 might be only a single word to all words. For example the word relevant in S1 will match to ”not relevant” in S2. After finding matching phrases, the algorithm de- composes the phrase into similar part and dissimilar part. For example if salmon is matched to the word sockeye (which is red salmon), then the distance between the salmon and sockeye is decomposed into two scalars.After decomposition we are left with two vectors, the similarity vector and the dissimilarity vector (Qiu et al., 2006). In the composition step the algorithm applies a convolutional layer (Kim et al., 2014) followed by max pooling layer to the two vectors and generates a feature vector. The above three steps are applied both ways, from S1 to S2 and vice versa, therefore we get two features vectors from the composition step. The final step is to apply a similarity function which outputs a score between 0 and 1.

In Bilateral Multi-Perspective Matching for Natural Language Sentences (Wand et al., 2017). The word representation consists of representing each word in both sentences with a vector of word embeddings Glove (Pennington et al., 2014) or word2vec (Mikolov et al., 2013) and character-composed embeddings using LSTM (Hochreiter and Schmidhuber et al., 1997) Context Representation Layer is represented by encoding contextual embeddings for each time step of both the sentences using bi-directional LSTM (BiLSTM). In matching layer,they matched each time-step of one sentence against all time-steps of the other sentence using a multi-perspective matching operation. Then they repeated the process for the other sentence to obtain two matching vectors. As part of aggregration layer,they applied another BiLSTM model to each of the sequence of matching vectors individually. Then concatenated vectors gen- erated from the last time-steps of the BiLSTM models to generate a fix-length matching vector.Finally in prediction layer, they evaluated the probability of the two sentences being paraphrases of each other by using a two layer feed-forward neural network that consumes the fix-length matching vector.
(Bowman et al., 2015) in their paper introduce a data set for entailment and also suggest methods to solve the task which we have borrowed, reproduced and tweaked.

## 3 Dataset

Quora recently released a labelled data set consisting of question pairs with the task of finding pairs which have duplicate questions. The dataset has 40,000 question pairs. A statistical overview of the data:

- Average words share between question pairs: 0.22
- Average number of words in a question: 11
- Average number of characters in a question: 60
- Maximum number of characters in a question: 1169

Figure 1 shows the number of duplicate and non-duplicate pairs present in the dataset. 0 denote the pair is not duplicate of each other while 1 denote the pair is duplicate of each other.

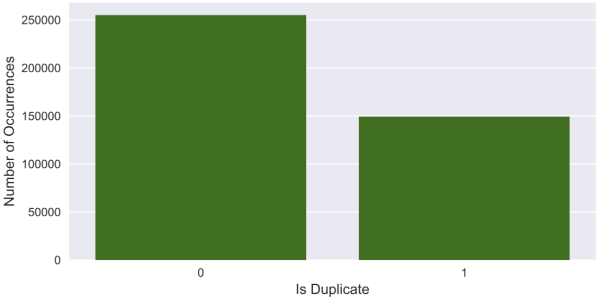

Figure 1: Distribution of duplicate and nonduplicate pairs

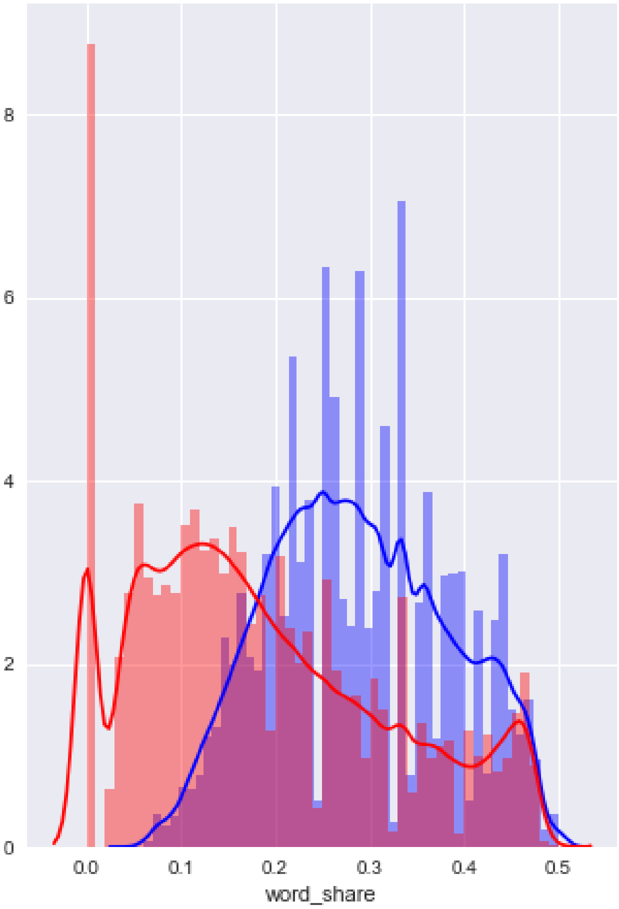

Figure 2: Word share between question pairs

We define word share between two sentences as the fraction of words which are present in both the sentences or number of words common to both the sentences. Figure 2 represents a distribution of word share where pink color represents non duplicates and blue represents duplicate pairs. There is a higher score of word share for duplicates but considerable overlap between the two.

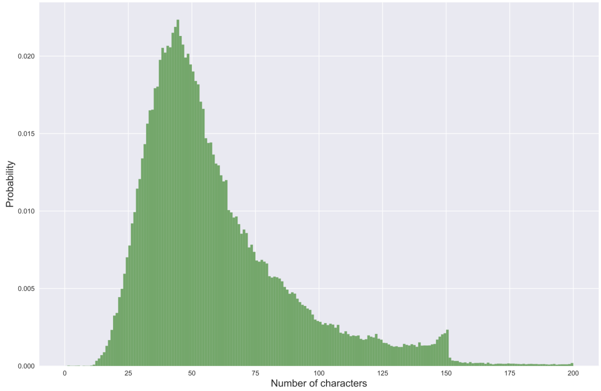

Figure 3: Normalised histogram of character count in questions

To understand if there is any size limit on Quora questions and how many characters are presents on an average in the questions. Wr have plotted the normalised histograms of character count in questions. Figure 3 shows that on an average most questions have characters anywhere from 15 to 150 characters in them. Another interesting thing to note is the steep cut-off at 150 characters, for most questions, which supported our claim that there could be size limit on Quora questions.

Another important thing worthy of attention is that the histogram is truncated at 200 character, and that the max of the distribution is at just under 1200 characters, although samples with 200 characters are very rare

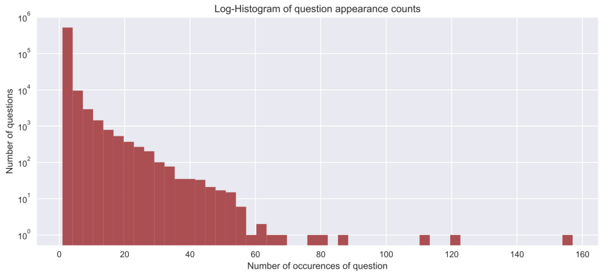

Figure 4: Log-Histogram of question appearance counts

To understand if there exists multiple entries of the same question in the dataset, we plotted the log histogram of question appearance counts. Figure 1 shows the majority of the questions only appear a few time, with very few questions appearing several times (and a few questions appearing many times). It can be observed in the graph that one question has appeared more than 160 but this is definitely an outlier.

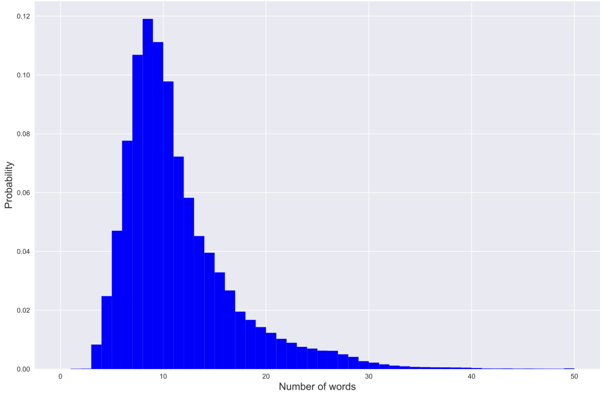

Figure 5: Normalised histogram of word count in questions

To understand in depth the structure of sentences, question pairs, we plotted the normalised histogram of word count in questions. This was necessary to understand if there is larger difference in the word count of two questions would it affect it classification as either duplicates or non-duplicates. Figure 5 shows similar distribution as Figure 3 for word count, with most questions being about 10-11 words long.

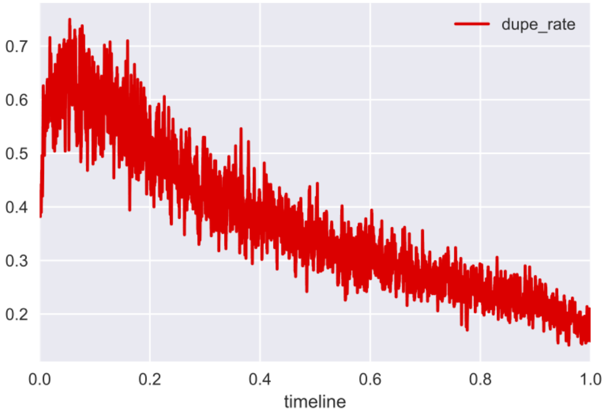

Figure 6: Analysis of temporal pattern in the data

We were also interested to analyse if there exists a temporal pattern to the data, and qid is used as proxy for it. Higher the qid is more recent is the question in the dataset. To understand this we plotted the mean response rate on a sliding window. To achieve this task whole data is sorted by increasing qid. Figure 6 shows the pattern where it is clearly visible that there does exists temporal behavior to qid. In other words qids indeed seem to be in time order. We could use the above fact to more appropriately construct validation data splits and training data samples.

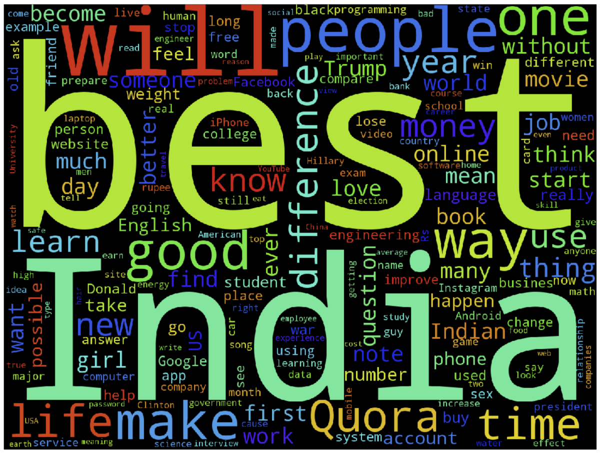

Figure 7: Word Cloud of the Question pairs in the dataset

## 4 WordMatch

WordMatch matches nouns and verbs of the questions in each pair. The intuition is that nouns represent topics and verbs represent actions on the topics. The words in each question are first tokenized and then individual words are POS tagged.
After creating a list of nouns and verbs for each question, the algorithm checks for at least one noun that is common to both the questions. If the nouns match then it can be said that both the questions are about the same subject or topic. If none of the nouns match, the pair could be classified as not duplicate. Upon finding a noun match, a verb match will be done. To match verbs, not only common verbs are checked but also if any of the verbs synonyms of one question have a match in the other questions verbs. The synonyms are found using WordNet (Christiane Fellbaum, 1998). The question pair is classified as duplicate if a verb match is found.

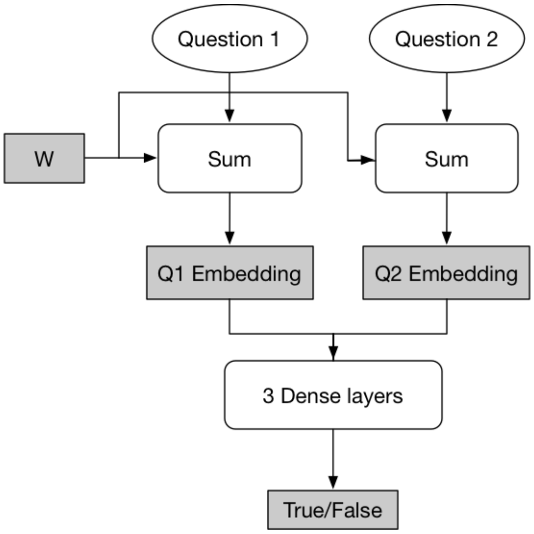

Figure 8: Summed Embeddings model

## 5 Summed Embeddings

In this method, a vector representation of the question is formed by summing the embeddings of individual words in the question. The vectors of the two questions are then concatenated and passed through three dense layers and finally a softmax to predict whether the pair is a duplicate or not. The architecture is shown in figure 8. **W** is the word embedding matrix. By using word embeddings, words which are synonyms would be closer to each other than words which are antonyms of each other and words that are not related. Hence the model would be able to identify a duplicate pair even if the other question is worded differently. In theory, we suspect this model achieves what was intended in WordMatch.

## 6 Question embedding using recurrent nets

By summing up the embeddings as in the previous method, information about the structure of the sentence is lost. So the same question asked a different way with different words might not be picked up by the previous model. An LSTM (Hochreiter and Schmidhuber et al., 1997) is able to capture the structure and capture important words. For example in a sentence like ”Is there an alternative version of Quora?”, the words ’an’, ’of’ do not have meaning individually, the forget gate of the LSTM, a variant of recurrent net, might learn to filter out these words

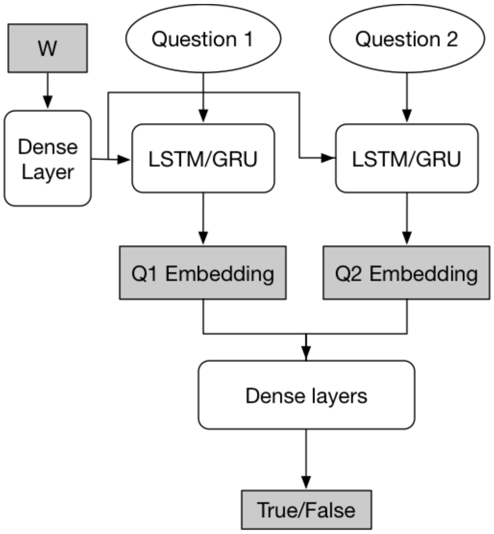

Figure 9: Recurrent net model

in the update. The downside to using recurrent nets is it is much slower to learn compared to the previous model and needs more examples. The architecture is similar to the previous model and is shown in figure 9. The embedding from the recurrent net can be formed in different ways:

- Last output: Take the output of the last neuron as the sentence embedding
- Pooling of outputs: Do a mean/max pooling of the outputs of all neuron

In addition, different recurrent nets can be tried, popular ones being traditional RNN (Cho et al., 2014), Long Short Term Memory(LSTM) (Hochreiter and Schmidhuber et al., 1997), Gated Recurrent Unit(GRU) (Chung et al., 2014)

## 7 Experiments and Results

For experiments, we created a 70:15:15 split of the data, with 70% of pairs in the training set, 15% each in validation and training set. The results are summarized in table **??** Each sub section discusses the results of the three methods.

### 7.1 WordMatch

Due to some technical issues, the experiment was only run on first 600 pairs of questions. For POS tagging, different libraries were tried including:

- TexBlob (Loria et al., 2015)
- nltk.pos_tag (Loper and Bird et al., 2002), (Loper et al., 2004)
- StanfordPOSTagger (**?**)

A maximum accuracy of $66\%$ was achieved. Amongst the selected pairs, 186 were actual duplicates. Our experiment detected 123 true positives and 152 false positives.
The approach of declaring a pair duplicate if at least one of the nouns and verbs match, increases true positives but also ends up increasing false positives. We tried to mitigate this issue by matching at least $50\%$ of verbs in a pair before classifying it as duplicate but this ends up restricting detection of true positives heavily.

The false detection stems from the fact that many pairs have questions whose nouns and verbs match even when they are not duplicates. For example, consider the following pairs:
**Example 1:**
Question 1: What is the step by step guide to invest in share market in india?

- *Nouns:*:{’step’, ’share’, ’guide’, ’market’}
- *Verbs:*:{’invest’, ’is’}

Question 2: What is the step by step guide to invest in share market?

- *Nouns:* {’step’, ’share’, ’guide’, ’market’}
- *Verbs :* {’invest’, ’is’}

The above pair is not a duplicate even though the identified nouns and verbs have a $100\%$ match. Such questions are very difficult to be detected as false positives.

**Example 2:**
Question 1: Which is the best digital marketing institution in Bangalore?

- *Nouns:* {’marketing’, ’Bangalore’, ’institution’}
- *Verbs:* {is}

Question 2: Which is the best digital institute in Pune?

| No | Question 1 | Question 2 | Dup? | Sum | LSTM |
|---|---|---|---|---|---|
| 1 | who is better, clinton or trump? | why is hillary clinton a better choice than donald trump? | Yes | Yes | Yes |
| 2 | is world war 3 likely? | are we getting closer to world war 3? | Yes | Yes | Yes |
| 3 | is twed.com legit? | is zooqle.com legit? | No | No | No |
| 4 | how do i replace the battery in a movado watch? | where can i replace the battery in my movado watch? | No | No | No |
| 5 | does swimming burn more calories than running? if so, why? | does walking burn more calories than running? | No | Yes | No |
| 6 | is the philippines becoming a superpower? | how can philippines become a superpower? | No | Yes | No |
| 7 | how long will quora survive? | how long do you think quora will last? | Yes | No | Yes |
| 8 | how do i stop lying? | what are good ways to stop lying? | Yes | No | Yes |
| 9 | is singapore bigger than india? | is singapore bigger than china? | No | No | Yes |
| 10 | what was winston churchill like? | was winston churchill bad? | No | No | Yes |
| 11 | what is the cfp? | what is cfp? | Yes | Yes | No |
| 12 | do girls like tall guys? | do girls prefer tall guys? | Yes | Yes | No |

Table 1: Sample test cases

- *Nouns:* {’marketing’, ’Pune’, ’institute’}
- *Verbs:* {is}

The above pair is again not a duplicate but there is a significant match in the nouns and verbs. On the other hand, a few cases do not get detected because of limited number of synonyms thats can be found using WordNet (Christiane Fellbaum, 1998). For example, consider the following pair

**Example 3:**
Question 1: How do I read and find my YouTube comments?

- *Nouns:* {’youtube’, ’comments’} item *Verbs:*: { ’read’, ’do’, ’find’}

Question 2: How can I see all my Youtube comments?

- *Nouns:*{’youtube’, ’comments’}
- *Verbs:* {’see’}

Synonyms of ’read’: {’read’} Synonyms of ’do’: {’bash’, ’do’, ’brawl’} Synonyms of ’find’: {’breakthrough’, ’find’} Synonyms of ’see’: {’see’}

The above pair is a clear duplicate in intent reflected in the dataset label but WordNets (Christiane Fellbaum, 1998) limitations prevent from declaring it as a duplicate due to lack of a verb match.

### 7.2 Summed Embeddings

We tested two versions of this model:

- Train embeddings from scratch
- Use pre-trained embeddings from Facebook Fasttext (Loper et al., 2016) , (Mikolov et al., 2013)

The model with pre-trained embeddings gave a much better accuracy compared to training from scratch which was expected.

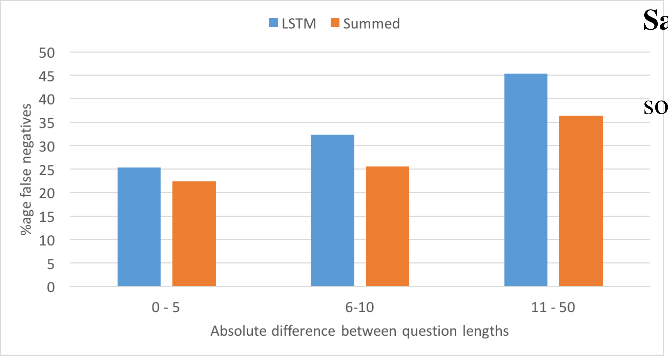

Figure 10: %age false negatives w.r.t difference in question length

### Sample test cases

A sample of some test cases are presented in table 1 to understand what is going on under the hood.

- True Positives: Test case 1 and 2 show the model is able to find duplicates even if the two questions have different words.

- True Negatives: The model is able to identify non duplicates even if there is one or two word difference such as test case 3 and 4.

- False Positives: There were some test cases which the model wrongly predicted as duplicate such as test case 5 and 6. In test case 5, the important words are quite close to each other i.e. all of them belong to sports so the sentence representation might not have been far enough for the model to predict correctly.

- False Negatives: From test case 7 it can be seen that the model some times is not able to account for unnecessary words such as 'do you think'. Figure 10 shows that as difference of length increases, the percentage of false positives increases. This is true even for recurrent models.

## 7.3 Sentence embedding using recurrent nets

We tried different versions of this model, changed the cell type of the recurrent net, varied the dense layers, trained word embeddings from scratch or used pre trained embeddings, dropout. All the results are in figure **??**.

### Sample test cases

Similar to the previous method, we will analyze some of the cases sampled from the test set.

- True Positives: Test case 1 and 2 show the model is able to find duplicates even if the two questions have different words and is also able to handle unnecessary words like test case 7 and 8.

- True Negatives: The model is able to identify non duplicates even if there is one or two word difference such as test case 3 and 4. Also it was able to identify test case 5 was not a duplicate. The dimensions of the pre trained word embeddings are reduced before passing it to the recurrent net, and we suspect that maybe the reason the particular words were now far enough to be classified correctly. The reduction loses information which is not required for the prediction process.

- False Positives and False negatives: We observed many false positives for this model had questions which were very short like test case 9 to 12. We believe this is because we are padding the questions with zero inputs and then taking mean of all outputs. Figure 11 reinforces this assumption. It can be seem that for question pairs with at least one question having length less than 10 the percentage false predictions for LSTM model is much higher than that of the summed embeddings model and this percentage decreases as lengths of questions increase.

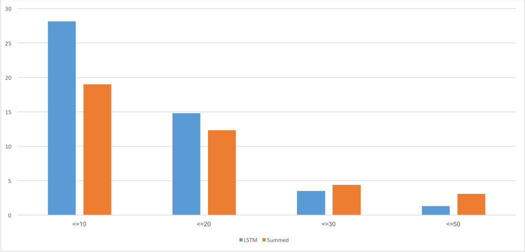

Figure 11: Comparing %age false predictions of methods for different question lengths

| No. | Method details |
|---|---|
| 1 | BasicLSTM, Dimension of embedding reduced to 100, Mean Pooling, Dropout of 0.7 , 2 dense layers |
| 2 | BasicLSTM, Dimension of embedding reduced to 100, Mean Pooling, Dropout of 0.9 , 2 dense layers |
| 3 | LSTM, Dimension of embedding reduced to 50, Mean Pooling, Dropout of 0.9 , 2 dense layers |
| 4 | BasicLSTM, Dimension of embedding reduced to 100, Mean Pooling, Dropout of 0.7 , 4 dense layers |
| 5 | BasicLSTM, Dimension of embedding reduced to 50, Mean Pooling, Dropout of 0.7 , 4 dense layers |
| 5 | Summed Embeddings, 4 dense layers |

Table 2: Method details

### 7.4 Overall comparison

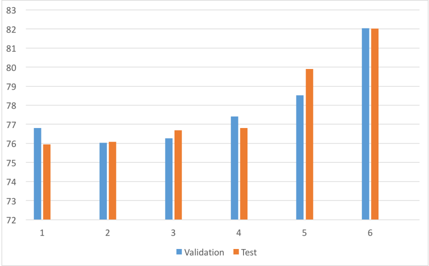

Figure 12: Comparing accuracies of different models

Figure 12 lists the accuracies obtained for different methods. The methods are listed in table 2. The baseline accuracy for the dataset is 50% got by randomly predicting duplicate or not for each pair. The simple model of summing the embeddings of individual words using pre-trained word embeddings acheives the maximum test accuracy of 82%. The recurrent net model is not far behind with 79.9% accuracy for the best variant.

## 8 Conclusion

In this paper, we explored the question pairs dataset (Iyer et al., 2017) released by Quora and methods to predict if the pair is duplicate or not. We also analyzed a sample of the test set to understand where the model is failing. From the observations the model can be tweaked to get better accuracy:

### Summed Embeddings

- Reduce the dimension of the embedding before summing them up might help in losing some information which is not helpful for the prediction process and decrease the %age false negatives.

### Recurrent model

- Variable output size: The LSTM and GRU network had 50 cells with max question length being 50. In cases where the question was of length less than 50, it was padded with zeros at the end. This probably contributed in the increased false prediction rate for question pairs with higher difference in lengths. We could instead do the mean pooling from only the first n cells, n being the question length to generate the question embedding.

- Use a bidirectional network

### Other models

- In both the models discussed, we could include hand engineered features along with the concatenated question embeddings to be fed to the dense layers.

- Using a CNN instead of a recurrent net might be helpful

- Attention networks are also relevant to this problem

## References

Shankar Iyer, Nikhil Dandekar, and Kornl Csernai 2017. *First Quora Dataset Release - Question Pairs*

Zhiguo Wang, Haitao Mi, Abraham Ittycheriah 2016. *Sentence Similarity Learning by Lexical Decomposition and Composition*

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*

Samuel R. Bowman, Gabor Angeli, Christopher Potts, Christopher D. Manning. 2015. *A large annotated corpus for learning natural language inference*, volume 1. Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP).

Zhiguo Wang, Wael Hamza, Radu Florian. 2017. *Bilateral Multi-Perspective Matching for Natural Language Sentences*

Kristina Toutanova, Dan Klein, Christopher D. Manning. 2003. *Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network*, Proceedings of HLT-NAACL

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hrve Jgou, Tomas Mikolov. 2016. *FastText. zip: Compressing text classification models*

Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean. 2013. *Efficient Estimation of Word Representations in Vector Space*

Long Qiu, Min-Yen Kan, and Tat-Seng Chua. 2006. *Paraphrase recognition via dissimilarity significance classification.*

Yoon Kim 2014. *Convolutional neural networks for sentence classification*

Jeffrey Pennington, Richard Socher, and Christopher D Manning 2014. *Glove: Global vectors for word representation*

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. *Distributed representations of words and phrases and their compositionality*

Sepp Hochreiter and Jurgen Schmidhuber 1997. *Long short-term memory. Neural computation*

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, Yoshua Bengio 2014 *Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling*

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, Yoshua Bengio 2014 *Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation*

Steven Loria 2015 *TextBlob: Simplified Text Processing*

Edward Loper and Steven Bird 2002 *NLTK: The Natural Language Toolkit*

Edward Loper 2004 *NLTK: Building a pedagogical toolkit in Python*

Armand Joulin, Edouard Grave, Piotr Bojanowski, Tomas Mikolov 2016 *Bag of Tricks for Efficient Text Classification*