Question and Answer Classification with Deep Contextualized Transformer

Haozheng Luo, Ningwei Liu, Charles Feng

Recent literature has focused on the Standford Parse Tree and how it has been used for Question and Answer problems in Natural Language Processing. This parser tree with deep learning algorithms has analyzed the makeup of question and answer classifications. In this study, the authors have built a model using a Deep Contextualized Transformer that can manage some aberrant expressions. We conducted extensive evaluations on SQuAD and SwDA datasets and our model showed significant improvements to QA problem classifications for industry needs. Further analysis investigated the impact of various models on the accuracy of the results. Research outcomes showed that our new method is more effective in solving QA problems with a higher accuracy of up to 83.1% compared to other models.

Keywords: QA Classification, NLP, Self-attention, Self-attention.

1 Introduction

The Question and Answer system (QA) is widely used in the industry. Every week, one company faces thousands of questionnaires for the products they launch. QA is a massive problem in Natural Language Processing (NLP), including the application of problem answering, sentence recognitions, etc. There are several types of problems, such as wh-questions, statement questions, statements and other question patterns. Each type of question has a corresponding label for a question or statement. In this study, we want to discover a better algorithm to analyze the question and answer data from the huge text files. Earlier work in this field mainly used the Bag-of-words (BoW) technique to classify sentence types (Dani et.al, 2014). Many recent studies have adopted supervised, deep-learning methods concerning question and answer classification and have shown promising results (Lee and Dernoncourt, 2016). However, most of these approaches have treated the sentence as a text classification without considering the context of the writing across sentences or interdependently; therefore, this method is unable to reflect conceptual dependencies of the words within the sentences. In reality, the different order of the same words in a sentence can have very different meanings. As a result, it is necessary to determine a high compatibility algorithm to classify question and answer sentences by considering all sentence configurations.

This research draws on some recent advances in NLP research such as BERT (Jacob et al., 2018) and Elmo (Peters et al., 2018) to produce a sentence classification model to quickly and correctly pick out the question and answer sentence from the target text. Compared with regular algorithms for treating QA problems such as word2vec (Pennington et al.), this self-attention algorithm can perform contextualized word representation to obtain contextualized word meaning from the sentences. As a result, with BERT and Elmo, we can quickly judge the contexture relationship of the sentence and figure out what kind sentence type it might be. Specifically, we have used a hierarchical deep neural network with a self-attention algorithm to process different types of question and answer text, including statement questions that are a specific type of question in the questionnaires. This research aims to achieve state-of-the-art outcomes for classifying the QA problem. As a result, we mainly contribute to: 1.) a huge the improvement of performance on QA classification problems with the self-attention method, especially one such as BERT; 2. demonstrating how performance could be improved with a combination of different levels of models including the hierarchical deep neural network for classification, the self-attention model like BERT for single word embedding, and the previous label of the training data with the SQuAD dataset; and 3. exploring different methods to find a high compatibility method to classify the QA problem.

2 Related Work

We focus on two primary methods used in recent research. One treats text as text classification, in which each word is classified in isolation, while another treats the text using Contextualized Word Representation Algorithms, such as BERT with self-attention or Elmo.

Text Classification: Lee and Dernoncourt (2016) build a vector representing each utterance and use either RNN or CNN to predict the text details to classify the sentence type.
Self-attention: Jacob et al. (2018) used the BERT, and Peters et al. (2018) used Elmo to embed the text into the vector to give the contextual relationship of the sentence for each word. Along with these two tools, we use RNN-based or CNN-based hierarchical neural networks to learn and model multiple levels of word.

3 Model

The task of QA classification takes the sentence S as an input, which varies the length sequence of the word U=$\{u_1, u_2, u_3, \ldots, u_N\}$. For each word $u_1 \in$ U, there has a length value of $l_i \in$ L and a corresponding target label $y_i \in$ Y, which represents the QA's result associated with the corresponding sentence.

Figure 1 shows the overall architecture of the model, which involves several main components. (1) A self-attention algorithm to encode the sentence with the self-attention, (2) A Combination-level RNN to handle the output of the encoding and to classify the label of the sentence. We describe the details below.

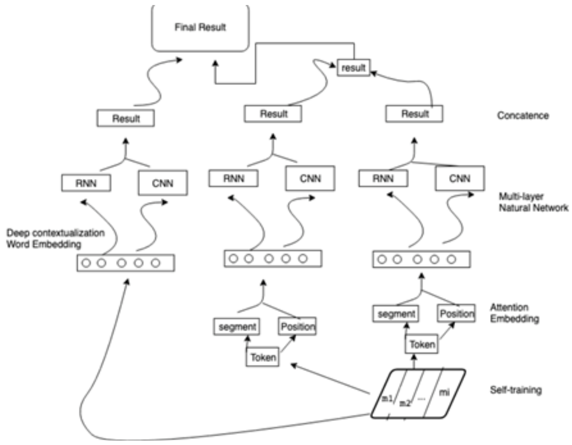

Figure 1: The graph of the model Architecture

3.1 Context-aware Self-attention

Our self-attention algorithm encodes a variable-length sentence into a fixed size. There are two types of the algorithm; one based on Self–Attention and another based on deep contextualization word representation.

3.1.1 Deep contextualization word representation

This model uses the BiLM to consider the different position of words within the sequence. Inspired by Peters et al. (2018), we use PCA and t-SNE to reduce the dimensions from a higher level to reduce the dimensions from a higher level to a lower level. Then, we use the Combination-Level RNN (Section 3.2) which provides us with the previous hidden state of the encoded word. It provides us the contextual relationship in the sentences and combines all hidden states of words in sentences. After that, the deep modifications contextualization word representation encoder encodes the combination into the 2-D vectors of each sentence. We follow the instruction of Peters at el. (2018) to explain below.

A word t_i, which is the sequence of the sentence, is mapped onto the embedded layer. The deep contextualization representation uses BiLM to combine the forward and backend LM. The formulation of the process is as follows:

$$\sum_{k=1}^{N} \big(\log p(t_k \mid t_1, \ldots, t_{k-1}; \Theta_x, \overrightarrow{\Theta}_{LSTM}, \Theta_s) \\ + \log p(t_k \mid t_{k+1}, \ldots, t_N; \Theta_x, \overleftarrow{\Theta}_{LSTM}, \Theta_s) \big) .$$

Moreover, we weigh the performance of the model with computing as indicated here:

$$E(R_k; \Theta^{task}) = \gamma^{task} \sum_{j=0}^{L} s_j^{task} \mathbf{h}_{k,j}^{LM}. \quad (1)$$

In (1), the s_j^{task} represents softmax-normalized weights, and the scalar parameter γ^{task} allows the task model to scale the entire vector. In the simple case, the representation would choose the top layer and $E(R_k) = \mathbf{h}_{k,j}^{LM}$.

3.1.2 Self-Attention

For each word in the word, we would use some Self-Attention model to encode them. The most popular Self-Attention model base is BERT (Devin et al. 2018). The model will encode a variable-length sequence using an attention mechanism that considers the different position token and segment within the sequence. Inspired by Devin et al. (2018) and Tran et al. (2017), we apply the Combination-Level RNN (Section 3.2) into a self-attractive encoder (Lin et al. 2017). We use the 24 layers and 1024 Hidden Uncased BERT also with the RobertaBERT as the base of the embedding to encode the context to the 3D tensor. We follow the instruction of Vipuls Raheja and Joel Tetreault (2019) and Joel Tetreault and Liu et al. (2019) to explain the modification mentioned below.

The word t_i is also mapped onto the embedding layer and results in s-dimensional embedding for each word in the sequence based on the Transformer (Vaswani et al. 2017). Then, the embedding is put into the bidirectional-GRU layer.

Vipul Raheja and Joel Tetreault (2019) describe the contextual self-attention score as:

$$S_i = W_{s2} tanh(W_{s1} H_i^T + W_{s3} \overrightarrow{g_{i-1}} + \mathbf{b}) \quad (2)$$

Here W_{S1} is a weight matrix, W_{S2} and W_{S3} is a matrix of parameters. b is a bias of the vector represented in Equation 2. This can be treated as a 2-layer MLP with bias, and d_a with a hidden unit.

3.2 Combination-level RNN

The word representation hi from the past two models are passed into the combination-level RNN. Based on Figure 1, we would pass all of the hidden layers concatenated into a final representation Ri of each word. The previous model step will help us build a encoding vector to represent the relationship of the each words, and in order to consider all the words in the sentence together to make the classification, it is necessary to make the combination RNN model.

This is more suitable for the problem classification to put the layers with the proper percentages in the final representation. Then, we place the result into the combination model layer to figure out the relationship between the label and the context of the words. This method is not independently decoding the label of the words; it should consider all of the relationships of the sentences. Then, it should determine the most related decoder to decode them to the related labels. The combination-level RNN would also have the function to supervise the labels and improve the accuracy of the classification of our model.

3.3 Super-attractive

The model that we use combines the final representative of the combination for hidden layers via self-attention. It can help us figure out what the labels of those words are and produce the results. The score we compute for the algorithm is to calculate the accuracy of the correct labels in the classifications as Hossin M. and Sulaiman M.N. (2015) suggest. Also, we apply an advanced check for the question and answer problem. For sentences without clear results, we put them into the parser tree for another classification. The parser tree we use is based on Huang (2018). We use its Tensor Product Representation to rebuild our parser tree for our model. The original Stanford Parser Tree (2008) is good to classify the relationship of the sentences. However, in our model, we use the Bi-LSTM with the attention algorithm to rebuild the parser tree and get the tree graph with POS tags. This is useful to calcify the structure of the sentence. After that, we use the graph we obtain to analyze the structure of words and produce the classification of the unsure sentence in the document. Finally, we determine the combination result for the users to check the question and answer problems.

4 Data

We evaluate the accuracy of the classification model with one standard dataset - the Switchboard Dialogue Act Corpus (SwDA)

(Jurafsky et al., 1997) consisting of 43 classes (listed as , and we make a program to create the sentences based on the data with the Stanford Question Answering Dataset to use self-attention for the task. The Natural Language Toolkit Dataset (NLTK) (Steven Bird and Edward Loper, 2002) is another significant resource for the test case. We use the nltk.corpus.nps_chat dataset as data for the experiment. We then use the training, validation, and test splits as defined in Lee and Dernoncourt (2016).

Table 1 shows the statistics for both datasets. There are many kinds of labels of the class to classify the kind of sentences they are. There are some special DA classes in both datasets, such as Tag-Question in SwDA and Statement-Question in NLTK. Both datasets make over 25% of the question type labels in each set.

Data set	Train	Validation	Test	|T|	|N|
SwDA+ SQuAD	87k	10k	3k	43	100k
NLTK	8.7k	1k	0.3k	15	10k

Table 1: Number of Sentences in the Dataset. |T| represents the number of classes and |N| represents the sentence size

5 Result

We have compared the classification accuracy of our model with several other models (Table 2). For methods using attention and deep contextualization word representation in some approaches to model the sentence of questionnaire documents, some of them use the self- attention word representation for the task. However, they did not perform as well as our model. All models and their variables were trained eight times, making an average for the performance as a result. And we find these previous algorithms did not perform as well as our model. Our model is better than Raheja and Tetreault (2019) by 0.4% in SwDA dataset after measuring its accuracy score and 3.9% for the Li and Wu (2016) methods in SWQA dataset. It also beats the TF-IDF GloVe baseline by 17.2% in SwDA.

Model	SwDA+ SQuAD	NLTK
TF-IDF GloVe (2014)	66.1	70.3
Li and Wu (2016)	79.2	-
Peters et al. (2018)	76.3	-
Raheja and Tetreault (2019)	82.7	85.8
Lee and Dernoncourt (2016)	75.9	77.4
Our Method	83.1	85.5
RoBERTa	82.2	84.7

Table 2: QA Classification Accuracy of the different approaches

The improvements based on our model have a significant meaning for other models. However, the performance in NLTK is still not as good as that of the Raheja and Tetreault (2019) model. The reason for the lower accuracy is dependent on the contextual details and label noisy inside the dataset. The label noisy is caused with reason of that the NLTK dataset has the fewer and difference dialog act class, it would cause our model could not actually defined them by our model. The context in the NLTK dataset indicated the existence of some data not easily readable for the machine such as some error codes. Also, the label type in the NLTK dataset is only 35% of the label type for the SwDA ones. As a result, due to the label noisy and the contextual details, the performance of NLTK did not show significant gains over that of SwDA.

The performance of our model is more sensitive than the model used commonly for the problems, including the error code. However, it has a higher accuracy, considering the complete problem classification. In future research, we should improve our algorithm, which has a higher ability to handle the problem of the label noisy and context detail.

6 Conclusion

We developed a new model which carefully performed the QA classification and made comparisons with common-use algorithms by testing the SwDA dataset. We used different word representation methods and determined that the context details depend highly on the classification performance. For example, the reason NLTK is not as good as the Raheja and Tetreault (2019) results was because there were too many label noises and the context details which were not so easy to read. Working with attention and combination levels during the classification, which has not been previously applied in this kind of task, enables the model to learn more from the context and get more real meaning from the words than previously It helps to improve the performance of the classification for these kinds of tasks.

In our future work, we will explore more attention mechanisms, such as block self-attention (Shen et al., 2018b), or hierarchical attention (Yang et al., 2016) and hypergraph attention (Song et al. 2019). These approaches can incorporate the information from different representations for the various positions and can capture both local and long-range context dependency. Also, this approach should help with the problem of the hard-readable context, such as the problem of the NLTK dataset that causes accuracy to become lower than usual. We will seek more dataset combinations to do the question and answer classification work. We will also use RACE (Lai et al., 2017) and GLUE (Wang et al., 2019) datasets to do more test work and make more stable algorithms to solve the question and answer classification issues.

Reference

Ji Young Lee and Franck Dernoncourt. 2016. Sequential short-text classification with recurrent and convolutional neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 515–520. Association for Computational Linguistics.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489. Association for Computational Linguistics.

Dan Jurafsky, Liz Shriberg, and Debra Biasca. 1997. Switchboard SWBD-DAMSL shallow-discoursefunction annotation coders manual, draft 13. Technical report, University of Colorado at Boulder Technical Report 97-02.

Rajpurkar P, Zhang J, Lopyrev K, Liang P. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. arXiv [cs.CL].

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. July 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv [cs.CL]

Quan Hung Tran, Ingrid Zukerman, and Gholamreza Haffari. 2017. A hierarchical neural model for learning sequences of dialogue acts. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*: Volume 1, Long Papers, pages 428–437. Association for Computational Linguistics.

Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. In *International Conference on Learning Representations 2017* (Conference Track).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS 2017*, 4-9 December 2017, Long Beach, CA, USA, pages 6000–6010.

Song Bai, Feihu Zhang, and Philip H.S. Torr. Jan 2019. Hypergraph Convolution and Hypergraph Attention. arXiv [cs.CL].

Wei Li and Yunfang Wu. 2016. Multi-level gated recurrent neural network for dialog act classification. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1970–1979. The COLING 2016 Organizing Committee.

Hossin M. and Sulaiman M.N. March 2015, A Review On Evaluation Metrics For Data Classification Evaluations. *International Journal of Data Mining & Knowledge Management Process (IJDKP)* Vol.5, No.2

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543. Association for Computational Linguistics.

Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, and Chengqi Zhang. 2018b. Bi-directional block selfattention for fast and memory-efficient sequence modeling. In International Conference on Learning Representations.

Steven Bird and Edward Loper. July 2002. NLTK: The Natural Language Toolkit. arXiv [cs.CL]

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations (ICLR)*.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. Race: Large-scale reading comprehension dataset from examinations. arXiv preprint arXiv:1704.04683.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations.

Marie-Catherine de Marneffe, Christopher D. Manning. Sept 2008. “Stanford typed dependencies manual”. Stanford University NLP group.

Qiuyuan Huang, Li Deng, Dapeng Wu, Chang Liu, and Xiaodong He. Feb 2018. Attentive Tensor Product Learning. arXiv [cs.CL].

Vipul Raheja and Joel Tetreault. May 2019. Dialogue Act Classification with Context-Aware Self-Attention. arXiv [cs.CL]

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. CoRR, abs/1810.04805.

Yogatama, Dani, and Noah Smith. "Making the most of bag of words: Sentence regularization with alternating direction method of multipliers." International conference on machine learning. 2014.

Appendices for Question Classification with Deep Contextualized Transformer

A. Finetuning Hyperparameters

Hyperparam	SQuAD
Learning Rate	1e-5
Weight Decay	0.1
Epochs	7
Batch Size	8k

Table 3: Hyperparameters of Finetuning RoBERTa on SQuAD

B. Pretraining Hyperparameters

Hyperparam	RoBERTa	BERT
No. of Layers	24	24
Hidden Size	1024	1024
FNN Inner Hidden	4096	-
Attention Heads	16	16
Attention Head size	64	64
Dropout	0.1	0.1
Batch Size	8k	8k

Table 4: Hyperparameters of Pre-training RoBERTa and BERT

SWBD-DAMSL	SWBD	Example	Cnt	%
Statement-non-opinion	sd	*Me, I'm in the legal department.*	72,824	36%
Acknowledge (Backchannel)	b	*Uh-huh.*	37,096	19%
Statement-opinion	sv	*I think it's great*	25,197	13%
Agree/Accept	aa	*That's exactly it.*	10,820	5%
Abandoned or Turn-Exit	% -	*So, -*	10,569	5\%
Appreciation	ba	*I can imagine.*	4,633	2%
Yes-No-Question	qy	*Do you have to have any special training?*	4,624	2%
Non-verbal	x	*[Laughter], [Throat_clearing]*	3,548	2%
Yes answers	ny	*Yes.*	2,934	1%
Conventional-closing	fc	*Well, it's been nice talking to you.*	2,486	1%
Uninterpretable	%	*But, uh, yeah*	2,158	1\%
Wh-Question	qw	*Well, how old are you?*	1,911	1%
No answers	nn	*No.*	1,340	1%
Response Acknowledgement	bk	*Oh, okay.*	1,277	1%
Hedge	h	*I don't know if I'm making any sense or not.*	1,182	1%
Declarative Yes-No-Question	qy^d	*So you can afford to get a house?*	1,174	1%
Other	o,fo,bc,by,fw	*Well give me a break, you know.*	1,074	1%
Backchannel in question form	bh	*Is that right?*	1,019	1%
Quotation	^q	*You can't be pregnant and have cats*	934	.5%
Summarize/reformulate	bf	*Oh, you mean you switched schools for the kids.*	919	.5%
Affirmative non-yes answers	na,ny^e	*It is.*	836	.4%
Action-directive	ad	*Why don't you go first*	719	.4%
Collaborative Completion	^2	*Who aren't contributing.*	699	.4%
Repeat-phrase	b^m	*Oh, fajitas*	660	.3%
Open-Question	qo	*How about you?*	632	.3%
Rhetorical-Questions	qh	*Who would steal a newspaper?*	557	.2%
Hold before answer/agreement	^h	*I'm drawing a blank.*	540	.3%
Reject	ar	*Well, no*	338	.2%
Negative non-no answers	ng,nn^e	*Uh, not a whole lot.*	292	.1%
Signal-non-understanding	br	*Excuse me?*	288	.1%
Other answers	no	*I don't know*	279	.1%
Conventional-opening	fp	*How are you?*	220	.1%
Or-Clause	qrr	*or is it more of a company?*	207	.1%
Dispreferred answers	arp,nd	*Well, not so much that.*	205	.1%
3rd-party-talk	t3	*My goodness, Diane, get down from there.*	115	.1%
Offers, Options Commits	oo,cc,co	*I'll have to check that out*	109	.1%
Self-talk	t1	*What's the word I'm looking for*	102	.1%
Downplayer	bd	*That's all right.*	100	.1%
Maybe/Accept-part	aap/am	*Something like that*	98	<.1%
Tag-Question	^g	*Right?*	93	<.1%
Declarative Wh-Question	qw^d	*You are what kind of buff?*	80	<.1%
Apology	fa	*I'm sorry.*	76	<.1%
Thanking	ft	*Hey thanks a lot*	67	<.1%

Table 5: list of SwDA Dialog Act class and example