A Self-attentive Similarity Based Approach for Community QA Ranking

Antonio Šajatović, Tome Radman, Lukrecija Puljić

University of Zagreb, Faculty of Electrical Engineering and Computing
Unska 3, 10000 Zagreb, Croatia
{antonio.sajatovic,tome.radman,lukrecija.puljic}@fer.hr

Abstract

One of the challenges in maintaining a community question answering service is ranking the answers and related questions by their relevance to the original question. We propose a neural network model which exploits sentence similarity in determining the ranks of given answers or related questions. The model consists of a sentence encoder and a self-attention mechanism applied on interaction features between the calculated encodings. Two different encoders are presented: a hierarchical convolutional neural network and a bidirectional gated recurrent unit (Bi-GRU) network with mean and max pooling. Training and evaluation was done on the dataset for Task 3 of SemEval-2017. Both approaches are shown to give promising results through experimental analysis.

1. Introduction

Online forums are by far one of the most practical ways for people to ask questions and get immediate answers from their peers. The amount of data that is accumulating on such online sites is constantly growing and already presents a valuable data source for various purposes. The main problem with this source of data is that it is highly unstructured, which means it is not straightforward to find out whether this particular question has already been asked or if there is a similar question already and to extract relevant information from available answers.

In this paper, we describe a system to automatically find relevant content from online forum. The system is designed to solve two tasks as proposed on the SemEval 2017 competition. The first task, called subtask A, is *Question-Comment Similarity*. The system is given a question and the first ten answers in its thread. The goal is to rank these answers according to their relevance to the given question. For the ssubtask B the system is given a question and the first ten related question from the forum, as retrieved by a search engine. The goal is to rank these question according to their relevance to the given question. This is *Question-Question Similarity* task.

We experimented with GRU (Cho et al., 2014) and CNN encoders in the modified conventional architecture for training NLI data (Bowman et al., 2015). Learning to rank was approximated via a pointwise approach, i.e. learning a binary classifier that can tell if an answer is relevant or non-relevant for a question in subtask A, and if two questions are similar or not in subtask B.

The rest of the paper is organized as follows: section 2 describes relevant work, section 3 explains systems architecture, section 4 describes the datasets and section 5 presents the experiments and their results.

2. Related work

The motivation behind SemEval 2017 Task3 is to automate the process of finding good answers to new questions in a Community Question Answering (CQA) discussion forum. To tackle this problem, the researchers proposed the two previously mentioned tasks.

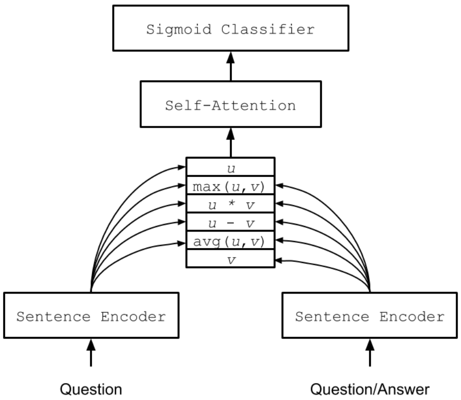

Figure 1: Overall system architecture.

In the past, *Question-Question Similarity* was tackled using various approaches like using question topic and focus (Duan et al., 2008) or using LDA topic language model that matches the questions both at the term and topic level (Zhang et al., 2014). Syntactic structure features were widely used for some of the top systems that participated in SemEval-2016 Task 3 (Nakov et al., 2017).

The most common approach for *Question-Answer Similarity* was to match the syntactic structures of the question and the candidate answer, particularly in the top systems that participated in SemEval-2016 Task 3 (Nakov et al., 2017). A similar approach was used in (Filice et al., 2017), one of the top systems on both subtasks.

Our approach differs from the previous ones in several ways: we use the same neural network model for both subtasks and we hypothesize that slightly modified architecture for NLI and neural network classifiers used in (Conneau et al., 2017) are beneficial for our task, especially given the pointwise approach. As far as we are aware, our model is the first one that uses a self-attention mechanism as described in (Shen et al., 2017).

3. Architecture

The overall system architecture is shown in Figure 1. The inputs are either a question/answer embeddings (subtask A) or question/question embeddings (subtask B). Shared weight encoders output a representation for the question u and the answer v. Then, six matching methods are applied to extract the relations between u and v :

- question encoding u,
- element-wise maximum $max(u, v)$,
- element-wise average $avg(u, v)$,
- element-wise product $u * v$,
- element-wise difference $u - v$
- question/answer encoding v

These six vectors representations are then fed into the soft-attention layer. The resulting vector is fed into a binary logistic regression classifier. As mentioned in the previous section, we hypothesize that multiple matching methods are better able to capture the relevant information for similarity between the inputs and we hope that the model can learn the importance of each via the self-attention mechanism.

The sentence encoders we employ are Bidirectional GRU and Hierarchical ConvNet with max or mean pooling, as described in (Conneau et al., 2017). All three achieve near state-of-the-art performance on the STS14 - Semantic Textual Similarity (Agirre et al., 2014) and SICK-R (Marelli et al., 2014) datasets, which are closely related to both subtasks A and B. Detailed model descriptions that we followed in our implementations are available in (Conneau et al., 2017). In the following subsections, each model is shortly described.

3.1. Hierarchical ConvNet

Hierarchical ConvNet is a streamlined version of the AdaSent (Zhao et al., 2015) convolutional architecture consisting of only 4 convolutional layers. The model captures hierarchical abstractions of an input sentence in a fixed-size representation by computing a mean-pooling or max-pooling operation over the hierarchical feature maps. The final representation $u = [u_1, u_2, u_3, u_4]$ concatenates representations at different levels of the input sentence, as shown in Figure 2.

3.2. BiGRU with mean and max pooling

A common approach in encoding sequences is using recurrent neural networks. We opted for using a bidirectional RNN with gated recurrent unit (GRU) cells (Cho et al., 2014). The model outline can be seen on Figure 4. For each word in the output sequence, a bidirectional GRU outputs a vector h_t, which is a concatenation of backward and forward GRU hidden states:

$$\overrightarrow{h_t} = \overrightarrow{GRU}_t(w_1, ..., w_T) \tag{1}$$

$$\overleftarrow{h_t} = \overleftarrow{GRU}_t(w_1, ..., w_T) \tag{2}$$

$$h_t = \left[\overrightarrow{h_j}, \overleftarrow{h_j}\right] \tag{3}$$

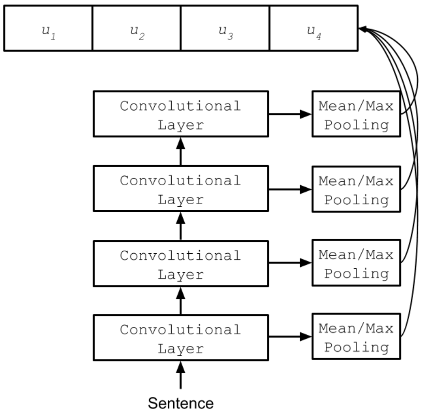

Figure 2: Hierarchical ConvNet encoder.

Instead of only using the output from the last timestep, vectors $(h1, ..., h_T)$ are aggregated using mean or max pooling: each dimension of the final vector contains the average or maximum value of that dimension from the Bi-GRU outputs at each timestep. This enables the model to utilize information from different timesteps more successfully.

Our final encoder uses two Bi-GRU networks, with outputs of the first network being fed as inputs into the second network. The outputs from both networks are passed through a pooling layer. The final encoding is a concatenation of the pooled outputs from both networks (Figure 3).

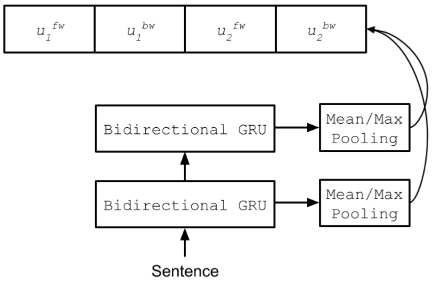

Figure 3: Bidirectional GRU encoder.

3.3. Self-Attention

Self-attention, also called intra-attention, has been used successfully in a variety of tasks, with the most recent success in neural machine translation (Vaswani et al., 2017). Inspired by such success, we have implemented the *"source2token"* multidimensional self-attention mechanism as defined in (Shen et al., 2017). Source2token self-attention explores the dependency between x_i and the entire sequence of vectors $\boldsymbol{x}$, compressing $\boldsymbol{x}$ (in our case, the six representation vectors) into a single vector:

$$z_i = f(x_i) = W^{(2)} ELU(W^{(1)} x_i). \tag{4}$$

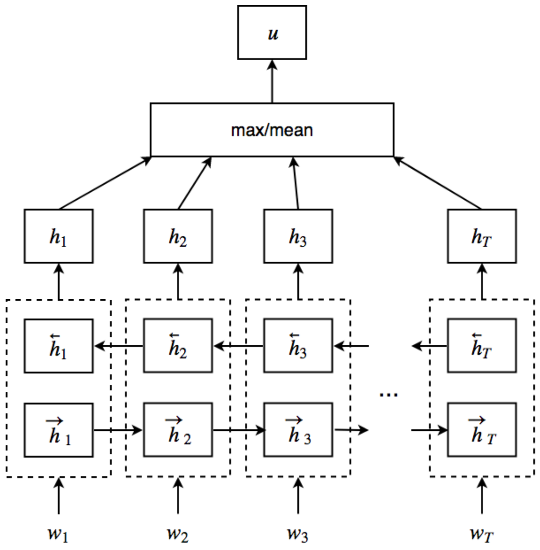

Figure 4: Detailed overview of the Bi-GRU network with mean/max pooling used as a part of the encoder.

Attention weights are calculated using the following equation:

$$\alpha_i = p(z = i|\boldsymbol{x}) = softmax(z_i). \quad (5)$$

The output of source2token self-attention for $\boldsymbol{x}$ is the context vector c:

$$c = \sum_{i=1}^{n} \alpha_i \odot x_i. \quad (6)$$

4. Dataset

We used the official data from SemEval 2017 competition, which is publicly available. The data for both subtasks were gathered from the Quatar Living forum. In the subtask A, for each of the forum topics, top 10 answers were taken and manually annotated as being ”Good”, ”PotentiallyUseful” or ”Bad” regarding the thread question being asked. For the subtask B, the dataset consisted of questions and the related thread questions which were annotated as being either a ”PerfectMatch”, ”Relevant” or ”Irrelevant” with respect to the original one. From subtask A’s labels, ”Bad” and ”PotentiallyUseful” were both taken as being bad during evaluation, while for subtask B, both ”PerfectMatch” and ”Relevant” were considered as good. Statistics about the dataset are shown in Table 1.

Category	Train	Dev	Test
Relevant Answers	1900	2440	3270
Good	6651	818	1329
Potentially Useful	3110	413	456
Bad	8139	1209	1485
Related Question	2669	500	700
TAR-2017-P Perfect Match	235	59	81
Relevant	848	155	152
Irrelevant	1586	286	467

Table 1: Dataset statistics.

5. Experiments

In this section we first describe the training setup in detail and then dicuss the obtained results.

5.1. Training setup

We initialize the word embeddings by GloVe 6B pre-trained vectors (Pennington et al., 2014). The Out-of-Vocabulary words in training set are initialized to an all-zero vector. The models were trained for 30 epochs as we observed that training any longer would lead to overfitting, even when using Dropout (Srivastava et al., 2014). On subtask A we used the Adadelta optimizer (Zeiler, 2012), while on subtask B we used the Adam optimizer (Kingma and Ba, 2014), due to convergence, both with default settings. Batch size was set to 32 in both subtasks. In order to find the optimal hyperparameter values, we tested several values from an arbitrarily chosen range for each hyperparameter and chose those that minimized the network loss on the subtask A development dataset. The optimal values are shown in Table 2. Grid search method was not feasible in our case due to a large parameter search space. All three models are implemented using Keras (Chollet and others, 2015) and if not specified, all other hyperparameters were set to default Keras values. The models were run on the Google Colaboratory platform[1] with GPU runtime enabled.

Hyperparameter	Value
Dropout rate	0.25
BiGRU hidden size	300
Sentence timestep	100
CNN filter size	300
CNN kernel size	3
FC layer size	300
Activation function	ELU
Word embedding size	300
Pooling method	max
Loss function	binary cross-entropy

Table 2: Model hyperparameters.

5.2. Results

Tables 3 and 4 show the performances of all presented models for subtask A and B, respectively. The organizers chose *mean average precision* (MAP) as the official metric for this task.

In subtask A, all of our systems managed to outperform the baseline by a wide margin, with *bigru_maxpool* being the most successful, achieving a MAP score of 82.60. By conducting a permutation test, we were able to show that the difference between *bigru_maxpool* and all the convolutional models is statistically significant, as shown in Table 5.

Our models failed to outperform the IR baseline for subtask B. This could be attributed to our decision to do model selection only on subtask A and reuse the hyperparameters on subtask B, in an attempt to create a more general model

[1] https://colab.research.google.com

applicable to various sentence similarity tasks. The best-performing model was *cnn_maxpool* (MAP 33.68). Statistical significance results are shown in Table 6.

System	MAP	Acc	F1
bigru_avgpool	80.84	67.58	62.00
bigru_maxpool	**82.60**	65.43	54.80
cnn_avgpool	79.82	65.29	56.70
cnn_maxpool	80.18	65.56	58.22
KeLP	**88.43**	73.89	69.87
Beihang-MSRA	88.24	51.98	68.40
SwissAlps	86.24	61.30	43.30
Baseline 1 (IR)	72.61	-	-
Baseline 2 (random)	62.30	52.70	62.54
Baseline 5 (TF-IDF)	0.43	0.63	0.49

Table 3: Subtask A results.

System	MAP	Acc	F1
bigru_avgpool	29.22	63.52	31.26
bigru_maxpool	33.36	60.00	33.33
cnn_avgpool	30.68	61.14	30.20
cnn_maxpool	**33.68**	52.27	34.98
SimBow	**47.22**	52.39	42.37
LearningToQuestion	46.93	18.52	31.26
KeLP	46.66	69.20	50.64
Baseline 1 (IR)	41.85	-	-
Baseline 2 (random)	29.81	34.77	30.00
Baseline 5 (TF-IDF)	0.43	0.40	0.57

Table 4: Subtask B results.

model	bigru_maxpool	cnn_avgpool	cnn_maxpool	Baseline IR	Baseline TF-IDF	
bigru_avgpool		=	=	=	*	*
bigru_maxpool			*	*	*	*
cnn_avgpool				=	*	*
cnn_maxpool					*	*

* Statistically significant, $p \leq 0.05$
$^=$ Not significant, $p > 0.05$.

Table 5: Significance of MAP differences between system pairs for subtask A.

6. Conclusion

We presented a deep neural network architecture for relevancy ranking in community question answering services, using two kinds of sentence encoders: a hierarchical convolutional neural network and a bidirectional GRU with mean/max pooling. A self-attention mechanism was used to assign importance to different interaction features. The models were evaluated on SemEval-2017 task 3 dataset, achieving a MAP score of 82.60 on subtask A and 33.68 on subtask B.

model	bigru_avgpool	bigru_maxpool	cnn_avgpool	cnn_maxpool	Baseline IR	Baseline TF-IDF
bigru_avgpool		=	=	=	*	*
bigru_maxpool			=	=	*	*
cnn_avgpool				=	*	*
cnn_maxpool					*	*

* Statistically significant, $p \leq 0.05$
$^=$ Not significant, $p > 0.05$.

Table 6: Significance of MAP differences between system pairs for subtask B.

Our goal was to create a general system reusable in different sentence similarity scenarios. Hence, we only used the dataset for subtask A for model selection, which resulted in a drop in performance on subtask B.

For future improvements, more focus should be given to finding different ways to combine the encoded sentences and it would also be worth trying a character-based model.

Acknowledgements

We thank dr.sc. Mladen Karan[2] from TakeLab for providing us with the adequate statistical significance tests.

References

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2014. Semeval-2014 task 10: Multilingual semantic textual similarity. In *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)*, pages 81–91.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

KyungHyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *CoRR*, abs/1409.1259.

François Chollet et al. 2015. Keras. `https://keras.io`.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*.

Huizhong Duan, Yunbo Cao, Chin-Yew Lin, and Yong Yu. 2008. Searching questions by identifying question topic and question focus. *Proceedings of ACL-08: HLT*, pages 156–164.

Simone Filice, Giovanni Da San Martino, and Alessandro Moschitti. 2017. Kelp at semeval-2017 task 3: Learning pairwise patterns in community question answering. In

[2] `http://takelab.fer.hr/mladen/`

Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), pages 326–333.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, Roberto Zamparelli, et al. 2014. A sick cure for the evaluation of compositional distributional semantic models. In *LREC*, pages 216–223.

Preslav Nakov, Doris Hoogeveen, Lluís Màrquez, Alessandro Moschitti, Hamdy Mubarak, Timothy Baldwin, and Karin Verspoor. 2017. Semeval-2017 task 3: Community question answering. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 27–48.

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.

Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, Shirui Pan, and Chengqi Zhang. 2017. Disan: Directional self-attention network for rnn/cnn-free language understanding. *arXiv preprint arXiv:1709.04696*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010.

Matthew D Zeiler. 2012. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.

Kai Zhang, Wei Wu, Haocheng Wu, Zhoujun Li, and Ming Zhou. 2014. Question retrieval with high quality answers in community question answering. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 371–380. ACM.

Han Zhao, Zhengdong Lu, and Pascal Poupart. 2015. Self-adaptive hierarchical sentence model. In *IJCAI*, pages 4069–4076.