A Comparative Study of Models for Answer Sentence Selection

Alessio Gravina
Università di Pisa
gravina.alessio@gmail.com

Federico Rossetto
Università di Pisa
fedingo@gmail.com

Silvia Severini
Università di Pisa
sissisev@gmail.com

Giuseppe Attardi
Università di Pisa
attardi@di.unipi.it

Abstract

Answer Sentence Selection is one of the steps typically involved in Question Answering. Question Answering is considered a hard task for natural language processing systems, since full solutions would require both natural language understanding and inference abilities. In this paper, we explore how the state of the art in answer selection has improved recently, comparing two of the best proposed models for tackling the problem: the Cross-attentive Convolutional Network and the BERT model. The experiments are carried out on two datasets, WikiQA and SelQA, both created for and used in open-domain question answering challenges. We also report on cross domain experiments with the two datasets.

1 Introduction

Answer Sentence Selection is an important subtask of Question Answering, that aims at selecting the sentence containing the correct answer to a given question among a set of candidate sentences. Table 1 shows an example of a question and a list of its candidate answers, taken from the *SelQA* dataset (Jurczyk et al., 2016). The last column contains a binary value, representing whether the sentence contains the answer or not.

Answer extraction involves natural language processing techniques for interpreting candidate sentences and establishing whether they relate to questions and contain an answer. More sophisticated methods of Answer Sentence Selection that go beyond Information Retrieval approaches involve for example tree edit models (Heilman and Smith, 2010) and semantic distances based on word embeddings (Wang et al., 2016).

Recently, Deep Neural Networks have also been applied to this task (Rao et al., 2016), 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.

Research on this problem has benefited in the last few years by the development of better datasets for training systems on this task. These datasets include *WikiQA* (Yang et al., 2015) and *SelQA* (Jurczyk et al., 2016). The latter 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, with less risk of overfit.

The state of the art model on the *SelQA* dataset (Jurczyk et al., 2016), up to 2018, was *Cross-attentive Convolutional Network* (Gravina et al., 2018), with a score of 0.906 MRR (Craswell, 2009).

In this paper we present further experiments with the *Cross-attentive Convolutional Network* model as well as experiments that exploit the BERT language model by Devlin et al. (2018).

In the following sections we survey relevant literature on the topic, we describe the datasets used in our experiments and present the models tested in our experiments. Finally, we describe the experiments conducted with these models and report the results achieved.

2 Related work

We present a brief survey of the most recent approaches for answer selection in question answer-

All authors contributed equally to this manuscript.

Table 1: Sample question/candidate answers.

How much cholesterol is there in an ounce of bacon?	
One rasher of cooked streaky bacon contains 5.4g of fat, and 4.4g of protein.	0
Four pieces of bacon can also contain up to 800mg of sodium.	0
The fat and protein content varies depending on the cut and cooking method.	0
Each ounce of bacon contains 30mg of cholesterol.	1

ing.

Tan et al. (2015) present four Deep Learning models for answer selection based on biLSTM (bidirectional LSTM) and CNN (Convolutional Neural Network), with different complexities and capabilities. The basic model, called QA-LSTM, implements two similar flows, one for the question and one for the answer. The biLSTM builds a representation of the question/answer pair that is passed 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.

A more complex solution, called QA-LSTM/CNN, uses a similar model, which replaces the pooling layer with a 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* (Feng et al., 2015) and *TREC-QA* (Yao et al., 2013) datasets, achieving quite good performances.

The HyperQA (Tay et al., 2017) model uses a pairwise ranking objective to represent the relationship between question and answer embeddings in a hyperbolic space instead of an euclidean space. This empowers the model with a self-organizing ability and enables automatic discovery of latent hierarchies while learning embeddings of questions and answers.

Wang et al. (2016) present a model that takes into account similarities and dissimilarities between sentences by decomposing and composing lexical semantics over sentences. In particular the model represents each word as a vector and calculates 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. Afterwards, a CNN model is used to capture features by composing these parts and a similarity score is estimated over the composed feature vectors to predict which sentence is the answer to the question.

3 Models

We describe here the models used in our experiments.

3.1 Simple Logistic Regression Classifier

Jurczyk et al. (2016) state that the SelQA dataset was created through a process that tried to reduce the number of co-occurrent words, so that simple word matching methods would be less effective. To evaluate whether this aim was indeed achieved, we built a simple linear regression classifier using as features the sentence and question length, the number of co-occurrent words and the *idf* coefficients of the word co-occurrences.

3.2 Cross-attentive Convolutional Network

The Cross-attentive Convolutional Network (CACN) is a model designed for the task of Answer Sentence Selection and in 2018 had achieved state of the art performance (Gravina et al., 2018). The model relies on a *Convolutional Neural Network* with a double mechanism of attention between questions and answers. The model is inspired by the light attentive mechanism proposed by Yin and Schütze (2017), which it improves by applying it in both directions to question and answer pairs.

The CACN model achieved top score in the "Fujitsu AI NLP Challenge 2018" [1], that used the

[1] https://openinnovationgateway.com/ai-nlp-challenge/

SelQA dataset.

3.3 BERT language representation model

BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2018) is a language representation model. BERT usage involves two steps: *pre-training* and *fine-tuning*. During pre-training, the model is trained on a large collection of unlabeled text on a language modeling task. Fine-tuning BERT on a downstream task involves extending the model with additional layers tailored to the task, initializing the model with the pre-trained parameters, and then training the extended model with labeled data from the task. The extended model might consist just of a single output layer. Such models have been shown capable to achieve state-of-the-art accuracy for a wide range of tasks, such as question answering, machine translation, summarization and language inference.

Several pre-trained BERT models are publicly available, including the following ones that we used in our experiments:

- BERT-Base Uncased: with 12 layers, hidden size of 768 and a total number of 110M parameters;
- BERT-Large Uncased: with 24 layers, hidden size of 1024 and a total number of 340M parameters.

4 Datasets

We tested the models on two datasets: SelQA and WikiQA. The first one is the one used in the Fujitsu AI-NLP Challenge, while the second one is a commonly used dataset for open-domain Question Answering. A more detailed description follows.

4.1 SelQA

The *SelQA* dataset (Jurczyk et al., 2016) was specifically created to be challenging for question answering systems, in particular by explicitly reducing word co-occurrences between question and answers. Questions with associated long sentence answers were generated through crowd-sourcing from articles drawn from the ten most prevalent topics in the English Wikipedia.

The dataset consists of a total of 486 articles that were randomly sampled from the topics of: Arts, Country, Food, Historical Events, Movies, Music, Science, Sports, Travel, TV. The original data was 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. To add some noise, annotators were also asked to create another set of questions from the same selected sections excluding the original sentences previously selected as answers. Then all questions were paraphrased using different terms, in order to ensure the QA algorithm would be evaluated by their reading comprehension ability rather than from statistical measures like counting word co-occurrences. Lastly if ambiguous questions were found, they were rephrased again by a human annotator.

4.2 WikiQA

The *WikiQA* dataset (Yang et al., 2015) dataset consists of 3047 questions sampled from *Bing* query logs from the period of May 1st, 2010 to July 31st, 2011. Each question is associated to sentences taken from a *Wikipedia* page assumed to be the topic of the question based on the user clicks. In order to eliminate answer sentence biases caused by key-word matching, the sentences were taken from the summary of this selected page.

The *WikiQA* dataset contains also questions for which there are no correct sentences to enable researchers to work on *answer triggering*.

This dataset has the drawback to be smaller compared to SelQA. Because of this, a model is more likely to over-fit the training set. To avoid this problem we added some strong regularization to the models.

5 Experiments

5.0.1 GloVe, ELMo and FastText

We carried out some preliminary experiments on the SelQA dataset, in order to determine which embeddings would work best with the CACN.

We tested three types of embeddings: *GloVe* (size 300), *ELMo* (Che et al., 2018) (size 1024) and *FastText* (Joulin et al., 2016) (size 300). With *ELMo* the model achieved comparable results to *GloVe*, but the training time was almost twice.

Model	Dev MRR	Test MRR
ELMo	91.09%	90.00%
FastText	89.47%	88.43%
GloVe	**91.37%**	**90.61%**

Table 2: Results for CACN on SelQA with various embeddings.

5.1 SelQA results

The logistic regression classifier obtains a score of 83.36 %, which is 7 points lower than CACN, not bad considering the simplicity of the model. Nevertheless this confirms that a simple word matching method is not competitive with more sophisticated methods on SelQA.

CACN was the best performing model on the Fujitsu AI NLP Challenge 2018, with a MRR of 90.61 %.

After the introduction of BERT, we decided to compare CACN with several versions of BERT, both alone and in combination with CACN.

We tried a few variant approaches. First, we fine-tuned a fully connected layer on top of BERT, leaving his parameters frozen, on the SelQA training set. This model achieved 91.17, a marginal improvement over CACN.

We then explored adding different networks on top of the BERT architecture.

We added a full CACN, on top of either the BERT-Base and BERT-Large models, with no improvement and even a drop with BERT-Large. Also in this case we froze the parameters of the BERT model.

Since these experiments did not provide improvements, we didn't try to train the entire model.

The best results were achieved by fine-tuning the BERT model on the SelQA dataset with a simple feed-forward layer, that achieved an impressive improvement of about 5 points to a MRR score of 95.29 %. Fine-tuning required about 4 hours on a server with an Nvidia P100 GPU.

The results of all our experiments on SelQA are summarized in table 3.

5.2 WikiQA results

In the experiments with CACN on WikiQA, we removed from the training set questions with no correct answer, but left the test set unchanged, so that the results are comparable with thos in the literature. This was done to preserve a similar structure to the SelQA dataset, which contains at least one correct answer for each question. This significantly reduced the number of training examples but, despite this, the MRR score of the CACN model improved.

Model	MRR
LR Classifier	83.36
CACN GloVe	90.61
BERT-Base + FCN	91.17
BERT-Base + CACN	91.11
BERT-Large + CACN	89.97
BERT-Base Fine-tuned	**95.29**

Table 3: Results on SelQA with various models.

Also in this case we kept the word embeddings fixed during training the CACN. We also added a dropout and normalization to regularize the model, that helped the model to better learn from the training set.

We then fine-tuned BERT on the WikiQA training set, performing full updates to the model, achieving again a significant improvement to a top score of 87.53 % MRR.

From the current leaderboard on the WikiQA dataset [2], we have extracted the top 5 entries and added the results with CACN and BERT-Base fine-tuned, as reported in Table 4.

Model	MRR	Year
BERT-Base Fine-tuned	**87.53 %**	2019
Comp-Clip + LM + LC	78.40 %	2019
RE2	76.18 %	2019
HyperQA (Tay et al., 2017)	72.70 %	2017
PWIM	72.34 %	2016
CACN (Gravina et al., 2018)	72.12 %	2018

Table 4: Experimental results on WikiQA.

5.3 Cross-domain experiments

In this section we report the results of our cross-domain experiments. The aim was to evaluate how well the CACN model performs in a context different from the one in which it was trained. In other words, we test the transfer learning ability of the model to a different domain.

The experiments consisted in training a model on one dataset and then testing it on the other one. We report in Table 5 the results of these experiments.

[2] https://paperswithcode.com/sota/question-answering-on-wikiqa

Trainset	Testset	MRR	Transfer score
SelQA	SelQA	90.61%	
SelQA	WikiQA	59.94%	82.95%
WikiQA	WikiQA	72.12%	
WikiQA	SelQA	69.45%	76.64%

Table 5: Cross domain experiments.

The drop in MRR score is small when training on WikiQA and testing on SelQA and larger in the other direction.

This is possibly due to the size of the datasets. In the second case in fact we are training on only 8000 pairs and testing on more than 80000 question/answer pairs.

However, the transfer score, computed as the ratio between the in-domain and out-domain MRR, is fairly good: about 83% in the SelQA to WikiQA case and over 76% in the other direction.

6 Conclusions

We compared the Cross-attentive Convolutional Network and several BERT based models on the task of Answer Sentence Selection on two datasets.

The experiments show that a BERT model, fine-tuned on an Answer Sentence Selection dataset, improves significantly the state of the art, with a gain of 5 to 9 points of MRR score on SelQA and WikiQA respectively. As a drawback, this approach takes a considerable amount of time to be trained even on GPUs.

The BERT-Base model without fine-tuning achieves almost the same accuracy as the CACN with GloVe embeddings, which uses a much smaller number of parameters in the model. The CACN also requires less data to train. On the other hand, BERT is quite effective at leveraging the knowledge collected from large amounts of unlabeled text, and at transferring it across tasks.

We also evaluated the abilities of CACN at transfer learning. BERT is a model that has been pre-trained on a large corpus, while CACN leverages the GloVe embeddings as a starting point for the training.

We also exploited the WikiQA and SelQA datasets in a cross-domain experiment using CACN. We found that the model maintains a good score across domains, with a transfer score of about 83% from SelQA to WikiQA.

We confirmed that the SelQA dataset is not easily solvable using simple word-occurrences methods like a logistic regression classifier on word count features.

BERT models confirmed their superiority to previous state of the art models for the task of Answer Sentence Selection. This was to be expected since they perform quite well also on the more complex task of Reading Comprehension, which requires not only to select a sentence but also to extract the answer from that sentence.

7 Acknowledgements

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.

References

Wanxiang Che, Yijia Liu, Yuxuan Wang, Bo Zheng, and Ting Liu. 2018. Towards better UD parsing: Deep contextualized word embeddings, ensemble, and treebank concatenation. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 55–64, Brussels, Belgium, October. Association for Computational Linguistics.

Nick Craswell. 2009. Mean Reciprocal Rank. In Ling Liu and M. Tamer Özsu, editors, *Encyclopedia of Database Systems*. Springer US, Boston, MA.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Minwei Feng, Bing Xiang, Michael R. Glass, Lidan Wang, and Bowen Zhou. 2015. Applying deep learning to answer selection: A study and an open task. *arXiv preprint arXiv:1508.01585*.

Alessio Gravina, Federico Rossetto, Silvia Severini, and Giuseppe Attardi. 2018. Cross attention for selection-based question answering. In *NL4AI@ AI* IA*, pages 53–62.

Michael Heilman and Noah A. Smith. 2010. 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*, pages 1011–1019. Association for Computational Linguistics.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hrve Jgou, and Tomas Mikolov. 2016. Fasttext.zip: Compressing text classification models. cite arxiv:1612.03651Comment: Submitted to ICLR 2017.

Tomasz Jurczyk, Michael Zhai, and Jinho D. Choi. 2016. SelQA: A New Benchmark for Selection-based Question Answering. In *Proceedings of the 28th International Conference on Tools with Artificial Intelligence, of ICTAI'16*, pages 820–827.

Jinfeng Rao, Hua He, and Jimmy Lin. 2016. 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)*, pages 1913–1916. ACM.

Ming Tan, Bing Xiang, and Bowen Zhou. 2015. Lstm-based deep learning models for non-factoid answer selection. *CoRR*, abs/1511.04108.

Yi Tay, Anh Tuan Luu, and Siu Cheung Hui. 2017. Enabling efficient question answer retrieval via hyperbolic neural networks. *CoRR*, abs/1707.07847.

Zhiguo Wang, Haitao Mi, and Abraham Ittycheriah. 2016. Sentence similarity learning by lexical decomposition and composition. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1340–1349. The COLING 2016 Organizing Committee.

Yi Yang, Scott Wen tau Yih, and Chris Meek. 2015. WikiQA: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. ACL Association for Computational Linguistics, September.

Xuchen Yao, Benjamin Van Durme, Chris Callison-Burch, and Peter Clark. 2013. 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*, pages 858–867.

Wenpeng Yin and Hinrich Schütze. 2017. Attentive Convolution. *CoRR*.