Representation Learning for Answer Selection with LSTM-Based Importance Weighting

Andreas Rücklé[†] and **Iryna Gurevych**[†‡]

[†]Ubiquitous Knowledge Processing Lab (UKP)
Department of Computer Science, Technische Universität Darmstadt
[‡]Ubiquitous Knowledge Processing Lab (UKP-DIPF)
German Institute for Educational Research
www.ukp.tu-darmstadt.de

Abstract

We present an approach to non-factoid answer selection with a separate component based on $BiLSTM$ to determine the importance of segments in the input. In contrast to other recently proposed attention-based models within the same area, we determine the importance while assuming the independence of questions and candidate answers. Experimental results show the effectiveness of our approach, which outperforms several state-of-the-art attention-based models on the recent non-factoid answer selection datasets InsuranceQA v1 and v2. We show that it is possible to perform effective importance weighting for answer selection without relying on the relatedness of questions and answers. The source code of our experiments is publicly available.[1]

1 Introduction

Answer selection is an important subtask of question answering (QA) that enables choosing one final answer from a list of candidate answers in regard to the input question (Feng et al., 2015; Wang and Nyberg, 2015). QA itself can be divided into factoid QA, which enables the retrieval of facts, and non-factoid QA, which enables finding of complex answer texts (e.g. descriptions, opinions, or explanations). Answer selection for non-factoid QA is especially difficult because we usually deal with user-generated content, for example questions and answers extracted from community question answering platforms or FAQ websites. As a consequence, candidate answers are complex multi-sentence texts with detailed information. Two examples are shown in Figures 2 and 3.

To deal with this challenge, recent approaches employ attention-based neural networks to focus on segments within the candidate answer that are most related to the question (Tan et al., 2016; Wang et al., 2016). For scoring, dense vector representations of the question and the candidate answer are learned and the distance between the vectors is measured. With attention-based models, segments with a stronger focus are treated as more important and have more influence on the resulting representations.

Using the relatedness between a candidate answer and the question to determine the importance is intuitive for correct candidate answers because the most important segments of both texts are expected to be strongly related. However, we also deal with a large number of incorrect candidate answers where the most important segments are usually dissimilar to the question. In such cases, the relatedness does not correlate with the actual importance. Thus, different methods for determining the importance could lead to better representations, especially when dealing with incorrect candidate answers.

In this work, we therefore determine the importance of segments in questions and candidate answers with a method that assumes the independence of both items. Our approach uses CNN and $BiLSTM$ for representation learning and employs a separate network component based on $BiLSTM$ for importance weighting. Our general concept is similar to self-attention mechanisms that have recently been integrated

[1] https://github.com/UKPLab/iwcs2017-answer-selection

to models for natural language inference and sentiment classification (Lin et al., 2017; Liu et al., 2016). They however employ feedforward components to derive importance values and deal with classification problems. In contrast, we directly compare learned representations with a similarity measure and derive the importance using a separate $BiLSTM$, which was motivated by the effectiveness of stacked models in answer selection (Tan et al., 2016; Wang and Nyberg, 2015).

We evaluate our approach on two non-factoid answer selection datasets that contain data from a community question answering platform: InsuranceQA v1 and InsuranceQA v2. In comparison to other state-of-the-art representation learning approaches with attention, our approach achieves the best results and significantly outperforms various strong baselines. An additional evaluation on the factoid QA dataset WikiQA demonstrates that our approach is well-suited for other scenarios that deal with shorter texts. In general, we show that it is possible to perform effective importance weighting in non-factoid answer selection without relying on the relatedness of questions and candidate answers.

2 Related Work

Earlier work in answer selection relies on handcrafted features based on semantic role annotations (Shen and Lapata, 2007; Surdeanu et al., 2011), parse trees (Wang and Manning, 2010; Heilman and Smith, 2010), tree kernels (Moschitti et al., 2007; Severyn and Moschitti, 2012), discourse structures (Jansen et al., 2014), and external resources (Yih et al., 2013).

More recently, researchers started using deep neural networks for answer selection. Yu et al. (2014), for example, propose a convolutional bigram model to classify a candidate answer as correct or incorrect. Similar but more enhanced, Severyn and Moschitti (2015) use a CNN with additional dense layers to capture interactions between questions and candidate answers, a model that is also part of a combined approach with tree kernels (Tymoshenko et al., 2016). And Wang and Nyberg (2015) incorporate stacked $BiLSTMs$ to learn a joint feature vector of a question and a candidate answer for classification.

Answer selection can also be formulated as a ranking task where we learn dense vector representations of questions and candidate answers and measure the distance between them for scoring. Feng et al. (2015) use such an approach and compare different models based on CNN with different similarity measures. Based on that, models with attention mechanisms were proposed. Tan et al. (2016) apply an attentive $BiLSTM$ component that performs importance weighting before pooling based on the relatedness of segments in the candidate answer to the question. Dos Santos et al. (2016) introduce a two-way attention mechanism based on a learned measure of similarity between questions and candidate answers. And Wang et al. (2016) propose novel ways to integrate attention inside and before a GRU.

In this work, we use a different method for importance weighting that determines the importance of segments in the texts while assuming the independence of questions and candidate answers. This is related to previous work in other areas of NLP that incorporate self-attention mechanisms. Within natural language inference, Liu et al. (2016) derive the importance of each segment in a short text based on the comparison to a average-pooled representation of the text itself. Parikh et al. (2016) determine intra-attention with a feedforward component and combine the importance of nearby segments. And Lin et al. (2017) propose a model that derives multiple attention vectors with matrix multiplications. Within factoid QA, Li et al. (2016) weight the importance of each token in a question with a feedforward network and perform sequence labeling.

In contrast to those, we apply this concept to answer selection, we directly compare vector representations of questions and candidate answers, and we use a separate ***BiLSTM*** for importance weighting.

3 Representation Learning for Answer Selection

We formulate answer selection as a ranking task. Given a question q and a pool A of candidate answers, the goal is to re-rank A according to a scoring function that judges each candidate answer $a \in A$ for relevancy in regard to q. The best-ranked candidate answer is then selected. For scoring we learn dense vector representations of q and a and calculate the similarity between those vectors.

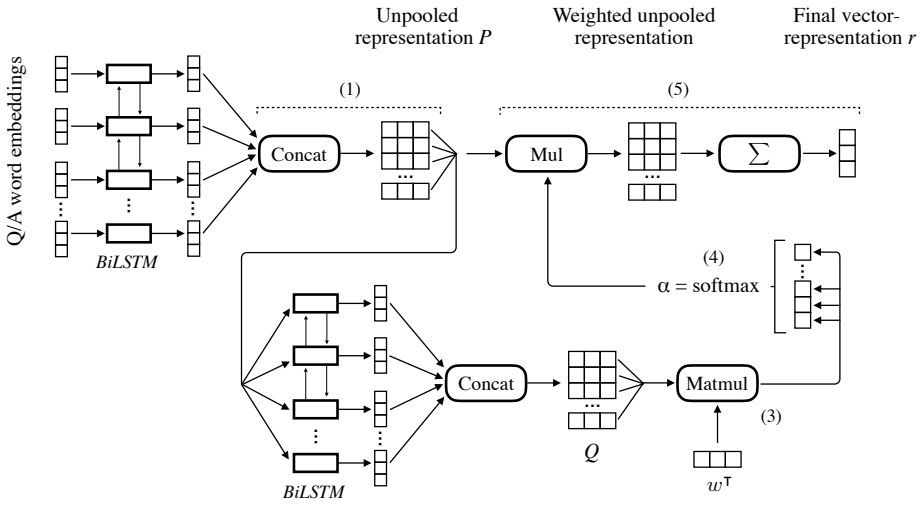

Figure 1: The network structure of LW with $BiLSTM$ to learn the unpooled representation (LW_{BiLSTM}). Numbers in parentheses refer to the related Equations.

Basic BiLSTM Model The best-performing models for representation learning in non-factoid answer selection are usually based on $BiLSTMs$ (Tan et al., 2016; Dos Santos et al., 2016). Thus, we build our own approach on a variation of such model. To obtain a representation for an input text we apply an $LSTM$ on the concatenated d-dimensional word embeddings $E \in \mathbb{R}^{l \times d}$ of the input text with length l in forward direction and in backward direction. As a result, we obtain two matrices $H^{\rightarrow}, H^{\leftarrow} \in \mathbb{R}^{l \times c}$ that contain the state vectors of each recurrence (c is the $LSTM$ cell size). We define the unpooled representation P as the row-wise concatenation of both matrices and create a fixed-size dense vector representation r of the question or candidate answer by applying 1-max pooling:

$$P_i = [H_i^{\rightarrow},\ H_i^{\leftarrow}] \tag{1}$$

$$r_j = \max_{1<i<l} (P_{i,j}) \tag{2}$$

where $P \in \mathbb{R}^{l \times 2 \cdot c}$ and $r \in \mathbb{R}^{2 \cdot c}$.

We can also use CNN for learning text representations. In this case, P contains the values of all filter operations applied on all n-grams in the input text and the dense vector representation r is calculated with 1-max pooling as before. Formal definitions can be found in (Feng et al., 2015; Dos Santos et al., 2016).

LSTM-Based Importance Weighting (LW) The basic $BiLSTM$ model is often extended with different attention mechanisms that utilize the relatedness between questions and candidate answers to focus on the most relevant segments of the texts (Tan et al., 2016; Wang et al., 2016; Dos Santos et al., 2016). In contrast, we perform importance weighting while assuming the independence of both items. As a consequence, we do not rely on the relatedness to determine the importance.

Our approach LW is an extension to simple representation learning models and can be used instead of 1-max pooling. We first create an encoding of the importance for each segment in the unpooled representation P of a prior component (e.g. the basic $BiLSTM$) by applying an additional, separate $BiLSTM$. We obtain the concatenated output states $Q \in \mathbb{R}^{l \times 2 \cdot c}$ of this $BiLSTM$ where the ith row Q_i contains the state vectors that encode the importance of the ith row in P. We then reduce each row Q_i to a scalar v_i and apply softmax on the vector v to obtain scaled importance values that sum to 1.0:

$$v_i = w^{\intercal} Q_i \tag{3}$$

$$\alpha = softmax(v) \tag{4}$$

Dataset	Train Questions	Valid Questions	Test Questions	Candidates per Question	Correct Answers per Question	Answer Length in Tokens
InsuranceQA v1	12,887	1,000	3,600	500.0	1.4	96.5
InsuranceQA v2	12,889	1,592	1,625	500.0	1.6	111.8
WikiQA	873	126	243	9.8	1.2	25.2

Table 1: Dataset statistics.

where $w \in \mathbb{R}^c$ are learned network parameters for the reduction operation, $v_i \in \mathbb{R}$ is the (unscaled) importance value of the ith segment in P, and $\alpha \in \mathbb{R}^l$ is the resulting importance vector (or attention vector). Applying softmax is important because we do not want more accumulated importance for longer texts compared to shorter texts. Finally, we reduce P to a fixed-size dense vector representation r according to our importance vector α:

$$r_j = \sum_{i=1}^{l} \alpha_i P_{i,j} \tag{5}$$

In contrast to average pooling or 1-max pooling, this operation allows different segments in the input to contribute to r with different strengths (having more or less influence on r). A visualization of LW that uses $BiLSTM$ to learn the unpooled representation P is shown in Figure 1.

In general, we always use shared network weights to learn the unpooled representation P of questions and candidate answers as it is more effective compared to using separate network weights (Feng et al., 2015). Within the components of LW we however use separate network weights, which allows the network to learn different importance weighting behavior for questions and candidate answers. We analyze the impact of this choice later in Section 5.

4 Experimental Setup

Training We define the loss $\mathcal{L}$ as follows:

$$\mathcal{L} = max\left(0,\ m - s(r^q, r^{a+}) + s(r^q, r^{a-})\right)$$

where r^q is the learned question representation, r^{a+} and r^{a-} are learned representations of correct and incorrect candidate answers, s is cosine similarity, and m is the desired margin between the similarities. Because such triples are not pre-defined in our datasets, we construct them during training. For a pair of question and correct answer we randomly sample 50 incorrect candidate answers from the whole training set and select the candidate with the highest similarity according to our currently trained model.

Datasets We evaluate our models on the two recent non-factoid answer selection datasets InsuranceQA v1 and InsuranceQA v2 (Feng et al., 2015). In general, both datasets contain more than 15,000 questions and the candidate answers are long multi-sentence texts. Even though InsuranceQA v1 and v2 were crawled from the same community question answering website, they model different setups due to a different sampling strategy that was used to create the candidate answer pools. Whereas in InsuranceQA v1 the pools were created randomly (plus the correct answers), the pools in InsuranceQA v2 were created by querying a search engine to retrieve candidate answers that are lexically similar to the question.[2]

In addition, we also test our approaches on the factoid answer selection dataset WikiQA, which was constructed by means of crowd-sourcing through the extraction of sentences from Wikipedia articles (Yang et al., 2015). We use this dataset to test our models within the different scenario of factoid answer selection that deals with significantly shorter texts. The dataset statistics are listed in Table 1.

[2]Since the correct answers were not separately inserted in InsuranceQA v2, the pools are not guaranteed to contain a correct answer. We discard all questions without any correct answer in the associated pool of candidate answers.

Model	Valid	Test
$AttentiveBiLSTM$ (Tan et al., 2016)	68.9	66,9
$IABRNN$ (Wang et al., 2016)	69.1	67.0
AP_{BiLSTM} (Dos Santos et al., 2016)	68.4	69.1
CNN	60.5	58.3
$BiLSTM$	68.2	65.7
$CNN+BiLSTM$	68.5	67.3
$BiLSTM+BiLSTM$	67.5	66.3
LW_{CNN}	70.0	67.9
LW_{BiLSTM}	**70.9**	**70.0***

Table 2: Experimental results on InsuranceQA v1 (accuracy). * = significant improvement against our other models ($p < 0.05$, Wilcoxon test).[3]

Model	Valid	Test
AP_{BiLSTM} *(reimplementation)*	32.2	31.9
CNN	24.4	24.4
$BiLSTM$	32.4	31.1
$CNN+BiLSTM$	33.0	31.4
$BiLSTM+BiLSTM$	31.2	32.0
LW_{CNN}	33.5	33.7
LW_{BiLSTM}	**35.4**	**36.9***

Table 3: Experimental results on InsuranceQA v2 (accuracy). * = significant improvement against all other models ($p < 0.05$, Wilcoxon test).

Models and Baselines We evaluate LW with $BiLSTM$ (LW_{BiLSTM}) and CNN (LW_{CNN}) to learn the unpooled representations. As baselines we employ $BiLSTM$ and CNN with 1-max pooling and the stacked variants $CNN+BiLSTM$ and $BiLSTM+BiLSTM$, which use a $BiLSTM$ with 1-max pooling to process the unpooled representation P of the prior component.

A comparison against the stacked models is particularly important because they employ the same components as LW_{CNN} and LW_{BiLSTM}, but use a different network structure.

Neural Network Setup We performed grid search over several hyperparameter combinations and found the optimal choices to be similar to hyperparameters of previous work. The cell size of all $LSTMs$ is 141 (each direction), and the number of filters for all $CNNs$ is 400 with size 3. The only exception is $CNN+BiLSTM$ with 282 filters and a cell size of 282. We use the Adam optimizer (Kingma and Ba, 2015) with a learning rate of $4 \cdot 10^{-4}$ and a margin $m = 0.2$. We initialize the word embeddings with off-the-shelf 100-dimensional uncased GloVe embeddings (Pennington et al., 2014) and optimize them further during training. Dropout of 0.3 was applied on the representations before comparison.

We chose different hyperparameters for WikiQA, which we do not list here due to space restrictions. Details can be found in our public source code repository.

5 Experimental Results

InsuranceQA v1 Our evaluation on InsuranceQA v1 allows us to compare our approach against a broad list of recently published attention-based models. Table 2 shows the results of our evaluation where we measure the ratio of correctly selected answers (accuracy). We observe that by adding LW to either CNN or $BiLSTM$ we can significantly improve the answer selection performance by 9.6% and 4.3% respectively. This clearly shows that LW is effective and can be used to extend basic models to learn better representations of questions and candidate answers. Additionally, LW models are more effective than stacked models due to the different network structure that we use to explicitly learn importance weights. Stacked models are less effective because they need to carry the full representation through all components. Overall, LW_{BiLSTM} significantly outperforms all our other tested models. LW_{BiLSTM} also achieves the best results compared to other state-of-the-art representation learning approaches with attention such as the two-way attention model AP_{BiLSTM}, which derives attention from a learned measure of similarity between questions and answers. This clearly shows that we can successfully perform importance weighting without relying on the relatedness of questions and answers.

It is important to mention that Wang and Jiang (2017) very recently experimented with a novel

[3]We did not have access to the predictions of other top-performing approaches, hence, we report significance against our own models. We note that the differences are however within the usual margins of this dataset.

Model	MAP	MRR
AP_{CNN} (Dos Santos et al., 2016)	0.6886	0.6957
$ABCNN$ (Yin et al., 2016)	0.6921	0.7127
$IABRNN$ (Wang et al., 2016)	**0.7341**	**0.7418**
CNN	0.6204	0.6365
$BiLSTM$	0.6174	0.6310
$CNN+BiLSTM$	0.6560	0.6737
$BiLSTM+BiLSTM$	0.6735	0.6789
LW_{CNN}	**0.7102**	**0.7240**
LW_{BiLSTM}	0.6941	0.7039

Table 4: Experimental results on WikiQA compared to recent approaches with attention.

Model	InsuranceQA		WikiQA	
	V1	V2	MAP	MRR
LW_{CNN} / shared	67.8	**34.0**	0.6992	0.7112
LW_{CNN} / sep.	**67.9**	33.7	**0.7102**	**0.7240**
LW_{BiLSTM} / shared	68.5	36.1	0.6854	0.6954
LW_{BiLSTM} / sep.	**70.0**	**36.9**	**0.6941**	**0.7039**

Table 5: Experimental results with shared vs. separate LW weights.

method that achieves state-of-the-art results on the InsuranceQA v1 dataset.[4] Instead of learning dense vector representations, they classify pairs of questions and candidate answers with a compare-aggregate model that performs comparisons on the word level, aggregates this information with CNN, and uses additional layers to determine the classification result. Because their approach is not learning dense vector representations, we did not directly compare against it. It would however be possible to use our approach in their framework to compare segments of weighted unpooled representations.

InsuranceQA v2 The evaluation on InsuranceQA v2 allows us to compare our models within a more realistic answer selection scenario due to the different creation of candidate answer pools. Because there are no previously published results, we re-implemented Attentive Pooling with $BiLSTM$ (AP_{BiLSTM}) as proposed by Dos Santos et al. (2016) for a better comparison.[5] We report the experimental results in Table 3. Similar to our previous findings, LW significantly improves the answer selection performance of CNN and $BiLSTM$. In contrast, AP_{BiLSTM} only achieves minor improvements against $BiLSTM$. We expect this to be an effect of the more realistic candidate answer pools where all incorrect candidates are lexically similar to the question. Because AP_{BiLSTM} uses an explicitly learned measure of similarity between questions and candidate answers to determine the importance, it assigns high scores to lexically similar incorrect candidate answers. On the other hand, our experimental results suggest that LW is not affected by this issue. As a consequence, our best model LW_{BiLSTM} significantly outperforms all other approaches, showing that importance weighting without relying on the relatedness of questions and answers is very effective within the realistic answer selection scenario of InsuranceQA v2.

Since our best observed accuracy on this dataset is significantly lower than on InsuranceQA v1, we tried to determine the actual usefulness of our approach. We manually labeled the first 100 incorrectly selected answers of $BiLSTM$ and LW_{BiLSTM} for correctness, where a candidate answer is correct if it contains the information that was requested in the question. In the case of LW_{BiLSTM}, 50 answers were labeled as correct, and for $BiLSTM$ the number of correct labels is 44. The improvement of LW_{BiLSTM} is often driven by a sharp question focus, which enables to better retrieve answers that contain the requested information. These numbers indicate that the actual usefulness of our models is higher than the reported accuracy scores. The primary issue is the number of missing labels in the dataset, which is a result of the different sampling strategy and the lack of manual relevance annotations. We however did not notice any particular consequences from this situation beyond under-estimating the model performance.

WikiQA Experiments on WikiQA allow us to test our proposed approach within a different scenario that deals with considerably shorter texts. Following Yang et al. (2015), we measure MAP and MRR within our evaluation. The results are listed in Table 4.

Similar to our results on both InsuranceQA datasets, the addition of LW substantially improves the answer selection performance. Neither the reduced length of the answers nor the significantly reduced

[4]They evaluated many different variations of their approach and achieve a maximum accuracy of 74.3%.

[5]Our re-implementation achieves similar results on InsuranceQA v1 as reported by (Dos Santos et al., 2016).

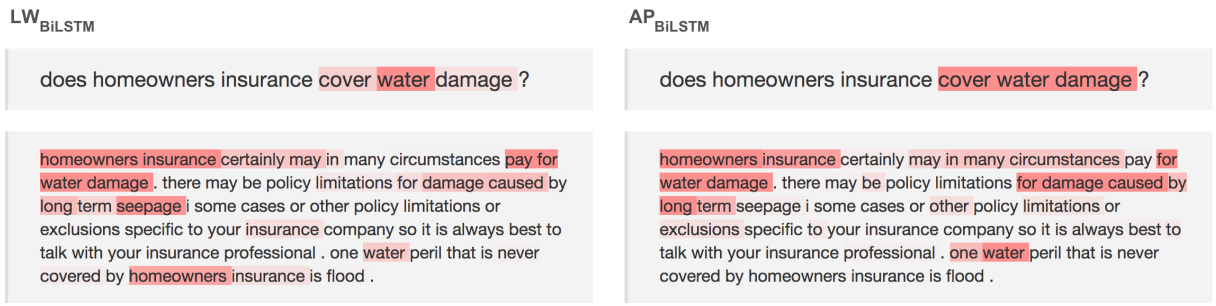

Figure 2: A visualization of the attention weights of LW_{BiLSTM} and AP_{BiLSTM} for a question and a correct answer. Red colors visualize the relative importance.

size of the training data has a noticeable influence on the performance. Compared to the stacked models, the performance increase of LW models is also considerable. Even though our best model LW_{CNN} does not achieve state-of-the-art results on this dataset (the best results are currently achieved by Wang and Jiang (2017) with 0.7433 MAP), we note that it performs on the same level as other top-performing attention-based models. This suggests that our approach can be suitably applied to scenarios that are different to non-factoid answer selection.

Separate vs. Shared LW Network Weights To measure the impact of our choice to use separate LW parameters for questions and candidate answers, we re-ran all experiments with shared parameters and provide a comparison in Table 5.

We observe that using separate LW parameters leads to improvements in 5 out of 6 cases, where LW_{BiLSTM} obtains the biggest gains of up to 1.5% accuracy. This suggests that learning separate parameters for the importance weighting of questions and candidate answers can lead to better representations. Even though this is intuitive because questions and answers are different types of texts, previous work has shown that using separate network parameters usually results in performance declines (Feng et al., 2015). However, since we still use shared parameters to learn the unpooled representations and only use separate parameters in LW, our approach does not suffer from the same optimization issues.

6 Analysis

Importance Weights We qualitatively analyzed the importance weights of LW_{BiLSTM} and AP_{BiLSTM} using an end-to-end QA framework with attention visualization (Rücklé and Gurevych, 2017) and configured it to use InsuranceQA v2. In general, we oberserved that for pairs of questions and correct candidate answers, the most important segments determined by LW_{BiLSTM} and AP_{BiLSTM} are very similar. An example is given in Figure 2. We also noticed two important attributes of LW that contribute to the previously reported improvements.

First, for incorrect candidate answers with high lexical similarity to the question, LW_{BiLSTM} often focusses on segments that happen to be unrelated and thus creates dissimilar representations (desired). In contrast, AP_{BiLSTM}, by design, focusses on similar segments and creates similar representations (undesired). An example is shown in Figure 3, where our approach strongly focusses on a segment within the question that corresponds to the word *when*. This requires candidate answers to have a similar focus in order to achieve a high score (e.g. by describing a date).[6] Since this is not the case for the presented incorrect candidate answer, the representations are dissimilar and the score is low. This allows LW to better handle incorrect candidate answers.

And second, we found that LW_{BiLSTM} very strongly focusses on few highly relevant segments that are well-suited to describe the overall topic of the text. This leads to representations that are strongly based on individual aspects and allows the model to filter out noise more effectively because irrelevant segments

[6]Our approach sometimes focusses on words indicative for the question type (wh-type words), but this is not always the case. If an important noun is present in the question, *LW* most often focusses on that (e.g. *fire*, *water*, *electricity*).

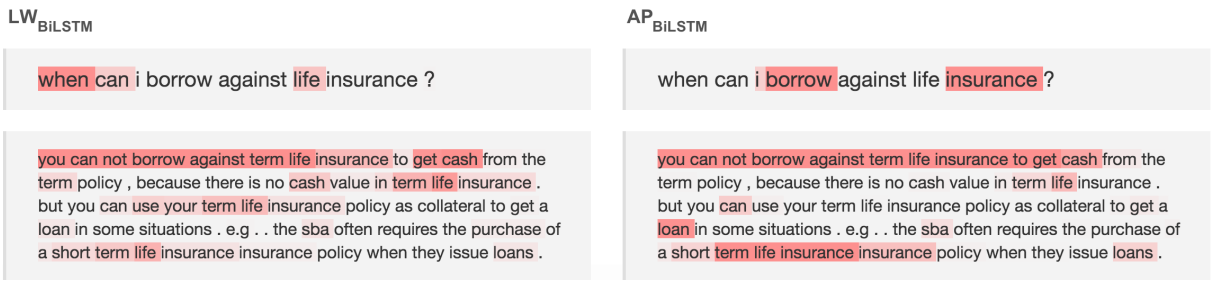

Figure 3: A visualization of the attention weights of LW_{BiLSTM} and AP_{BiLSTM} for a question and an incorrect candidate answer (with high lexical similarity). Red colors visualize the relative importance.

receive lower relative importance. We quantitatively analyzed this property by measuring the strength of the importance weights for all answers in InsuranceQA v2. For each individual question/answer pair (correct or incorrect) we determined the maximum values of the importance weights with LW_{BiLSTM} and AP_{BiLSTM}. Interestingly, LW_{BiLSTM} derives at least one importance weight greater or equal 0.10 within 77% of all answers, and one importance weight greater or equal 0.20 within 24% of all answers.[7] AP_{BiLSTM} on the other hand does not apply such a strong focus (0% of cases; a very small number). As a consequence, LW can better ignore irrelevant content because it strongly focusses on few important segments within the relatively long texts found in InsuranceQA v2.

Error Analysis and Limitations The most common error we observed is related to important aspects of the question that are not addressed in the selected answer. The question *"What is a renters insurance declaration page?"*, for example, contains the aspects *what* (question type), *renters insurance*, and *declaration page*. When LW_{BiLSTM} fails, it usually selects an answer that differs in only one aspect. For the previous question, our approach selects an answer that describes what the *auto insurance* declaration page is (a similar topic). The reason is the inability of LW to focus on all important aspects of the question separately. This can also be observed in our previous example in Figure 2, where our approach focusses on the aspects *cover* and *water damage* but ignores *homeowners insurance*. In this case our approach would not be able to effectively differentiate between candidate answers that write about *renters insurance* instead of *homeowners insurance*.

To tackle this issue, future work could add a separate classification step after ranking that discards any top-ranked answers that do not cover all aspects of the question.

7 Conclusion

In this work, we presented an approach to non-factoid answer selection that determines the importance of segments within questions and answers while assuming the independence of both items. Our experimental results on the two non-factoid answer selection datasets InsuranceQA v1 and v2 show that our approach is effective and substantially outperforms various strong baselines and different state-of-the-art attention-based approaches. Our additional evaluation on WikiQA demonstrates that our proposed approach is also suitable for different scenarios with shorter texts. We showed that it is possible to perform effective importance weighting for answer selection without relying on the relatedness of questions and answers.

Acknowledgements

This work has been supported by the German Research Foundation as part of the QA-EduInf project (grant GU 798/18-1 and grant RI 803/12-1). We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Tesla K40 GPU used for this research. Some calculations for this research were conducted on the Lichtenberg high performance computer of the TU Darmstadt.

[7] Segments with a related importance weight of 0.10 have a high influence on the representation (10%).

References

Dos Santos, C., M. Tan, B. Xiang, and B. Zhou (2016). Attentive Pooling Networks. *arXiv preprint*.

Feng, M., B. Xiang, M. R. Glass, L. Wang, and B. Zhou (2015). Applying deep learning to answer selection: A study and an open task. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pp. 813–820.

Heilman, M. and A. N. 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*, pp. 1011–1019. Association for Computational Linguistics.

Jansen, P., M. Surdeanu, and P. Clark (2014). Discourse complements lexical semantics for non-factoid answer reranking. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 977–986. Association for Computational Linguistics.

Kingma, D. P. and J. L. Ba (2015). Adam: a Method for Stochastic Optimization. In *3rd International Conference on Learning Representations (ICLR)*.

Li, P., W. Li, Z. He, X. Wang, Y. Cao, J. Zhou, and W. Xu (2016). Dataset and Neural Recurrent Sequence Labeling Model for Open-Domain Factoid Question Answering. *Arxiv preprint*.

Lin, Z., M. Feng, C. N. Dos Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio (2017). A Structured Self-attentive Sentence Embedding. *5th International Conference on Learning Representations (ICLR)*.

Liu, Y., C. Sun, L. Lin, and X. Wang (2016). Learning Natural Language Inference using Bidirectional LSTM model and Inner-Attention. *Arxiv preprint*.

Moschitti, A., S. Quarteroni, R. Basili, and S. Manandhar (2007). Exploiting syntactic and shallow semantic kernels for question answer classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL)*, pp. 776–783. Association for Computational Linguistics.

Parikh, A. P., O. Täckström, D. Das, and J. Uszkoreit (2016). A Decomposable Attention Model for Natural Language Inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 2249–2255. Association for Computational Linguistics.

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

Rücklé, A. and I. Gurevych (2017). End-to-end non-factoid question answering with an interactive visualization of neural attention weights. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics-System Demonstrations (ACL)*, pp. 19–24. Association for Computational Linguistics.

Severyn, A. and A. Moschitti (2012). Structural relationships for large-scale learning of answer re-ranking. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 741–750. ACM.

Severyn, A. and A. Moschitti (2015). Learning to rank short text pairs with convolutional deep neural networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 373–382. ACM.

Shen, D. and M. Lapata (2007, June). Using semantic roles to improve question answering. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pp. 12–21. Association for Computational Linguistics.

Surdeanu, M., M. Ciaramita, and H. Zaragoza (2011). Learning to rank answers to non-factoid questions from web collections. *Computational Linguistics 37*(2), 351–383.

Tan, M., C. Dos Santos, B. Xiang, and B. Zhou (2016). Improved representation learning for question answer matching. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 464–473. Association for Computational Linguistics.

Tymoshenko, K., D. Bonadiman, and A. Moschitti (2016). Convolutional neural networks vs. convolution kernels: Feature engineering for answer sentence reranking. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 1268–1278. Association for Computational Linguistics.

Wang, B., K. Liu, and J. Zhao (2016). Inner attention based recurrent neural networks for answer selection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 1288–1297. Association for Computational Linguistics.

Wang, D. and E. Nyberg (2015). A long short-term memory model for answer sentence selection in question answering. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, pp. 707–712. Association for Computational Linguistics.

Wang, M. and C. Manning (2010). Probabilistic tree-edit models with structured latent variables for textual entailment and question answering. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING)*, pp. 1164–1172.

Wang, S. and J. Jiang (2017). A Compare-Aggregate Model for Matching Text Sequences. *5th International Conference on Learning Representations (ICLR)*.

Yang, Y., W.-t. Yih, and C. Meek (2015). Wikiqa: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 2013–2018. Association for Computational Linguistics.

Yih, W.-T., M.-W. Chang, C. Meek, and A. Pastusiak (2013). Question answering using enhanced lexical semantic models. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 1744–1753. Association for Computational Linguistics.

Yin, W., H. Schütze, B. Xiang, and B. Zhou (2016). Abcnn: Attention-based convolutional neural network for modeling sentence pairs. *Transactions of the Association of Computational Linguistics (TACL) 4*, 259–272.

Yu, L., K. M. Hermann, P. Blunsom, and S. Pulman (2014). Deep Learning for Answer Sentence Selection. In *NIPS Deep Learning Workshop*.