Cross Attention for Selection-based Question Answering

Alessio Gravina, Federico Rossetto, Silvia Severini and Giuseppe Attardi

Dipartimento di Informatica
Università di Pisa
gravina.alessio@gmail.com, fedingo@gmail.com, sissisev@gmail.com,
attardi@di.unipi.it

Abstract. Answer Sentence Selection (ASS) is one of the steps typically involved in Question Answering, a hard task for natural language processing since full solutions would require both natural language understanding and world knowledge. We present a new approach to tackle ASS, based on a Cross-Attentive Convolutional Neural Network. The approach was designed for competing in the *Fujitsu AI-NLP* challenge Fujitsu [4], which evaluates systems on their performance on the *SelQA*[7] dataset. This dataset was created on purpose as a benchmark to stress the ability of systems to go beyond simple word co-occurrence criteria. Our submission achieved the top score in the challenge.

1 Introduction

Typical approaches to Question Answering involve primarily the following steps: *question analysis*, which determines what to look for; *candidate extraction*, which exploits Information Retrieval (IR) techniques to search through documents for candidate answers; *answer selection*, which prunes the set of candidates and *answer extraction* which extracts the correct answer from the selected sentences.

Given large enough document collections, IR techniques are often capable of providing satisfactory results for both candidate extraction and answer selection: however relying on simple keyword matching is not sufficient when question and answer do not match closely enough, e.g. the question is phrased in different terms from those present in the document containing the answer. More sophisticated techniques have been proposed, for example query rewriting or query expansion [6, 8, 18], involving for example dictionaries of synonyms or word embeddings [9], or using topic modeling to identify a shared latent topic between question and answer [19]. These approaches fail though when deeper knowledge is required, for example world knowledge or inference from given facts.

Answer Sentence Selection is an important sub-task of Question Answering, that aims at selecting the correct answers to a given question among a set of candidate sentences. Answer extraction involves Natural Language Processing techniques for interpreting candidate answer sentences and establishing how they relate to questions.

More sophisticated methods of ASS that go beyond IR approaches involve for example tree edit models [5] and semantic distances based on word embeddings [15].

Recently, Deep Neural Networks have also been applied to this task [11], providing performance improvements with respect to previous techniques. The most common approaches exploit either *recurrent* or *convolutional* neural networks. These models are good at capturing contextual information from sentences, making them a nice fit for the problem of answer sentence selection.

The improvements in the state of the art on ASS over the years are listed in [1], with the current top score of 0.863 MRR [2] on the TREC QA dataset reported by Tayyar Madabushi et al. [14].

Research on this problem has benefited in the last few years from the development of specific datasets for training systems on this task, like *SelQA* [7]. This dataset is notable for its larger size, that reaches more that 60.000 sentence-question pairs. This allows for the creation of deeper and more complex models, that do not risk much to over-fit. Moreover the *SelQA* dataset has been specifically crafted in order to make it harder to handle by systems based on purely Information Retrieval techniques that rely on word co-occurrences. All questions were paraphrased using different terms, in order to ensure that solutions would involve more sophisticated techniques such as reading comprehension capabilities.

In this paper we present a new model for the task of answer sentence selection that improves the current state of the art performances. The model relies on a *Convolutional Neural Network* with a double mechanism of attention between question and answer. The model is inspired by the light attentive mechanism proposed by Yin and Sch utze [17], which we improve and apply in both directions to question and answer pairs.

In the sections below we first survey the more relevant literature, highlighting the context of the Question Answering in which our model fits. After that, we explain the model architecture and the results achieved in our experiments, on the *SelQA* dataset.

2 Related work

Deep learning (DL) approaches have been exploited for the task of answer selections and significantly outperforming traditional method. Attention-based mechanisms have shown very promising results on a variety of NLP task and have been recently proposed also for the answer selection task. In particular we mention the approaches based on either Convolutional Neural Networks (CNN) or Long Short-Term Memory (LSTM) networks, with various types of attention mechanisms, like for example the attentive pooling network by dos Santos et al. [12] and the LSTM-based models for non-factoid answer selection by Tan et al. [13].

dos Santos et al. [12] introduce the mechanism of attentive pooling that enables the pooling layer to be aware of the current question/answer pair, so

that information from the two items influences the computation of each other's representation. This enables joint learning of both the representations of the input pairs as well as a measure of their similarity. An attention vector is created, which guides the subsequent pooling. This model has the ability of embedding two inputs, not semantically comparable, into a common representation space, of working with input pairs of different length and the independence from the underlying representation learning like CNN or RNN. Attentive pooling can be effectively used with CNNs (*AP-CNN*) and biLSTM (*AP-biLSTM*) in the context of the answer selection task, achieving the best reported results on the *WikiQA* dataset.

Tan et al. [13] present four Deep Learning models for answer selection based on biLSTM and CNN, with different complexities and capabilities. The basic model, called QA-LSTM, implements two similar flows, one for the question and one for the answer. In general, a biLSTM creates a representation of the question/answer that is processed by a max or average pooling layer. The two flows are then merged with a cosine similarity matching that expresses how close question and answer are. More complexity is obtained with QA-LSTM/CNN, a model similar to the previous one but, instead of the pooling layer, exploits a more complex CNN. The output of biLSTM is sent to a convolution filter, in order to give a more complete representation of questions and answers. This filter is followed by 1-max pooling layer and a fully connected layer. Finally, the paper presents the most complex models, QA-LSTM with attention and QA-LSTM/CNN with attention, that extend the previous models with the addition of a simple attention mechanism between question and answer, which aims to better identify the best candidate answer to the question. The mechanism consists in multiplying the biLSTM hidden units of the answers with the output computed from the question pooling layer. These models are tested on the *InsuranceQA* [3] and *TREC-QA* [16] datasets, achieving quite good performances.

Wang et al. [15] propose an approach to answer selection that takes into account similarities and dissimilarities between sentences by decomposing and composing lexical semantics over sentences. In particular they represent each word as a vector and calculate a semantic matching vector for each word based on all words in the other sentence. Then each word vector is decomposed into a similar and a dissimilar component, based on the semantic matching vector. A CNN is then used to capture features by composing these parts and estimating a similarity score over the composed feature vectors to predict which sentence is the answer to the question.

The most influential work for our approach is the one on attentive convolution by Yin and Sch utze [17]. The authors of the paper apply an attention mechanism not only to the pooling operation like in dos Santos et al. [12], but also to the convolutional layer itself. They present in fact two different models: a simpler mechanism called light attentive ConvNet, and a more complex one where they split the attention computations and the convolution itself. This type of models are quite effective at comparing a text with a reference text, and are tested in many different applications, like Textual Entailment, Answer Sentence Selection

and Text Classification. In all the tested tasks they achieved state-of-the-art performances, overcoming previous model applied to those tasks.

We now describe in more detail the light attentive ConvNets, since they are the foundation of the model that we will present in the next section.

2.1 Light Attentive ConvNets

The aim of the model presented by Yin and Sch utze [17] is to compute a representation for the main sentence in a way that convolution filters encode not only local context, but also an attentive context over the reference sentence.

The first Attentive Convolution layer generates the *Attentive Context Vector*. To do this, an energy function is used to determine how much each hidden state in the sentence is relevant to the current hidden state of the question. Then the average of the hidden state of the sentence is computed, weighted by the matching score, in order to obtain the attentive context for the current question's hidden state.

After this layer, there is an *Attentive Convolution* layer. This layer performs first a standard convolution without attention over the window (h_{i-1}, h_i, h_{i+1}), where h_i is the i-th hidden state of the question. Secondly, there is a convolution using the attentive context. The final results are added element-wise, a bias term is added and a non-linear activation function is applied. The output of the i-th hidden layer of the *(n+1)*-th layer is:

$$h_i^{n+1} = tanh(W^1 \cdot [h_{i-1}^n, h_i^n, h_{i+1}^n] + W^2 \cdot c_i^n + b) \tag{1}$$

where $W^1 \in \mathbb{R}^{d \times 3d}$ and $W^2 \in \mathbb{R}^{d \times d}$ are weights, $b \in \mathbb{R}^d$ is the bias and c_i^n is the i-th attentive context of the n-th layer.

3 Model description

In this section we describe the architecture of our model, as illustrated in Figure 1, while the detailed design of the *cross-attentive layer* can be found in Figure 2.

3.1 Network Architecture

In this section we describe in more detail the overall network architecture. At first, the inputs are transformed using an embedding layer initialized using the *GloVe* word embeddings. Then we apply a number of stacked layers of Cross-attentive convolutions. We will explain below in section 3.2 how they are built. Then, for each layer, we apply a global max pooling to extract both a question and a sentence representation for that layer.

These representations are then concatenated together to obtain two vectors Q and S that represent each question/answer sentence pair. These vectors are then both added and multiplied and the results concatenated before being fed to a simple Feed-Forward Neural Network.

Finally, we use the predicted distribution from the previous network and we augment it with additional information, in order to feed a Logistic Regression layer. Based on our experiments we found it useful to apply some Information Retrieval techniques for the final classifier, like the *tf-idf* or the *number of co-occurrent words*.

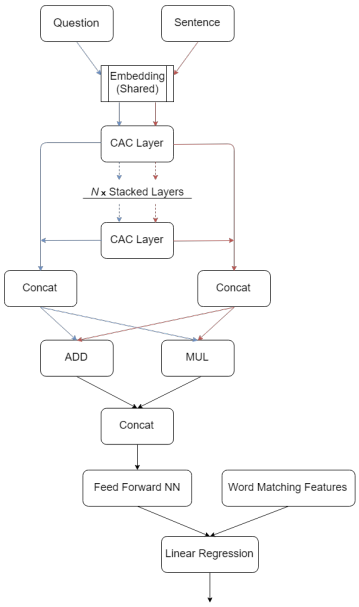

Fig. 1: Cross-attentive Convolutional Network

3.2 Cross-Attentive Convolutional Layer

This section describes the Cross-Attentive Convolutional Layer, as shown in Figure 2.

Our solution is derived from the Attentive Convolution layer presented in [17]. We use the light attentive mechanism described there in both directions, between the question and the candidate answer. The basic idea is to use a function f that creates a similarity matrix for the two sentence representations that we are convolving.

$$f : Q, S \rightarrow A, \mathbb{R}^{l \times e} \times \mathbb{R}^{l \times e} \rightarrow \mathbb{R}^{l \times l}$$

where l is the fixed (padded) length of each sentence, and e is the dimension of the embeddings that we are using.

After generating this A similarity matrix, we apply a *softmax* function to normalize the columns c_i, and rows r_j. We then use these weighting vectors to create the Attentive dense layer for each of the two sentences considered. The context is transformed by this Dense Layer and added to the base convolution.

As a final operation we route the results in two directions. For the next layer, we use a Max-Pooling operator, with a default 2 by 2 window. For the other direction instaed, we apply a global pooling to extract a Layer-Representation of the sentence. This representation throughout the layers are finally concatenated as described before.

4 Experimental Results

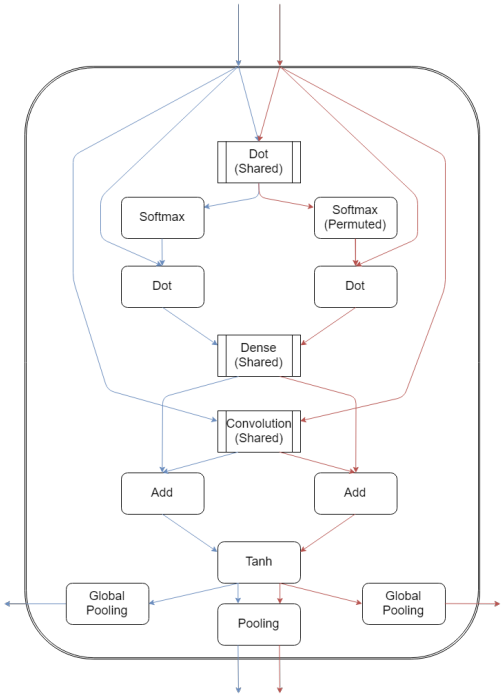

Fig. 2: Cross-attentive Convolutional Network

layer;

We investigate the performance of our model on the interesting dataset *SelQA*.
The dataset needs a preprocessing phase before feeding its data to the model. As evaluation metrics of the performance, we use the Mean Reciprocal Rank MRR as discussed below. After showing the results in terms of the MRR on our model, we compare them with the state-of-the-art outcomes and finally we include some discussion with an error analysis.

4.1 Data preprocessing

Before feeding the model with the data, some preprocessing of the sentences is needed. This operation consists of the following steps:

1. removal of non-ASCII characters, to improve the coverage of words present in the *GloVe* [10] embeddings, that are used to initialize the embeddings
2. replacement of digits with the 0 character, to reduce the number of tokens denoting numbers since, for the task at hand, they have similar meaning;
3. removal of punctuation since it doesn't have much relevance and we are dealing with single sentences. This improves the tokenization process;

4. normalize the length of sentences to a fixed length of 50 by adding padding, in order to feed them to our Neural Network.

The sentence representation consists of the list of word embedddings of its tokens. Each sentence is tokenized using the Keras Tokenizer API and the vector of each token is looked up in the *GloVe* word embeddings: if no exact match is found, the closest token within an edit distance of 2 is used, if present, otherwise the word embedding for the *unknown_token* is used.

4.2 Evaluation Metrics

As evaluation metrics to measure the accuracy of our model, we used *Mean Reciprocal Rank (MRR)* [2], a statistical measure for evaluating a process that produces a ranked list of possible responses to each query in a test sample. The mean reciprocal rank is the average of the reciprocal of the rank of the first correct result for each of a sample of queries Q:

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_i}$$

where $rank_i$ refers to the rank position of the first correct result for the i-th query.

4.3 SelQA dataset

As an experiment, we tested our model on the *SelQA* dataset [7].

The dataset introduces a corpus annotation scheme that enhances the generation of large, diverse, and challenging datasets by explicitly aiming to reduce word co-occurrences between the question and answers. The *SelQA* dataset consists of questions generated through crowd-sourcing and long sentence answers drawn from the ten most prevalent topics in the English Wikipedia. A total of 486 articles are uniformly sampled from the topics of: Arts, Country, Food, Historical Events, Movies, Music, Science, Sports, Travel, TV. After that, the original data is preprocessed into smaller chunks, resulting in 8,481 sections, 113,709 sentences and 2,810,228 tokens.

For each section, a question that can be answered in that same section by one or more sentences was generated by human annotators. The corresponding sentence or sentences that answer the question were selected. As an additional noise process, annotators were also asked to create another set of questions from the same selected sections excluding the original sentences selected as answers in previous task. Then all questions were paraphrased using different terms, in order to make sure the QA algorithm would be evaluated by reading comprehension

Table 1: SelQA results

	Validation MRR	**Test MRR**
Cross-Attentive CNN	91.37 %	90.61 %
CNN SelQA	86.67 %	85.68 %
RNN SelQA	88.25 %	87.59 %

rather than by the ability to count word co-occurrences. Lastly if ambiguous questions were found, they were rephrased again by a human annotator.

Table 1 shows the results obtained by testing our model and by the two models proposed by Jurczyk et al. [7]. The result on the Test set outperform the other two models by more than 3%.

5 Conclusions

The aim of this work was to try to improve the state of the art results for the task of Answer Sentence Selection using an architecture based on Convolutional Neural Networks. We implemented a Cross-attentive CNN and we tested it on *SelQA* dataset. The experiments show that our model is able to beat the current state of the art on the *SelQA* dataset. More precisely, we were able to achieve 90% of MRR on the test set. We think that this is due to two main factors. First, the dataset is fairly new and it has not been deeply experimented. Second, our model has a deep structure and a big amount of parameters. This means that with more data, the model performance might further improve.

The strong points of our model are simplicity and the ability to generalize. The simplicity is shown for example from the speed of the training phase that took only two hours on a 24 core machine without any GPU acceleration.

An interesting future development would be to test our model on other datasets reported in the literature, in order to obtain a more direct comparison with our results.

6 Acknowledgments

The experiments were carried on a Dell server with 4 Nvidia GPUs Tesla P100, partly funded by the University of Pisa under grant Grandi Attrezzature 2016.

We thank Fujitsu for organizing the challenge and giving us the opportunity to participate in a stimulating experiment.

Bibliography

[1] ACL: Question answering (state of the art). https://aclweb.org/aclwiki/Question_Answering_(State_of_the_art) (2018), accessed: 2018-10-30

[2] Craswell, N.: Mean Reciprocal Rank. In: Liu, L., Özsu, M.T. (eds.) Encyclopedia of Database Systems. Springer US, Boston, MA (2009), https://doi.org/10.1007/978-0-387-39940-9_488

[3] Feng, M., Xiang, B., Glass, M.R., Wang, L., Zhou, B.: Applying deep learning to answer selection: A study and an open task. arXiv preprint arXiv:1508.01585 (2015)

[4] Fujitsu: Fujitsu AI-NLP Challenge. https://openinnovationgateway.com/ai-nlp-challenge/challenge.php (2018), accessed: 2018-05-30

[5] Heilman, M., Smith, N.A.: Tree edit models for recognizing textual entailments, paraphrases, and answers to questions. In: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, HLT 10. pp. 1011–1019. Association for Computational Linguistics (2010)

[6] Jeon, J., Croft, W.B., Lee, J.H.: Finding similar questions in large question and answer archives. In: Proc. of the 14th ACM International Conference on Information and Knowledge Management (CIKM2005). pp. 84–90. ACL Association for Computational Linguistics (2005)

[7] Jurczyk, T., Zhai, M., Choi, J.D.: SelQA: A New Benchmark for Selection-based Question Answering. pp. 820–827 (2016)

[8] Komiya, K., Abe, Y., Morita, H., Kotani, Y.: Question answering system using q&a site corpus query expansion and answer candidate evaluation. Springerplus 396(2), 1–11 (2013)

[9] Kuzi, S., Shtok, A., Kurland, O.: Query expansion using word embeddings. In: Proc. of the 25th ACM International on Conference on Information and Knowledge Management (CIKM2016). pp. 1929–1932. ACM (2016)

[10] Pennington, J., Socher, R., Manning, C.D.: GloVe: Global Vectors for Word Representation. In: Empirical Methods in Natural Language Processing (EMNLP). pp. 1532–1543 (2014), http://www.aclweb.org/anthology/D14-1162

[11] Rao, J., He, H., Lin, J.: Noise-contrastive estimation for answer selection with deep neural networks. In: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management (CIKM 16). pp. 1913–1916. ACM (2016)

[12] dos Santos, C.N., Tan, M., Xiang, B., Zhou, B.: Attentive pooling networks. CoRR (2016)

[13] Tan, M., Xiang, B., Zhou, B.: Lstm-based deep learning models for non-factoid answer selection. CoRR abs/1511.04108 (2015), http://arxiv.org/abs/1511.04108

[14] Tayyar Madabushi, H., Lee, M., Barnden, J.: Integrating question classification and deep learning for improved answer selection. In: Proceed-

ings of the 27th International Conference on Computational Linguistics. pp. 3283–3294. Association for Computational Linguistics (2018), http://aclweb.org/anthology/C18-1278
[15] Wang, Z., Mi, H., Ittycheriah, A.: Sentence similarity learning by lexical decomposition and composition. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers. pp. 1340–1349. The COLING 2016 Organizing Committee (2016), http://www.aclweb.org/anthology/C16-1127
[16] Yao, X., Durme, B.V., Callison-Burch, C., Clark, P.: Answer extraction as sequence tagging with tree edit distance. In: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 858–867 (2013)
[17] Yin, W., Sch utze, H.: Attentive Convolution. CoRR (2017)
[18] Yu, Z.T., Zheng, Z.Y., Tang, S.P., Guo, J.Y.: Query expansion for answer document retrieval in chinese question answering system. In: Proc. of 2005 International Conference on Machine Learning and Cybernetics. pp. 72–77 (2005)
[19] Zhang, K., Wu, W., Wu, H., Li, Z., Zhou, M.: Question retrieval with high quality answers in community question answering. pp. 371–380 (2014)