Semantic Duplicate Detection in Questions Using Deep Learning

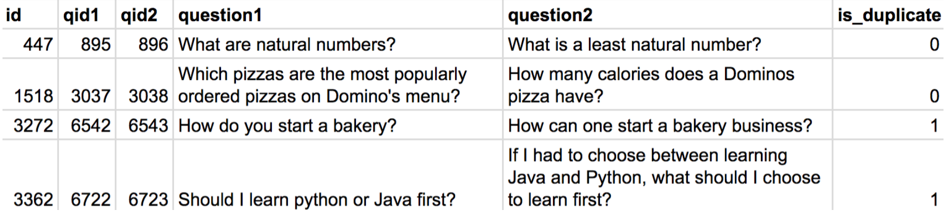
Pooya Khaloo, Ananya Ganesh

Problem

Detecting if two questions are similar to each other semantically is a significant problem to services like Quora or Stackoverflow. These services are based on questions and answers between their user and they need to make sure that their services provide unique questions to prevent the writers answering the same question multiple times. StackOverflow's real time duplicate detection is particularly interesting because it directs the user to a similar question even before the query is posted, which is useful and saves time. Detecting similarity between questions should take place at a semantic level so we can detect that "How can I make pizza?" and "What is a good recipe for pizza?" are actually similar questions. One way to tackle this problem is using deep learning which we describe in more detail in the next section.

Dataset

We are going to use datasets released by Quora which consist of over 400,000 lines of potential question duplicate pairs. The question pairs are annotated with a label indicating whether both the questions are the same or not. The sample of data is shown in Figure 1



id	qid1	qid2	question1	question2	is_duplicate
447	895	896	What are natural numbers?	What is a least natural number?	0
1518	3037	3038	Which pizzas are the most popularly ordered pizzas on Domino's menu?	How many calories does a Dominos pizza have?	0
3272	6542	6543	How do you start a bakery?	How can one start a bakery business?	1
3362	6722	6723	Should I learn python or Java first?	If I had to choose between learning Java and Python, what should I choose to learn first?	1

Figure 1: Sample of Quora question pair dataset.

Methodology

First we will develop a baseline approach which is a simple word matching between two questions, with a threshold on intersection over union to indicate similarity. However, we expect this to fail for many question pairs which are semantically similar but use different words. The current model used in production at Quora focuses on extensive handcrafted features, which are then fed to a random forest classifier. This kind of approach is detailed in [1] which uses seven handcrafted features and applied five different learning algorithms on those features. However, improvements in performance can be obtained by shifting to neural models, which are capable of learning features without explicit extraction.

Following this, we will encode the target question pairs as vectors using pre-trained embeddings such as Glove [2] and word2vec [3]. We plan to use both in a comparison study to see if they perform the same. Using these embeddings, we will then explore deep learning models that can learn to identify similar pairs. To incorporate context, we want to apply recurrent neural networks, specifically Long Short-Term Memory (LSTM) network [4]. An interesting architecture that we want to experiment with is Siamese

LSTM by Mueller and Thyagarajan [5]. We will also explore the effect of adding an attention based weighting mechanism [6] on the quality of our model.

Evaluation

The model is expected to provide a score indicating how similar the two questions are, which we will threshold to predict a label indicating whether they are duplicates. The metric for evaluation is to compute precision and recall of the predicted labels with respect to the ground truth. We will be forming a train/val/test split of 60/20/20 from the Quora dataset by randomly sampling but with balanced classes. As mentioned in the previous section we will first develop a baseline method based on word matching, and then compare it with more complex approaches that can understand the semantic meaning behind the sentences and use that to identify similarity.

We would also like to analyze our model qualitatively, and hope that attention maps will provide information regarding which words were judged to be most important while making a decision on similarity. We think it would also be interesting to use t-SNE to visualize the input sentences in vector space. We will be experimenting with two different available pre-trained word embeddings (word2vec [3] and GloVe [2]) to observe their effects on our models.

References

[1] J. Zhao, T. Zhu, and M. Lan, "Ecnu: One stone two birds: Ensemble of heterogenous measures for semantic relatedness and textual entailment," in *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, 2014, pp. 271–277.

[2] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[3] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[4] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997. [Online]. Available: http://dx.doi.org/10.1162/neco.1997.9.8.1735

[5] J. Mueller and A. Thyagarajan, "Siamese recurrent architectures for learning sentence similarity." in *AAAI*, 2016, pp. 2786–2792.

[6] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:1508.04025*, 2015.