Answer Selection in Community Question Answering via Attentive Neural Networks

Yang Xiang, Qingcai Chen*, *Member, IEEE,* Xiaolong Wang, *Member, IEEE,* and Yang Qin, *Senior Member, IEEE,*

***Abstract*—Answer selection in community question answering is a challenging task in natural language processing. The difficulty lies in that it not only needs the consideration of semantic matching between question answer pairs but also requires a serious modeling of contextual factors. In this paper, we propose an attentive deep neural network architecture so as to learn the deterministic information for answer selection. The architecture can support various input formats through the organization of Convolutional Neural Networks, attention-based Long Short Term Memory and Conditional Random Fields. Experiments are carried out on the SemEval-2015 cQA dataset. We attain 58.35% on macro averaged F_1, which outperforms the Top-1 system in the shared task by 1.16% and improves the state-of-the-art deep neural network based method by 2.21%.**

***Index Terms*—answer selection, community question answering, deep neural networks, attention mechanism**

I. Introduction

RESEARCHES on question answering (QA) systems attract more and more attention in recent years with the explosive growth of data and the breakthroughs on machine learning. Answer selection from community question answering (cQA) (i.e. Yahoo!Answers[1]) is an important task for developing automatic QA systems. It aims at extracting useful QA pairs [1] from multiple cQA threads. The main difficulty lies in how to bridge the semantic gaps between QA pairs. Supervised machine learning forms the mainstream in tackling the above issue, such as statistical learning [2]–[4] and deep learning [1], [5]. End-to-end deep neural networks (DNN) can avoid complex feature engineering and has much stronger learning ability than former learning algorithms, hence it has become a main flow in mining QA matching [6]–[9].

Nevertheless, distinct from other matching paradigms (i.e. paraphrase [10] or machine translation [11]), matching in cQA is often heavily influenced by contextual factors. For instance, in addition to matching the question, a latter comment is usually related to some former ones (i.e. C2→C1 in Fig. 1). [9] is one of the representative studies in deep learning that integrate the contextual information. They applied Long Short Term Memory (LSTM) over the answers and gained improvements over many existing models. However, the performance is still hindered by the insufficient context modeling.

Q: I want to buy either a PS3, XBox 360 or Wii. Please suggest me which one is good and why? (question)
C1: I suggest PS3 .. its a complete entertainment station... (good)
C2: Thanks K_Sama for your valuable piece of advice. (dialog)
C3: No more suggestions? (dialog)
C4: ps3 cuz not alot of ppl buy xbox in qatar...ps3 free network xbox u have to pay monthly or by year (good)
C5: buy ps3 becoz the most popular games like GOD F… (good)
C6: Thanks alot guys. I would definitely go for PS3. (dialog)

Fig. 1. A cQA thread with comment quality tags in brackets.

In this paper, we propose an attentive deep neural network architecture so as to learn the deterministic information for answer selection from a *global* perspective, namely *A-ARC*. The architecture is constituted by several network components including Convolutional Neural Networks (CNN), attention-based LSTM and Conditional Random Fields (CRF). Through communications between the components, A-ARC is flexible on the input format as well as internal substructures. We test three variants based on the architecture. Their inputs are organized in distinct ways (i.e. the concatenation of the question and comments, the comments or the sequence of QA pairs). Moreover, they differ in the addition of other internal modules according to diverse matching strategies.

Experiments are settled on the SemEval-2015 cQA dataset. Comparisons prove that A-ARC is superior in answer quality tagging and can efficiently capture the dependencies along the QA thread. We achieve better macro averaged results compared with the baselines in both multi-class and binary classification for answer selection.

II. Attentive Neural Networks

A. Problem Definition and Model Overview

Answer selection in cQA is defined as: *Given a question and a thread of comments, tagging whether each comment is a good answer for the question or not*. Difficulty lies in that there may exist complex relations between the comments. For example, a comment is likely to be the refinement of an existing question or the complement towards a previous answer. We take the task as sequence labeling and apply *global context modeling* to address the above issue.

*Corresponding author

The authors were with the Intelligent Computing Research Center, Harbin Institute of Technology Shenzhen Graduate School, Shenzhen, Guangdong, China (e-mail: xiangyang.hitsz@gmail.com, qingcai.chen@gmail.com). The work is supported in part by National 863 Program of China (2015AA015405), NSFCs (National Natural Science Foundation of China) (61402128, 61473101 and 61272383), and Strategic Emerging Industry Development Special Funds of Shenzhen (JCYJ20140417172417105 and JCYJ20140508161040764)

Manuscript received April 19, 2005; revised August 26, 2015.

[1]https://answers.yahoo.com/

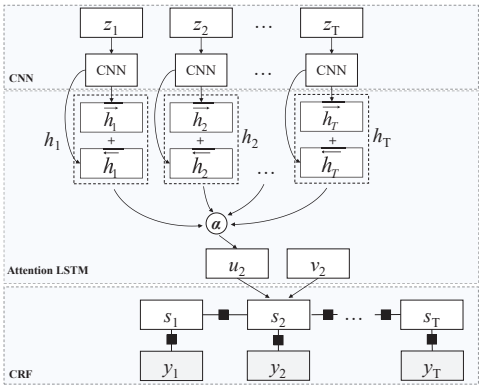

Fig. 2. The overview of the proposed attentive neural architectures.

The overview of the architecture is shown in Fig.2. The core components include CNN, attention LSTM and CRF. CNN acts as the first encoder which extracts features from each input sentence in each time step and compresses them into a fixed length vector, so that the following layers could easily make use of. Attention LSTM further encodes the compressed vectors and learns dependencies from the sequential steps. It is constituted by a bidirectional LSTM layer (h_t) followed by a soft attention layer (α), which can capture the correlations along the whole sequence in a large part. Before information propagated to the next layer, A-ARC enables the addition of external elements (v_t) so that we can add other features beyond the learned dependencies (u_t). The last component is CRF, which aims to generate final predictions considering both the encoded representation (s_t) and label transitions. Label dependency acts as another context information in addition to that learned by LSTM, based on our analysis that the occurrence of the quality labels tends to follow some regularities (i.e. a *good* answer may follow several *bad* ones).

We test three different input forms: $\{z_1, z_2, ..., z_T\}$=$\{q, a_1, a_2, ..., a_n\}$, $\{z_1, z_2, ..., z_T\}$=$\{a_1, a_2, ..., a_n\}$ and $\{z_1, z_2, ..., z_T\}$=$\{(q, a_1), (q, a_2), ..., (q, a_n)\}$, corresponding to three variants inherited from the architecture, namely *A-ARC-I* [12], *A-ARC-II* and *A-ARC-III* respectively. We believe these forms can cover most cases when providing input information to sequence labeling algorithms. To ensure the completeness of the input information, we set v_t as an individual QA matching module so as to leverage the role of the question in A-ARC-II but keep it as empty for the others. We can also extend the architecture to other variants based on different variable assignments in the network (i.e. z,u,v).

B. Convolutional Neural Networks

A sentence is encoded before it is further processed in neural networks. In this paper, we adopt the CNN sentence encoding method from [13]. Since most questions/comments are short (less than 100 words), we take each question/comment as a sentence and apply word embedding, multiple convolution and max-pooling operations.

Word Embedding Let $w_j \in \mathbb{R}^k$ denote the k-dimensional word embedding corresponding to the jth word in vocabulary V. With all-zero padding vectors to the end, the sentence of length l ($l \leqslant n$) is represented as

$$w_{1:n} = w_1 \oplus w_2 \oplus ... \oplus w_n \tag{1}$$

where $\oplus$ is the concatenation operator, n is the maximum length, and $w_{a:b}$ stands for $\{w_a, w_{a+1}, ..., w_b\}$.

Convolution To capture the local structures, a fixed length window $W^m \in \mathbb{R}^{h \times k}$ is applied on each position i to produce a new convolutional unit for layer m. Hence, the convolutional unit for position i in the mth layer is generated by

$$c_i^m = \sigma(W^m w_{i:i+h} + b^m) \tag{2}$$

where b^m is the bias factor for layer m and σ is the activation function (i.e. tanh or sigmoid). Convolutional units for more deeper layers can be produced in a similar way.

Pooling Following [13], we take a max pooling (output the maximum value for a local region) operation after convolution so as to learn the discrete representations for language.

$$c_i^m = max(c_i^m, c_{i+1}^m, ..., c_{i+d-1}^m), i = 0, d, 2d, ... \tag{3}$$

where d is the region size and set to be n-h+1 in this work. We use windows lengths of 3,4,5 as different filters for convolution and one pooling layer over the vectors correspondingly. The encoded result x_t is the concatenation of three parallel CNNs each with 100 feature maps.

C. Long Short Term Memory

RNNs with LSTM units have been proven useful on capturing long term dependencies [14]. Through the LSTM unit, information from early time steps can be preserved through the controlling of several gates and the memory cell.

Following [14], the output of LSTM at each time step h_t can be computed by the following equations:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \tag{4}$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \tag{5}$$

$$c_t = f_t c_{t-1} + i_t \tau(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \tag{6}$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \tag{7}$$

$$h_t = o_t \theta(c_t) \tag{8}$$

where i_t, f_t, and o_t denote the input, forget, and output gates respectively. σ is the activation function, τ and θ are tanh functions. Ws and bs represent weights and biases on different nodes. Following previous success in sequence modeling [15], a backward LSTM is also employed. We denote the outputs of forward LSTM, backward LSTM and biLSTM as $\overrightarrow{h_t}$, $\overleftarrow{h_t}$, and $h_t = [\overrightarrow{h_t}, \overleftarrow{h_t}]$ respectively.

D. Attention Mechanism

We reduce the information loss of LSTM through soft attention. Attention mechanism introduced by [11] can integrate the information across the whole sequence linearly, thus models the dependency relations from a global perspective. Attention is later popularized in other tasks, such as machine reading [16] and relation extraction [17].

In the proposed architecture, attention is applied over the output of biLSTM. Assume that the output of biLSTM is $\{h_j\}, j = 1, 2, ..., T$, where $h_j \in \mathbb{R}^m$ (m is the output dimension of biLSTM). For the ith step (the final output is denoted as s_i), the global context vector c_i is computed through a weighted sum of $\{h_t\}$:

$$c_i = \sum_{j=1}^{T} \alpha_{ij} h_j \tag{9}$$

and the attention weight α_{ij} is normalized by

$$\alpha_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{T} exp(e_{ik})} \tag{10}$$

where

$$e_{ij} = s_i^T \cdot h_j \tag{11}$$

The attention weight α_{ij} implies how much the previously encoded unit h_j contributes to the current step s_i. Intuitively, the attention context for a comment derives from the similarity with the question and correlations with other comments. The correlation between two comments usually reflects the relation of comment, complement or acknowledgement between them.

Distinct from the attention mechanism in machine translation [11], we simplify the attention computation by using a vector dot operation between the encoded units due to that the relations can be degenerated as similarities specific to the linear structure of cQA.

E. Conditional Random Fields

Linear chain CRF has been shown superior in many sequence labeling tasks [18]. Given an observation sequence $s = \{s_1, s_2, ..., s_T\}$. CRF jointly models the generating probability of the entire label sequence $y = \{y_1, y_2, ..., y_T\}$ by using the discriminative probabilities to y_i given s_i and the transition probabilities between adjacent labels. The original probability model of CRF is written as:

$$P(y|x; W, b) = \frac{\prod_{i=1}^{T} \psi_i(y_{i-1}, y_i, s)}{\sum_{y' \in \mathcal{Y}(s)} \prod_{i=1}^{T} \psi_i(y'_{i-1}, y'_i, s)} \tag{12}$$

where $\psi_i(y', y, s)$ are potential functions and $\mathcal{Y}(s)$ denotes all possible label sequences given s.

In this work, the observed variable s is the encoded sequence generated by previous steps. The hidden variable y is an assignment of comment quality labels. Therefore, Equation 12 is further expanded by replacing the potential function with a status function given s_i and a transition function given adjacent y_is. In accordance with the variable representations in Fig. 2 , we obtain the probability of a sequence y by:

$$P(y|s) = \frac{e^{S(s,y)}}{\sum_{y' \in \mathcal{Y}(s)} e^{S(s,y')}} \tag{13}$$

where $S(s, y)$ is defined by:

$$S(s, y) = \lambda \sum_{i=1}^{T} P(y_i = j|s_i) + \mu \sum_{i=1}^{T} T(y_i = j|y_{i-1} = i) \tag{14}$$

By using CRF, we model the label dependency globally. It is beneficial when there are possible constraints across the labels in a QA thread. We call the context modeling mechanism with attention LSTM and CRF as *global context modeling* since they both proceed from the sequence level.

F. Implementation

We pre-trained the word embeddings using word2vec [19] on the unlabeled data offered by the shared task with 100 dimensions. A dropout layer is added after pooling with the dropout rate fixed to be 0.5. The parameters are optimized using AdaDelta [20] with the decay rate 0.95 and constant 1e-6. Following [9], we set the dimension of the LSTM output as 360. We did not follow the settings of CNN in [9] since we found that using the CNN-arc by [13] can produce better results when adding biLSTM and CRF. We train the network using a complete end-to-end process. The model is validated on the development set and the optimal parameters are selected for testing. We implement the deep learning models with the help of Theano [21].

III. Experiments

A. Experiment Setup

Dataset We carried out the experiments on the SemEval-2015 cQA dataset [22]. The corpus consists of 3,229 questions: 2,600 for training, 300 for development, and 329 for testing. For each question, there is a thread of associated comments. The total number of comments is 20,162, with an average of 6.24 and a maximum of 143 per question. The class labels for the comments are distributed as follows: 9,941 *good* (49.31%), 8,208 *bad* (40.71%) and 2,013 *potential* (9.98%), in which *potential* is defined as potentially useful for the question. We didn't use external features other than plain text in our experiments.

Evaluation Matrices Following previous work, we report macro averaged $Precision$, $Recall$, F_1 and $Accuracy$ (*Acc.*) on the test set and also F_1 on individual categories. The official ranking is based on *macro* F_1.

Experiment Settings We compare our method against *JAIST* [8], *ICRC* [23] and *RCNN* [9]. JAIST and ICRC are the Top-1 and Top-2 systems in the shared task which employed statistical machine learning, and RCNN is one of the typical neural network based method. For binary classification, we compared with [24], [25], and [26]. They are all statistical machine learning methods in which [26] gets the state-of-the-art. We start by reporting the macro results and $\mathrm{F_1}$s on individual categories. Secondly, we compare the contributions of different components. Finally, we show some results on binary classification.

B. Results and Analysis

The macro average results on the testing set are shown in Table I with the optimal result for each column marked bold. We see that the best macro results and accuracies are all generated by the proposed architecture. The optimal $\mathrm{F_1}$ is achieved by A-ARC-III, which improves JAIST by 1.16% and

TABLE I
COMPARISONS ON MACRO AVERAGE RESULTS ON DIFFERENT MODELS.

Model	Precision	Recall	Macro F_1	Acc.
RCNN [9]	56.41	56.16	56.14	72.32
ICRC [23]	57.83	56.82	56.41	68.67
JAIST [8]	57.31	57.20	57.19	72.57
A-ARC-I	59.83	**58.41**	58.29	**76.42**
A-ARC-II	**60.12**	58.22	58.21	76.32
A-ARC-III	59.41	58.25	**58.35**	74.45

TABLE II
COMPARISONS ON INDIVIDUAL CATEGORIES.

Model	Good	Bad	Potential
RCNN [9]	77.31	75.88	15.22
ICRC [23]	76.52	74.32	**18.41**
JAIST [8]	78.96	78.24	14.36
A-ARC-I	81.22	79.60	14.05
A-ARC-II	**81.28**	79.06	14.29
A-ARC-III	79.34	**81.65**	13.77

TABLE III
IMPACT OF NETWORK MODULES.

Model	Precision	Recall	Macro F_1	Acc.
I†:LSTM+CRF	59.62	57.98	57.92	76.27
I:biLSTM	58.02	57.92	57.94	73.63
I:biLSTM+CRF*	58.01	58.25	58.11	72.52
I:biLSTM+CRF	59.83	**58.41**	58.29	76.42
II:LSTM+CRF	57.20	57.22	57.05	73.99
II:biLSTM	58.40	57.38	57.12	76.16
II:biLSTM+CRF*	57.58	57.35	57.26	75.51
II:biLSTM+CRF	**60.12**	58.22	58.21	76.32
III:CNN	57.95	56.34	55.75	75.76
III:CNN+CRF	58.66	57.32	56.44	76.62
III:LSTM+CRF	58.20	58.37	58.18	73.08
III:biLSTM	59.97	57.60	56.80	**77.18**
III:biLSTM+CRF*	58.20	58.37	57.69	71.91
III:biLSTM+CRF	59.41	58.25	**58.35**	74.45

† I,II,and III are short for A-ARC-I,II and III.
* models without attention.

RCNN by 2.21%. The maximum accuracy 76.42% is achieved by A-Arc-I, which significantly enhances the existed best by 3.85%. The results prove that the deep learning based methods have much stronger learning ability (compared with JAIST and ICRC) and the proposed global context modeling mechanism is superior than the existing deep context modeling method (R-CNN). The proposed variants have comparable results across all criteria, indicating the architecture can capture well the useful information given flexible input formats.

More details can be read from F_1s on individual classes (Table II). We find that all the three proposed variants are weaker than the competitors on *potential*. A possible explanation is that our models lack the ability of integrating numeric values as features (i.e. a similarity value) compared with statistical methods (i.e. ICRC) which may be helpful for identifying some special cases. We think it would be conductive to further improvements by adding external features (i.e. special words or similarities). Our methods have dramatic improvements on *good* and *bad*. We deem it is the context modeling mechanism that makes the deterministic information more complete and convincing when facing these two clear categories.

A statistical test (*t*-test with $\alpha = 0.05$) is further conducted to evaluate the significance of the improvement from the state-of-the-art method (JAIST). It is shown that the improvements by A-Arc-I and A-Arc-II are statistically significant ($p < 0.05$) and that by A-Arc-III shows a certain trend toward significance ($p = 0.12$). We believe some network configurations could be ulteriorly developed to produce better results.

The deep models gain profits from using LSTM, attention and CRF. LSTM learns correlations from the QA sequence along the timeline so as to model the cases of different types of dependencies. Attention is helpful in addressing information loss especially when the comment sequence is long (common in the dataset). CRF captures label dependency so that it can reinforce the constraints between the comment quality labels. Each component of the architecture plays a part among the three variants. We test their contributions by removing them from or adding them into the architecture. The comparisons are listed in Table III. It can be discovered that most configurations gain considerable improvement from the competitors' results in Table I and F_1 increases with the addition of more modules. The contribution of each module in each model distinguished. For example, the improvement by CRF is obvious in A-Arc-II (1.09%) and A-Arc-III (1.55%) but a bit mild in A-Arc-I (0.35%). We believe it is mainly due to the different input formats and the architectures that lead to disparate focuses during the optimization processes. And it is interesting to see that without attention, A-ARC-I still gains over 58% on F_1.

To test the necessity of context modeling in this task, we further remove the LSTM module (only for A-ARC-III based on the constraint of the input). The results are also shown in Table I by using *III:CNN* and *III:CNN+CRF*. It is noticed that the results drop markedly compared with those with LSTM, validating the effectiveness of context modeling.

Binary classification avoids the decision on *potential*. Better results were also generated based on context modeling. Differently, they added dependencies in their methods via syntactic and semantic features. The existing optimal F_1=81.5% is achieved by [26]. We get 82.22% by A-ARC-I and it outperforms the baseline by 0.72%. A-ARC-II (80.43%) and A-ARC-III(79.10%) also get competitive results compared with [25] (80.55%) and [24] (79.96%). However, different from them, we didn't rely on any complex feature engineering.

IV. CONCLUSION

This paper proposes an attentive deep neural network architecture with three variants to accomplish the answer selection task in cQA. Experiments on a benchmark dataset validate that by using attentive biLSTM and CRF for global context modeling, we achieve considerable improvements from the competitors, with the largest enhancement of 1.16% on macro averaged F_1 and 3.85% on classification accuracy. Based on the architecture, we also attain state-of-the-art results for binary classification. Future work will be focused on the exploring of effective features beyond word embedding.

REFERENCES

[1] L. Yu, K. M. Hermann, P. Blunsom, and S. Pulman, "Deep learning for answer sentence selection," *arXiv preprint arXiv:1412.1632*, 2014.

[2] C. Shah and J. Pomerantz, "Evaluating and predicting answer quality in community qa," in *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2010, pp. 411–418.

[3] A. Berger, R. Caruana, D. Cohn, D. Freitag, and V. Mittal, "Bridging the lexical chasm: statistical approaches to answer-finding," in *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2000, pp. 192–199.

[4] V. Punyakanok, D. Roth, and W.-t. Yih, "Mapping dependencies trees: An application to question answering," in *Proceedings of AI&Math 2004*, 2004, pp. 1–10.

[5] B. Wang, X. Wang, C. Sun, B. Liu, and L. Sun, "Modeling semantic relevance for question-answer pairs in web social communities," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2010, pp. 1230–1238.

[6] X. Qiu and X. Huang, "Convolutional neural tensor network architecture for community-based question answering," in *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*, 2015, pp. 1305–1311.

[7] Y. Shen, W. Rong, Z. Sun, Y. Ouyang, and Z. Xiong, "Question/answer matching for cqa system via combining lexical and sequential information," in *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[8] Q. H. Tran, V. Tran, T. Vu, M. Nguyen, and S. B. Pham, "Jaist: Combining multiple features for answer selection in community question answering," in *Proceedings of the 9th International Workshop on Semantic Evaluation, SemEval*, vol. 15, 2015, pp. 215–219.

[9] X. Zhou, B. Hu, Q. Chen, B. Tang, and X. Wang, "Answer sequence learning with neural networks for answer selection in community question answering," *arXiv preprint arXiv:1506.06490*, 2015.

[10] B. Hu, Z. Lu, H. Li, and Q. Chen, "Convolutional neural network architectures for matching natural language sentences," in *Advances in Neural Information Processing Systems*, 2014, pp. 2042–2050.

[11] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[12] Y. Xiang, X. Zhou, Q. Chen, Z. Zheng, X. Wang, and Y. Qin, "Incorporating label dependency for answer quality tagging in community question answering via cnn-lstm-crf," in *COLING*, 2016.

[13] Y. Kim, "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882*, 2014.

[14] A. Graves, "Generating sequences with recurrent neural networks," *arXiv preprint arXiv:1308.0850*, 2013.

[15] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *2013 IEEE international conference on acoustics, speech and signal processing*. IEEE, 2013, pp. 6645–6649.

[16] K. M. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom, "Teaching machines to read and comprehend," in *Advances in Neural Information Processing Systems*, 2015, pp. 1693–1701.

[17] Y. Lin, S. Shen, Z. Liu, H. Luan, and M. Sun, "Neural relation extraction with selective attention over instances."

[18] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proceedings of the eighteenth international conference on machine learning, ICML*, vol. 1, 2001, pp. 282–289.

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

[20] M. D. Zeiler, "Adadelta: an adaptive learning rate method," *arXiv preprint arXiv:1212.5701*, 2012.

[21] T. T. D. Team, R. Al-Rfou, G. Alain, A. Almahairi, C. Angermueller, D. Bahdanau, N. Ballas, F. Bastien, J. Bayer, A. Belikov *et al.*, "Theano: A python framework for fast computation of mathematical expressions," *arXiv preprint arXiv:1605.02688*, 2016.

[22] P. Nakov, L. Màrquez, W. Magdy, A. Moschitti, J. Glass, and B. Randeree, "Semeval-2015 task 3: Answer selection in community question answering," *SemEval-2015*, p. 269, 2015.

[23] Y. Hou, C. Tan, X. Wang, Y. Zhang, J. Xu, and Q. Chen, "Hitszicrc: Exploiting classification approach for answer selection in community question answering," in *Proceedings of the 9th International Workshop on Semantic Evaluation, SemEval*, vol. 15, 2015, pp. 196–202.

[24] A. Barrón-Cedeno, S. Filice, G. Da, S. Martino, S. Joty, L. M. Arquez, P. Nakov, and A. Moschitti, "Thread-level information for comment classification in community question answering," in *ACL*, 2015.

[25] S. Joty, A. Barrón-Cedeno, G. Da San Martino, S. Filice, L. Marquez, A. Moschitti, and P. Nakov, "Global thread-level inference for comment classification in community question answering," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP*, vol. 15, 2015.

[26] S. Joty, L. Màrquez, and P. Nakov, "Joint learning with global inference for comment classification in community question answering," in *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016.