Duplicate Question Detection With Auto Encoder based Data Augmentation

ZHANG Lin

Supervisor: Dit-Yan YEUNG

UROP 1100 Summer 2018

The Hong Kong University of Science and Technology

Abstract

Abstract—The goal of this project is to build an AI TA which can answer questions already having duplicate pairs and corresponding answers in our database. Therefore, how to detect duplicate questions accurately has been a major problem. In this report, we first introduce our baseline method, the vanilla GRU model, to detect duplicate questions. We then propose and analyze several methods to improve the accuracy. They ca be divided into two categories: Improving the model, including using deeper neural network, LSTM networks, Convolutional neural networks, adding POS tags; Or creating augmented dataset, especially applying Variational Auto Encoder (VAE) to generating duplicate questions. Finally, we will discuss the possible future development.

I. Introduction

Duplicate questions are defined as questions that can be replied with the same answer. Whenever a student raises a question in our course forum, a qualified Artificial Intelligent teaching assistant should be able to retrieve the corresponding duplicate questions and answer. This can save us time and labor. But how to improve the accuracy of detecting duplicate questions has always been a popular research problem. The biggest challenge lies in the diversity of human languages. Specifically, the same question can always vary in both grammatical structure and semantic structure which, in most cases, depends largely on the vocabulary. Therefore, the model we build needs to capture as many these features of each sentence as possible so that it can judge correctly just like human being.

Up to now, a great number of models have been experimented to help solving this problem. Bogdanova et al.(2015) have improved the CNN model to detect duplicate questions in Ask Ubuntu Community Questions and Answers site [1]. Socher et al.(2011) have taken the advantage of parse trees and use unfolding recursive autoencoder as well as dynamic pooling to detect paraphrase in Microsoft Research paraphrase corpus [2]. Zhiguo WANG et al.(2017) also create a complex model improved from LSTM to detect duplicate questions in Quora dataset [3]. In the following paper, we would mainly explore the efficiency of RNN model and its improvement in solving duplicate questions problem in Quora dataset.

II. Baseline Model

Our baseline Model is a GRU model, which produces one output variable for each sentence, followed by a two layer neural network comparing the two outputs and decide the final answer, i.e. is(1) or not(0) duplicate question. Figure 1 is the overview of our model [4].

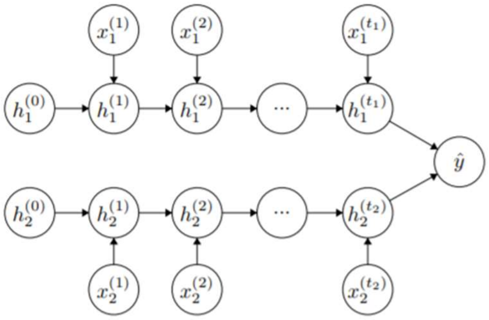

Figure 1. our baseline model

A. Model explanation

The GRU part is a Siamese GRU model. For the two-layer network, suppose we have input h1, h2 from GRU model output, we then get new input vector:

$$\boldsymbol{v} = [\boldsymbol{h}_1 \quad \boldsymbol{h}_2 \quad (\boldsymbol{h}_1 - \boldsymbol{h}_2)^2 \quad \boldsymbol{h_1} \odot \boldsymbol{h_2}].$$

Then we produce final result through the two-layer network by using formula:

$$\boldsymbol{v_1} = \mathrm{ReLU}(\mathrm{ReLU}(\boldsymbol{v})\boldsymbol{U_1} + \boldsymbol{b_1})$$
$$\hat{\boldsymbol{y}} = \mathrm{softmax}(\boldsymbol{v_1}\boldsymbol{U_2} + \boldsymbol{b_2})$$

And the final prediction is argmax(ŷ): 1 or 0

B. Dataset preparation

Our dataset is prepared with the help of Stanford Tokenizer and GloVe dataset, which forms a 300d vector representation for each word. We split the Quora dataset into 384,348-example training set, 60% of which are nonduplicate examples, 10,000-example development set and 10,000-example test set [].

C. Date preprocessing

During data processing, we first use Stanford Tokenizer to tokenize each sentence, after which we convert each word to its lower case, replace number with “num”, words outside GloVe vocabulary with “unk”. Then we construct our tok2id dictionary and replace each word with its specific id. Next we limit the sentence length to 40, prune longer sentence from tail and pad shorter sentence with “0”. Remind here the “0” id is reserved in tok2id dictionary for padding use. Then we embed each word with its GloVe vector representation, i.e. a 300d vector.

D. training method

We set all hidden variable size in GRU and Two-layer network to 300, use Adam optimizer provided by TensorFlow. Our loss is made up of cross entropy loss and l2 loss of all trainable matrices. There are also other parameters like batch size and learning rate which are omitted here.

E. Results

We achieved a result of 83.70% accuracy and 83.31% F1 score finally.

III. Model improvement

Our baseline model is a vanilla GRU model, one of the RNN model. However, the RNN model nowadays has evolved into many different types, like bidirectional RNN, LSTM model, multilayer RNN model, and even RNN model with attention. We first try those different models, hoping to see any significant improvement. Then, inspired by the popular trend and surprising achievement of using Convolutional Neural Network on computer vision, we also apply CNN to our model here.

A. LSTM model

LSTM is one popular model to replace vanilla RNN model. Similar as a GRU model, it solves the vanishing gradient problem existed in vanilla RNN. So, we replace GRU with LSTM to see the results. However, our LSTM model only achieves an accuracy of 82.01% and F1 score of 81.66%, worse than GRU model.

B. Multilayer RNN model

As multilayer RNN model is supposed to be able to decode more complex information from the inputs, we use 3-layer LSTM here. The results are below:

Model	Accuracy	F1 score

3-layer LSTM	82.38%	82.82%

Figure 2: the results of 3-layer LSTM

It can be seen from the table that Multilayer LSTM can improve the accuracy only in a small amount. The reason might be that LSTM are already powerful enough to decode much information and increasing layer can only push the result forward to the limit of LSTM model. So, we guess it is the LSTM that restrict the model's improvement rather than the multilayer structure.

C. Bi-directional RNN

Since our RNN model follows the original sentence order from first word to the last, during the processing, it may weaken the information decoded from the words in the head as time step goes by. Therefore, we decided to use Bi-directional GRU and LSTM here, which can add more weights to the head words compared to the unidirectional GRU. So, our final output from each sentence will be the concatenation of *hf* and *hb*, where *hf* is the forward direction output, *hb* is the backward direction output. The result is quite satisfying for GRU, but no improvement can be seen in LSTM.

Model	Accuracy	F1 Score
Bi GRU	84.71%	84.40%
Bi LSTM	82.04%	81.00%

Figure 3: the result of bidirectional GRU and bidirectional LSTM model

D. Add attention

Attention is a mechanism that has been widely used in sequence to sequence model, especially in machine translation. Due to its power function to decode information from inputs, we decide to use it here. We implement a Bidirectional LSTM model with attention, and the results show that attention can bring quite a lot of improvement to our model.

Model	Accuracy	F1 Score
Bi-LSTM with Attention	85.53%	85.67%

Figure 4: the results of bidirectional LSTM with attention model

E. Convolutional Neural Network

We use the idea from Kim [5] and applied the same CNN model on our task to substitute the RNN model. At last we achieved an accuracy of 81.49% and F1 score of 80.72%.

F. Summary

We can see from the result above that our duplicate task relies much on the model, specifically, the model's ability of decoding information from the inputs into the hidden variable. As mentioned in Part I, whether two sentences are duplicate depends on their grammatical structure and semantic structure. Multilayer model improves the model because it can find out more relation between words. Bi-directional model outperforms the vanilla model because, on one hand, the reversed direction not only provides more information on grammatical structure, but also solves the problem of memory loss during time steps, thus also provides more information on semantic meaning; On the other hand, the larger size of hidden variable is able to decode more information from inputs. Attention has powerful ability to decode information from inputs since it reinforces the information retrieval from each input to hidden variable by comparing hidden variable with each input word. The information it offers is thought to be more like semantic information. As for CNN, it relies more on the certain smaller phrase rather than the structure of the whole sentence, so unlike the sequence order RNN model, it loses some information on grammatical structure.

IV. Adding Features

Inspired by last part, we think the limit of RNN model lies in that they can not decode plenty of grammatic structure meaning of the sentence. So we use Part-of-Speech tagging (POS) to tag each input word. We use NLTK pos tag function, which has over 40 tags, and classify them into 10 tags. Together with the word embedding, we add one hot vector for each input word based on its tag. So now for each word, we have 310d vector representation. Then we run our Bi-GRU model again. However, the results shows that the extra features actually make the model in a mess, and the final accuracy is only over 75.06% and F1 score is 79.35%

V. Augment Data

As introduced by the original report [4], using augmented data can enlarge the training dataset, further improve the accuracy. So, we adopt a method different from the method in [4] to generate augmented data. The method is as following:

1. For original duplicate question pairs, we reverse the questions' order to create positive augmented examples.
2. To generate negative augmented examples, for each question in original dataset, we use question likelihood [6] as a distance function to measure the distance between it and its neighboring 100 sentences (except for its pair question). We then choose the 3rd most similar sentence as its negative pair question. Here, choosing 3rd most similar question is to ensure that they are nonduplicate questions, at the same, let them be similar so that our model can learn more.

By using this method, we enlarged the original training set from 384,348 examples to 523,772 examples. By using this augmented dataset, our Bi-GRU model was able to achieve a higher accuracy of 85.89% and higher F1 score of 85.87%

VI. Generate Duplicate Questions Using VAE

Inspired by the improvement of accuracy in Part V, we want to augment more sentences. However, how to generate Duplicate Questions is a big problem. So, we decided to use Variational Auto Encoder (VAE) [7] to generate duplicate questions.

A. Sequence-to-Sequence Model

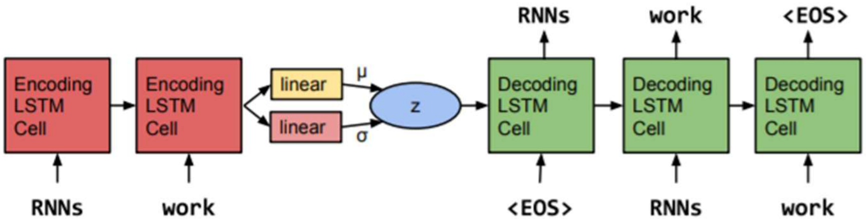

Figure 5: VAE Sequence-to-Sequence Model

To produce *z* from output of LSTM, we use a two-layer neural network. To create input to decoding layer from latent variable *z*, we also use a two-layer neural network.

B. Training Method

During training, we expect the input and output sentences to be the same. The train loss consists of two parts, which are show below:

$$L(\theta; x) = KL(q(z \mid x) \mid\mid p(z)) + E_{q_\theta(z|x)}[\log p(x \mid z)] \leq \log p(x)$$

What is worth mentioning is that the training has several tricks.

1. KL Annealing

   Gradually increase the weight of KL loss from 0 to 1 during training. It is said in

2. Dropout

   Replace the input to decoding layer with 'unk' embedding to force the model to learn more from latent variable

C. Results

First, we train an Auto Encoder using the same model except for leaving out the middle two neural networks. Result shows that the AE model can generate identical questions. We then trained the VAE model and found that the generated sentences are quite unstable. Actually, it is very hard to generate duplicate questions since the generated sentence are often different from the original sentence in those important nouns and adjectives, which makes the meaning of the sentence quite different, too.

	Input sentence	Generate sentence
1	what is the meaning of life ?	what is the purpose of life ?
2	which is the best way to learn coding ?	what is the best way to lose weight ?
3	how do i get web design clients ?	how do you get a good skills ?
4	what is your name ?	what is your favorite story ?
5	can you give me some suggestions on study ?	is you ever to to in a num ?

Figure 6: 5 samples from VAE output

We can still generate duplicate questions for some very simple questions as show in 1. But still, the system will generate many nonduplicate questions since the difference in nouns, as 2,3,4 shows. And even worse, the system can also generate some quite different sentences as 5 shows

D. Conclusion

As shown in the result, the VAE system is also very fragile to generate duplicate questions. In application, we can leave out those quite different sentences as example 5 shows by measuring the similarity of generated sentence and original sentence using the simple distance function, e.g. the question likelihood function, leaving out those sentences with very low score. But examples like 2,3,4 in the figure are quite difficult to deal with since the similarity score of these sentences will still be very high.

VII. Conclusion and Future Work

The potential of a system to detect duplicate questions lies in its ability to learn and compare both the grammatical and semantic structures of the two input sentences. Our baseline, the GRU model, has already achieved good results in these two but still can be improved. We analyzed several ways to improve it. Using multiple layer RNN can decode more information from inputs, more on semantic meaning of the sentence. Using Bidirectional structure will decode more information both on grammatically and semantically. And adding attention can greatly improve the system's ability to learn from the sentence. But overall these are all the RNN models, keeping improving them will only make the result converge to the ceiling. Besides, the result of CNN is not so good since it is weak to summarize information from sentence, which is a kind of sequence structure. Adding extra features has the potential to further improve the accuracy, but the choice of features is a question deserving thinking. In our experiment, we simply add pos tag as features, but it is still hard for the model to learn from those pos tags since the structures of the sentence, both grammatically and semantically, are still note clear and even messier. Using VAE to generate duplicate questions is another topic, the challenge is that how to know if the generated sentence is the duplicate of the input sentence? And even for VAE, it is still very important to capture the structure of the sentences. Generated questions like example 2,3,4 in the last figure show this kind of problems.

During experiment, we are using some models to try to capture the structure of the sentences, and even add pos tag to help the system to capture these differences. But in fact, we can directly make the system see the structure by inputting these structures, i.e. the tree structure.

Our future work will be using tree structured model to process input sentence. Currently we have already had tree-LSTM, e.g. child-sum tree LSTM and nary tree LSTM [8], to be the model. The tree can be constructed by using dependency parse, constituency parse and some other semantic parser. Our VAE can also be improved by using tree structure VAE. Hopefully these tree structure will provide information on grammatic structure and semantic structure for the system in a more straightforward way.

[1] Dasha Bogdanova, C´ıcero dos Santos, Luciano Barbosa, Bianca Zadrozny. **Detecting Semantically Equivalent Questions in Online User Forums**. CoNLL 2015.

[2] Richard Socher, Eric H. Huang, Jeffrey Pennington, Andrew Y. Ng, Christopher D. Manning. **Dynamic Pooling and Unfolding Recursive Autoencoders for Paraphrase Detection**. Advances in Neural Information Processing Systems 24, 2011.

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

[4]. Yushi Homma, Stuart Sy, Christopher Yeh. **Detecting Duplicate Questions with Deep Learning**. Stanford University CS224n final project.

[5] Yoon Kim. **Convolutional Neural Networks for Sentence Classification**. EMNLP 2014.

[6] WEI EMMA ZHANG and QUAN Z. SHENG. **Duplicate Detection in Programming Question Answering Communities**. ACM Transactions on Internet Technology (TOIT) - Special Issue on Artificial Intelligence for Security and Privacy and Regular Papers, Volume 18 Issue 3, May 2017. Article No. 37.

[7] Samuel R. Bowman, Luke Vilnis**. Generating Sentences from a Continuous Space**. SIGNLL Conference on Computational Natural Language Learning (CONLL), 2016.

[8] Kai Sheng Tai. **Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks**. ACL 2015

Appendix:

The result of testing

Model	*Accuracy*	*F1 score*
GRU	*83.70%*	*83.31%*
LSTM	*82.01%*	*81.66%*
3 Layer LSTM	*82.38%*	*82.82%*
Bi GRU	*84.71%*	*84.40%*
Bi LSTM	*82.04%*	*81.00%*
Bi LSTM with Attention	*85.53%*	*85.67%*
CNN	*81.49%*	*80.72%*
POS Tagged Bi GRU	*75.06%*	*79.35%*
Bi GRU + Augmented data	*85.89%*	*85.87%*