Semantic Question Matching - Is that a Duplicate Quora Question?

Prakruthi Prabhakar
Computer Science Department
Carnegie Mellon University
prakrutp@andrew.cmu.edu

Megha Arora
Computer Science Department
Carnegie Mellon University
marora@andrew.cmu.edu

Abstract

Semantic Question Matching measures the degree of equivalence between paired textual questions. In this paper, we use different variants of a Siamese LSTM neural network to model the questions and jointly learn the similarity between them. We show an increase in performance by using textual attention over individual questions on a dataset released by Quora[1].

1. Introduction

Detecting semantic similarity of sentences has various applications in the field of natural language understanding. Promising approaches to solve the problem can help with many tasks like information retrieval, document clustering, topic detection, question-answering, paraphrase identification, and machine translation. It is also an important problem for maintaining the quality of knowledge bases such as Quora and StackOverflow. It would enable knowledge seekers to access all answers to a question at one place, and for writers to reach a larger readership. Detecting duplicate questions automatically and scalably calls for a deep learning solution to the problem.

In this paper, we present our approaches to identify semantic equivalence between a pair of questions. We define semantic equivalence as - "two questions are semantically equivalent if they can be answered by the exact same answers" [1].

Deep learning researchers have achieved notable results in finding semantic similarity between a pair of sentences. Our work builds upon these existing set of models. In this paper, we use a Siamese network as our base architecture for all the models [3]. We use the Quora dataset to test the performance of our model. The dataset consists of 404,351 pairs of questions categorized into 'duplicates' (149,306 question pairs) or 'not duplicates' (255,045 question pairs). Sample questions from the dataset are shown in Figure 1. We consider duplicate questions in the dataset to be semantically equivalent. This is in line with Quora's duplicate question policy [2]. In order to capture an attention weighted representation of each individual question, we experiment with textual attention on top of this architecture to improve performance. We also explore different ways of combining the questions to learn a joint attention based model. After explaining our methodology, we present the results obtained using this architecture on the Quora dataset followed by a detailed error analysis based on the performance of our model.

2. Related work

Semantic matching for sentences or questions has been a widely studied problem in natural language processing and understanding. Dey *et al*. [2] used conventional techniques like SVMs on top of custom features and extensive preprocessing as deep learning techniques need substantial amount of data. Due to more recent advancements in deep learning techniques, many attempts have been made in the past few years to use deep learning to solve this problem. Majority of these attempts use the Siamese architecture to encode the two sentences to be compared using a shared network. Bogdanova *et al*. [1] use a distance metric over this representation to show significant improvement over the traditional techniques. The best performance on the SemEval-2015 dataset for the same problem has been achieved by Sanborn *et al*. [6] by using RNNs. A disadvantage of the Siamese network is that there is no explicit interaction between the two sentences during the encoding phase. "Matching-aggregation" based frameworks have been proposed to capture these interactions. [8]

There are many competitive results [3] for the Quora dataset in particular. The state-of-the-art model for this dataset was presented earlier this year by Wang *et al*. [9]. In this paper, we will be presenting simpler Siamese neural networks than those proposed by Wang *et al*. to further op-

[1] https://data.quora.com/First-Quora-Dataset-Release-Question-Pairs

[2] https://www.quora.com/Whats-Quoras-policy-on-merging-questions

[3] https://github.com/bradleypallen/keras-quora-question-pairs

id	qid1	qid2	question1	question2	is_duplicate
18	37	38	Why are so many Quora users posting questions that are readily answered on Google?	Why do people ask Quora questions which can be answered easily by Google?	1
19	39	40	Which is the best digital marketing institution in banglore?	Which is the best digital marketing institute in Pune?	0
20	41	42	Why do rockets look white?	Why are rockets and boosters painted	1
21	43	44	What's causing someone to be jealous?	What can I do to avoid being jealous of someone?	0

Figure 1: Sample from Quora's Question Pair dataset.

timize the use of this architecture for the task of semantic question matching.

3. Methodology

In the following sections, we will first elaborate on the data preprocessing steps, followed by descriptions of all the different models used for the problem.

3.1. Data Preprocessing

In order to process the data, we use the default tokenizer provided in Keras to obtain tokens for each question in the dataset. We then represent each token with its corresponding ID in the GloVe word embeddings [4] vocabulary. For the tokens not present in the GloVe vocabulary ($\sim 30\%$ of them), we use zeros for their representation. We use the GloVe matrix as an initialization to the Embedding layer before the LSTM layer in all our models.

Questions in the dataset vary significantly in terms of length from 0 upto 237 words. To avoid complexity and use standard matrix operations in our computation, we make all the questions have a fixed length of 40, which is a hyperparameter for our model. Shorter sentences are appended with 0s in the beginning, while the longer ones are truncated to this length.

Lastly, the model can overfit on the data due to the fact that the dataset is relatively smaller. We use 20% of the question pairs in our training data for validation. We monitor the performance improvement on the validation dataset at every epoch and save the best-performing model on the validation dataset while training.

3.2. Models

3.2.1 Network with Concatenation

Our first approach is to use an Long Short Term Memory (LSTM) Recurrent Neural Network to learn the representation for both the questions in the pair. As shown in Figure 2(a), we then concatenate the hidden state vectors at the last time step for both the questions and provide it as an input to a Multi-Layer Perceptron (MLP) to learn the similarity between the questions. We also experimented with a Bi-directional LSTM layer to model the dependencies for both the directions in individual questions. In this model as well, we use the last state for context representation, which therefore gives twice as much information in the case of BiLSTM as an input to the MLP.

3.2.2 Network with Cosine Similarity

Taking motivation from the distance metric based approaches that have been used for this problem, we tried to directly model cosine similarity between the last time-step hidden state vectors from the Bi-LSTM for both the questions using a separate layer as shown in Figure 2(b). This layer computes the following equation -

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

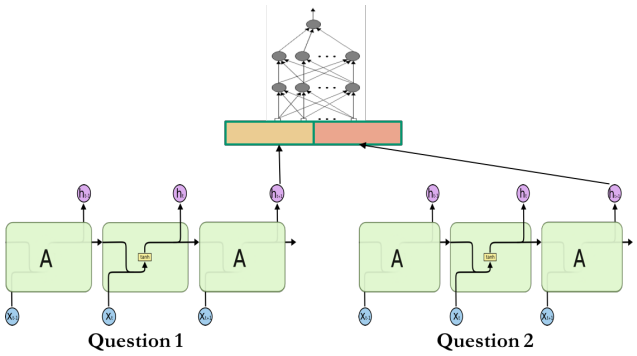
where, $\cdot$ is the dot-product, $\|\cdot\|$ is the L2-norm operator. This layer is differentiable and helps to directly learn the similarity between the vector representations of both the questions.

3.2.3 Network with Individual Attention

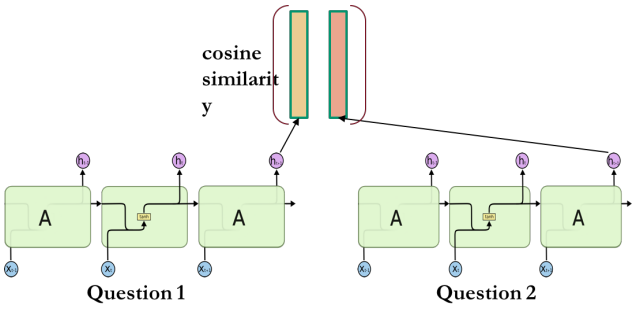
We use textual attention individually over both the questions. [5] shows a simple way to apply the attention mechanism which produces a single vector c from an entire sequence of vectors, formulated as follows -

$$e_t = a(h_t)$$
$$\alpha_t = \frac{\exp(e_t)}{\sum_{i=1}^{T} \exp(e_i)}$$
$$c = \sum_{t=1}^{T} \alpha_t h_t$$

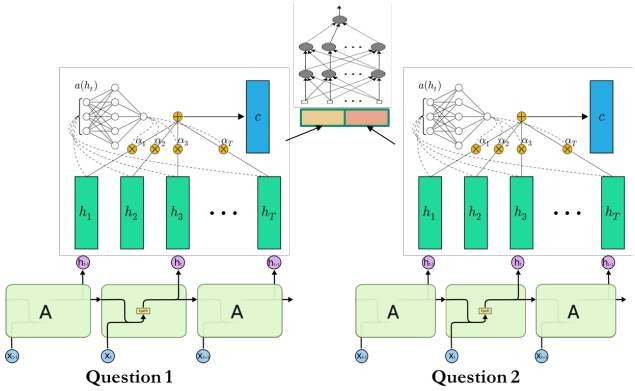
where, α_t are the learnt attention vectors over time and h_t are the hidden state vectors from the LSTM layer at every time step. We do this separately for each question and use the resultant c vectors from both the questions as their representation vector. We then concatenate the c vectors for



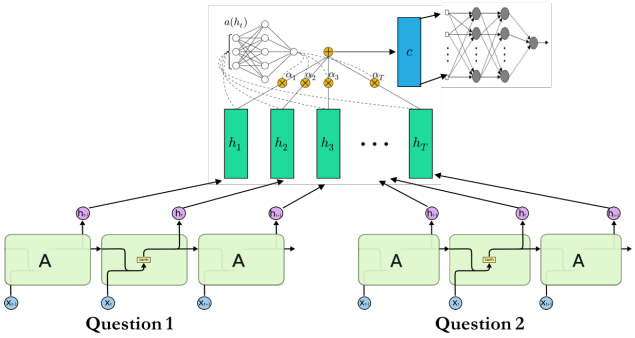
(a) Network with Concatenation



(b) Network with Cosine Similarity



(c) Network with Individual Attention



(d) Network with Combined Attention

Figure 2: Illustrations of all the model architectures used for our experiments.

both the questions and provide it as an input to a Multi-Layer Perceptron (MLP) to learn the similarity between the questions. This formulation helps to model the long term dependencies between the words in the question in a relatively better way. Since each question has 40 words, the last time-step's hidden state vector h_T is not enough to capture enough context in the question. This model is illustrated in Figure 2(c).

3.2.4 Network with Combined Attention

In order to model the attention jointly between both the questions, we modify the above architecture as shown in Figure 2(d). Here, we use a single attention mechanism to learn attention over vectors $h_1, h_2, ..., h_T$ from both the questions together. Hence, we concatenate all the hidden state vectors from both the questions to learn weighted attention for both questions as a single vector. We then use the c vector and provide it as an input to the MLP to learn the similarity between the questions.

3.3. Loss

For all the above models, the output is the probability of the question pair being duplicates. For the models in which we have an MLP for the final layer, we apply a sigmoid over the output. For the distance metric based model, the cosine similarity score is used directly as the probability output. We use binary cross entropy loss for all our models.

4. Experiments and Results

Table 1 shows the performance for all the models (and their variants) described in Section 3.2 on the Quora dataset. We provide the accuracy for Wang *et al*. state-of-the-art BiMPM model [9] for comparison. We also provide the accuracy score for the best non-neural network based model for the problem, which uses hand-picked features along with Xgboost [7].

We suspect that the cosine similarity based model didn't perform as well because the last hidden state vectors of both the questions may not be sufficient to capture the entire context for similarity calculation. We believe that the non-linearity in the final MLP Layer for the model described in

Method	Accuracy (in %)	F1 (Non-Duplicate)	F1 (Duplicate)
LSTM with Concatenation	82.62	**0.8649**	0.7563
Bi-LSTM with Concatenation	82.87	0.8639	0.7687
Bi-LSTM with Cosine Similarity	78.56	0.8195	0.7361
BiLSTM with Individual Attention	**83.20**	0.8617	**0.7860**
BiLSTM with Combined Attention	81.53	0.8566	0.7406
BiMPM* [9]	**88.17**	-	-
Hand-picked features with Xgboost* [7]	81.4	-	-

Table 1: Accuracy and F1 score (for both duplicate and non-duplicate class for all our models. The last two entires (marked with '*') have been added to facilitate comparison with state-of-the-art deep-learning and non-deep-learning based models for the Quora dataset.

3.2.1 helped, leading to better performance for this model as compared to directly using cosine similarity. The LSTM with individual attention is our best performing model with an accuracy of 83.2%. This model is capable of individually capturing the attention over both the questions to model the context vector better. For the combined attention model, we suspect that there is a major shortcoming which leads to a drop in performance. In this model, there is no explicit mechanism for the attention layer to distinguish which hidden state came from which question in the pair as both of them are trivially combined.

4.1. Error Analysis

We performed a detailed error-analysis to gauge where our best-performing model might be lacking. We found that

- For many duplicate questions which the model couldn't identify, there are several words which are out of GloVe's vocabulary. These tokens need better representation so that the model might be able to label them correctly. This arises from the dataset containing a vast majority of proper nouns.
- There are some questions which are not similar in understanding but have many overlapping words or phrases. For instance, the questions - 'How do I contact Amazon?' and 'How do I contact Amazon jobs?' are very similar. The model falsely labels these as duplicates.
- We also found some ambiguous labels in the dataset. For instance, the pair - 'What to do in my life, I dunno?' and 'What should I do with my life' seem to be duplicates but are marked otherwise in the data.

5. Conclusion

Our work is an extension of previous deep-learning based attempts to solve the problem of semantic sentence/question matching. In this paper, we presented few of our best-performing models that were able to achieve competitive results on Quora's 'Question Pairs' dataset. First, we show a simple model which encodes the two questions using a shared LSTM and uses an MLP to find similarity between the encoded representations. Second, we use a distance-metric based approach instead of the MLP on the same representations. Lastly, we experiment with attention based techniques for the similarity computation. We show that the model which uses individual attention achieves the best accuracy and F1 score (for duplicate examples).

We also present some observations from our results followed by error analysis to further the understanding of our model. As future work, there are several possible extensions to our research. Building upon the individual and the combined attention based models, there is scope for more research on jointly modeling attention for both the sentences being compared. Several other optimizations like character encodings to help with out of vocabulary words, matching against all time-steps of one sentence from multiple perspectives, etc. could be explored to further improve upon our results.

References

[1] D. Bogdanova, C. Santos, L. Barbosa, and B. Zadrozny. Detecting semantically equivalent questions in online user forums. *Proceedings of the 19th Conference on Computational Natural Language Learning*, 2015.

[2] K. Dey, R. Shrivastava, and S. Kaushik. A paraphrase and semantic similarity detection system for user generated short-text content on microblogs. *26th International Conference on Computational Linguistics*, 2016.

[3] H. Kamper, W. . Wang, and K. Livescu. Deep convolutional acoustic word embeddings using word-pair side information. *CoRR, abs/1510.01032*, 2015.

[4] R. Pennington and C. Manning. Glove: Global vectors for word representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 2014.

[5] C. Raffel and D. Ellis. Feed-forward networks with attention can solve some long-term memory problems. *International Conference on Learning Representations*, 2016.

[6] A. Sanborn and J. Skryzalin. Deep learning for semantic similarity. 2015.

[7] A. Thakur. Is that a duplicate quora question? Available at `https://www.linkedin.com/pulse/duplicate-quora-question-abhishek-thakur/`, year = 2017.

[8] S. Wang and J. Jiang. A compare-aggregate model for matching text sequences. *arXiv preprint arXiv:1611.01747*, 2016.

[9] Z. Wang, W. Hamza, and R. Florian. Bilateral multi-perspective matching for natural language sentences. *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, 2017.