COALA: A Neural Coverage-Based Approach for Long Answer Selection with Small Data

Andreas Rücklé,[†] Nafise Sadat Moosavi,[†‡] Iryna Gurevych[†]
[†]Ubiquitous Knowledge Processing Lab (UKP)
[‡]Research Training Group AIPHES
Department of Computer Science, Technische Universität Darmstadt
[†]www.ukp.tu-darmstadt.de
[‡]www.aiphes.tu-darmstadt.de

Abstract

Current neural network based community question answering (cQA) systems fall short of (1) properly handling long answers which are common in cQA; (2) performing under small data conditions, where a large amount of training data is unavailable—i.e., for some domains in English and even more so for a huge number of datasets in other languages; and (3) benefiting from syntactic information in the model—e.g., to differentiate between identical lexemes with different syntactic roles. In this paper, we propose COALA, an answer selection approach that (a) selects appropriate long answers due to an effective comparison of all question-answer aspects, (b) has the ability to generalize from a small number of training examples, and (c) makes use of the information about syntactic roles of words. We show that our approach outperforms existing answer selection models by a large margin on six cQA datasets from different domains. Furthermore, we report the best results on the passage retrieval benchmark WikiPassageQA.

Introduction

Question answering (QA) systems generally retrieve facts from knowledge bases (Bao et al. 2014) or from web documents (Wang et al. 2018). However, many types of questions require information that cannot be found in these resources, e.g., explanations, descriptions, or advice. In this work, we focus on community question answering (cQA), and in particular on cQA answer selection, where we retrieve relevant answers to *non-factoid* questions from social Q&A communities (Verberne et al. 2010; Tay et al. 2017; Nakov et al. 2017). In contrast to factoid QA where questions can be answered with an individual entity or a single sentence (Yang, Yih, and Meek 2015; Wang, Smith, and Mitamura 2007), in cQA we often deal with long multi-sentence texts—e.g., in StackExchange Academia we observe an average answer length of 229 words. This presents a difficult challenge to current neural answer selection approaches because they were primarily designed to retrieve short answers (Cohen, Yang, and Croft 2018).

A popular state-of-the-art approach for answer selection is the relevance matching model by Wang and Jiang (2017). It is based on the general *compare-aggregate* framework (He and Lin 2016; Parikh et al. 2016; Wang and Jiang 2017), which first compares the aspects of the answer (e.g., individual words) to the aspects of the question and then aggregates this information. Wang and Jiang (2017)'s approach is rather complex and consists of several deep layers.

Despite their strong performances across a number of text matching tasks, such approaches have three important limitations on which we focus in this work.

First, the ability to deal with long answers is a crucial property of cQA as opposed to classical QA, and typically state-of-the-art answer selection approaches fall short in these cases (Cohen, Yang, and Croft 2018).

Second, since they are based on complex deep neural network architectures, current cQA approaches require large amounts of training data. However, such data is not available in many cQA communities,[1] let alone non-English data. Thus, we need approaches that can effectively learn from small training data to handle low-resource scenarios.

And third, neural models often ignore the linguistic structure of sentences such as dependency relations. These structures can, however, discriminate similar words in different contexts, which is beneficial when all candidate answers have high lexical similarity to the question.

In this work, we tackle these three challenges and propose compare-aggregate for long answers (COALA), an answer selection approach based on the compare-aggregate framework with a coverage-based method that ranks answers based on the extent of covered question aspects. While our general neural network architecture is motivated by Wang and Jiang (2017)'s success on different text matching tasks, our proposed approach has three unique properties that correspond to the previously identified limitations.

First, Wang and Jiang (2017) focus on answer aspects and aggregate the comparisons of all answer aspects to the question. COALA, on the other hand, focuses on question aspects and measures to which extent all question aspects are covered by the answer. Therefore, its aggregation is independent from the answer's complexity.

[1]For instance, there are 35 communities in StackExchange with less than 2,000 questions in total (including unanswered questions).

Second, COALA uses simple yet effective operations: instead of attention-based alignment and neural aggregation, it uses max-pooling and averaging techniques, respectively. Therefore, our network does not require large amounts of training data and can be applied to low-resource scenarios.

And third, our approach is extensible with linguistic structures and more complex aggregation functions. We present two enhanced variants of COALA, one that utilizes the syntactic structures of sentences, which allows the network to differentiate similar words in different local contexts, and one that uses a number of (learned) power means (Hardy, Littlewood, and Pólya 1952) to extract additional information from the aspect comparisons for aggregation.

We evaluate COALA on six cQA answer selection datasets from different domains and show that:

(a) As a result of being independent of the answers's complexity, COALA outperforms different state-of-the-art answer selection models by a large margin, i.e., by more than 4.5pp accuracy. Our improved aggregation with power mean further improves the results by 1.6pp. Most importantly, COALA can handle long answers substantially better than other approaches; it achieves 21pp improvement over the state of the art when answers are longer than 250 words.

(b) Due to its simple network architecture, COALA can be applied to low-resource scenarios. For example, our approach outperforms a strong unsupervised baseline by more than 3pp and a state-of-the-art supervised approach by 32pp when both have access to only 25 training questions.

(c) The incorporation of syntactic information leads to consistent gains, which highlights the benefit of linguistically-informed language representations.

Further, we evaluate COALA on WikiPassageQA (Cohen, Yang, and Croft 2018), a recent benchmark dataset for passage retrieval and non-factoid QA within larger documents. Here, our approach also achieves new state-of-the-art results, which demonstrates its potential to serve as a strong baseline for other tasks that deal with the retrieval of long texts.

Finally, our analysis shows that while COALA is highly effective for tasks with long answers, other approaches that take all aspects of the answer into account are better suited to deal with very short answers. This reveals that tasks with varying answer lengths, e.g., cQA and factoid QA, require fundamentally different approaches to obtain optimal results.

Related Work

We divide cQA answer selection into two categories:

Semantic similarity approaches compare learned dense vector representations of questions and answers for scoring.

Early approaches use CNNs with max pooling (Feng et al. 2015), whereas more recent approaches rely on attentional LSTMs. For example, Tan et al. (2016) use an attention mechanism that assigns higher weight to words in the answer that are related to the question, Dos Santos et al. (2016) use bidirectional attention based on the similarity between question and answer representations, Wang, Liu, and Zhao (2016) propose attention inside and before GRUs, and Rücklé and Gurevych (2017) use a self-attentive approach with a separate LSTM that learns the importance of text segments.

Relevance matching varies from unsupervised approaches like TF*IDF and BM25 (Robertson and Walker 1994) to simple neural networks (Yu et al. 2014), tree kernels (Tymoshenko, Bonadiman, and Moschitti 2016; Romeo et al. 2016), and more complex multilayer neural networks.

For example, Lu and Li (2013) match short texts with local and hierarchical structures in a neural network, Hu et al. (2014) and Shen et al. (2015) use multiple CNNs to compare and match short texts, Severyn and Moschitti (2015) use interaction matrices and learned text representations for relevance scoring, Yang et al. (2016) propose a model based on attention and question term importance, and Zhang et al. (2017) combine question-answer interactions with attention mechanisms and additional hand-crafted features. Similar models have also been proposed in the context of ad-hoc retrieval (Guo et al. 2016; Pang et al. 2017).

Many of the existing relevance matching approaches can be described in the compare-aggregate framework (He and Lin 2016; Parikh et al. 2016; Wang and Jiang 2017) that first compares the relevance of individual aspects of question and answer and then aggregates this information for prediction.

A popular state-of-the-art approach, which achieves the best results on several text matching tasks, including relevant answer selection datasets, is the compare-aggregate variant by Wang and Jiang (2017). Their approach consists of four steps: (1) *aspect extraction*, which learns a representation for each word of the question and answer using either LSTM (for SNLI) or a gated importance-weighted representation of words; (2) *attention*, which learns an alignment between question and answer aspects using a standard attention mechanism such that the jth element of the attention vector represents the parts of the question that best match the jth aspect of the answer; (3) *comparison*, which combines the results of the question attention vector and answer aspects and captures their interactions; (4) *aggregation*, which uses a CNN with max-pooling to aggregate the interactions over the answer.

The most distinguishing difference of our approach compared to Wang and Jiang (2017) is that their comparison (and aggregation) determines how well each answer aspect is related to one or more question aspects. In contrast, we explicitly determine the coverage of all question aspects by the answer. Our aggregation is thus independent of the answer complexity and can scale better to long answers, which has recently been identified as an important and difficult problem in passage retrieval (Cohen, Yang, and Croft 2018).

COALA: Compare Aggregate for Long Answers

Answer selection requires finding a function f that scores each answer A in a pool of candidate answers according to its relevancy in regard to the question Q. The best candidate answer is then selected according to this score.

In this setup we can formalize relevance matching approaches as follows:

$$f(Q, A) = \Omega\left(\Phi(Q), \Phi(A)\right) \tag{1}$$

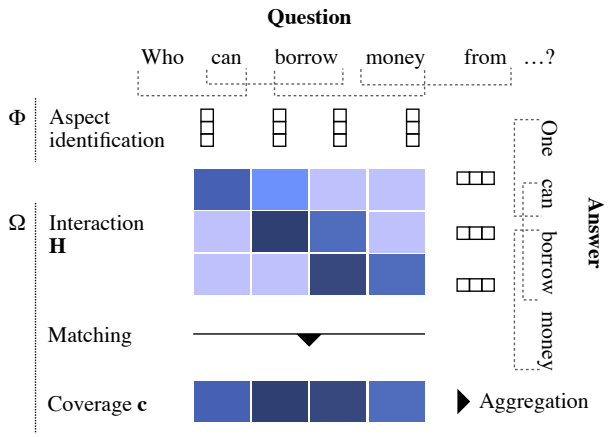

Figure 1: A simplified visualization of COALA. Dark colors visualize high interactions, i.e., larger values.

where Φ is a function that identifies aspects in Q and A (e.g., n-grams or syntactic structures), and Ω is a function that scores A based on interactions between these aspects.

In the following, we present our choices of Φ and Ω for COALA in more detail. A simplified visualization of the overall network architecture is shown in Figure 1.

Aspect Identification

There are different ways of identifying aspects in a text: aspects could be modeled as individual words, word n-grams (and representations thereof), or more linguistically aware units of the text, which take the syntax or semantic structure of the sentence into account—e.g., syntactic n-grams (Sidorov et al. 2012) or predicate-argument structures.

To capture local context in questions and answers, we define the aspects of Q and A as vector representations of all observed n-grams[2] and extract them with a convolutional operation. We apply a CNN on $\mathbf{Q} \in \mathbb{R}^{|Q| \times e}$ and $\mathbf{A} \in \mathbb{R}^{|A| \times e}$ which represent the sequence of word embeddings of Q and A, respectively (with dimensionality e):

$$\Phi(Q) = \text{CNN}(\mathbf{Q}) \tag{2}$$

$$\Phi(A) = \text{CNN}(\mathbf{A}) \tag{3}$$

where $\Phi(Q) \in \mathbb{R}^{|Q| \times d}$ and $\Phi(A) \in \mathbb{R}^{|A| \times d}$ are learned aspect representations (with d CNN filters). We share the parameters between the CNNs and use $\tanh$ activation to learn representations with positive and negative components.

By capturing local context information instead of the global context (e.g., with LSTMs) our approach needs less training data (which would be required for learning to optimally process the global context of long answers).

The extraction of aspects based on word sequences is generally advantageous because it does not require any preprocessing beyond tokenization. Extracting aspects from linguistic structures, however, could result in more informed comparisons between aspects. Thus, later, we propose a linguistically motivated extension of this approach.

[2]We use $n = 2$, i.e., bigrams. Preliminary experiments showed that bigrams usually achieve the best results with a slight improvement over trigrams and a significant improvement over unigrams.

Relevance Matching

We determine the matching of question aspects by the answer with three steps: (1) modeling the interaction between question and answer aspects, (2) determining the matching of question aspects by the answer, and (3) inferring the final score by aggregation.

Interactions We compute the dot product to capture the interactions between all aspects of Q and A in the interaction matrix $\mathbf{H}$. This does not introduce additional parameters to the network and has proved to be successful in other domains before (Cui et al. 2017).

$$\mathbf{H} = \Phi(Q)\Phi(A)^\intercal \tag{4}$$

Here, the value of the ith row and the jth column in $\mathbf{H}$ indicates the similarity of aspect i of Q to aspect j of A.

Aspect Matching We now determine how well the ith aspect of the question is covered by all aspects of the answer by selecting the maximum of each row in $\mathbf{H}$:

$$[\mathbf{c}]_i = \max_j \left([\mathbf{H}]_{i,j}\right) \tag{5}$$

This greatly simplifies the aggregation because we now deal with a vector instead of a matrix.

It is worth mentioning that—unlike in (Wang and Jiang 2017)—our aggregation function is now fully independent of the answer's complexity (and length) as we only consider the *best* match of a question aspect by all answer aspects. Furthermore, the aggregation determines how well all *question aspects* are covered (in contrast to aggregating how well the aspects of the answer are related to some aspects of the question). This better tests how well the whole content of the question is addressed by the answer.

Aggregation We finally infer a score with an aggregation function g that summarizes the sequence $\mathbf{c}$:

$$\Omega = g\left(\mathbf{c}\right) \tag{6}$$

To keep our approach conceptually and computationally simple, and to not introduce additional network parameters, we summarize the values in $\mathbf{c}$ with the arithmetic mean:

$$g(\mathbf{c}) = \frac{1}{|\mathbf{c}|} \sum_{i=1\ldots|\mathbf{c}|} [\mathbf{c}]_i \tag{7}$$

With these operations COALA contains only a small number of parameters and has a shallow network structure. Both can be advantageous for long answer selection and low-resource scenarios. To further improve the aggregation and to include syntactic information, in the following we propose two extensions to this approach.

Power Means for Aggregation

To extract more descriptive statistics, we use the *power mean* (Hardy, Littlewood, and Pólya 1952) defined as:

$$power\text{-}mean(\mathbf{x}, p) = \left(\frac{[\mathbf{x}]_1^p + \cdots + [\mathbf{x}]_n^p}{n} \right)^{1/p} \quad (8)$$

where $\mathbf{x} \in \mathbb{R}^n$ and $p \in \mathbb{R} \cup \{\pm\infty\}$.

A unique attribute of the power mean is its ability to retrieve well-known means that summarize different properties of a sequence, e.g., the arithmetic mean ($p = 1$), the geometric mean ($p = 0$) or the harmonic mean ($p = -1$). Recently, power means have been applied in the context of sentence embeddings to improve upon the average word embedding baseline (Rücklé et al. 2018). In contrast, we use power means to extract a number of complementary summaries from the question aspect coverage (i.e., from the vector $\mathbf{c}$).

In our extended approach COALA p-means we replace the arithmetic mean with m different power means, where we *learn* the values for p as part of the network (initialized with 1.0, i.e., the arithmetic mean). To infer a final score from the summaries, we use two feedforward layers.[3]

Learning the values for p has the unique advantage that we are not required to pre-define different power means here. Instead, the network itself learns the best operations for summarization. To the best of our knowledge, we are the first to learn power means for aggregation in a neural network.

Syntax-Aware Aspects

In our extended approach COALA syntax-aware, we add structured input to the network by obtaining (enhanced) dependency parse trees (Schuster and Manning 2016). We incorporate this information in the aspect identification layer, i.e., in addition to standard word embeddings we also feed syntactic embeddings of the words to the CNN. Here our syntactic embeddings are dense vector representations of each word's dependency relation, and the embeddings are learned as part of the network. As a result, COALA syntax-aware now learns to identify syntax-aware aspect representations.

With this approach, similar words (and n-grams) with different dependency relations are represented by different syntactic embeddings, whereas different words with the same dependency relation share common components.

More complex extensions of this approach are possible: we can identify aspects via connections in the dependency tree, or learn the importance of each aspect based on the syntactic roles of its containing words. However, based on our preliminary experiments, we find that our proposed extension through syntactic embeddings is the most effective of such integrations because the overall network is less affected by parsing errors.

[3]We use relu as activation for the first layer and sigmoid function for the output layer (to ensure that it is in $[0, 1]$). The number of hidden units is equal to m (number of power means).

Dataset	Number of Questions			Answer
	Train	Valid	Test	Length
Benchmarks				
InsuranceQA	12,889	1,592	1,625	112
WikiPassageQA	3,332	417	416	153
StackExchange (SE)				
Travel	3,572	765	766	214
Cooking	3,692	791	792	189
Academia	2,856	612	612	229
Apple	5,831	1,249	1,250	114
Aviation	3,035	650	652	281

Table 1: Dataset statistics. The answer length is the average number of tokens in an answer.

Experimental Setup

Data

We evaluate our approaches on several different datasets that cover a broad spectrum of domains for cQA answer selection. An overview of the datasets is given in Table 1.

InsuranceQA is a well-known answer selection benchmark and was introduced in (Feng et al. 2015). We use the most recent version (v2) in which candidate answers are retrieved with a search engine (using the question as a query). WikiPassageQA (Cohen, Yang, and Croft 2018) is a recent benchmark for passage retrieval where queries are non-factoid questions and relevant passages are paragraphs from Wikipedia. Even though this dataset does not contain cQA data, it models a related scenario.

For a more thorough evaluation we also obtain data from travel, cooking, academia, apple (computer), and aviation communities of StackExchange and create datasets that reflect real-life cQA scenarios. For a given question we retrieve similar questions in the dataset and use their accepted answers as candidate answers to the initial question.[4] The accepted answer of the initial question (and the accepted answers of the question's duplicates) are labeled as correct answers.

As we can see in Table 1, one of the distinguishing differences is the length of the answers in our different domains. However, all datasets contain long multi-sentence answer texts (e.g., explanations) which is different to classical QA.

Models and Baselines

We compare our approaches against a number of strong baselines and the recent state of the art:

(1) *IR baselines*: TF*IDF and BM25 are generally considered as strong baselines in both cQA (Lei et al. 2016) and passage retrieval (Cohen, Yang, and Croft 2018).[5]

(2) *Semantic similarity*: state-of-the-art approaches compare learned semantic representations of questions and an-

[4]We use the title of the question and discard the detailed description in the question body. We use ElasticSearch with BM25 to retrieve 100 similar questions.

[5]We use the gensim implementation of BM25 and the sklearn implementation of TF*IDF. We use NLTK's Porter stemmer to preprocess the texts.

Model	$\sum$	InsuranceQA	Travel	Cooking	Academia	Apple	Aviation	WikiPassageQA
Unsupervised IR Baselines								
BM25	30.3	24.9	38.1	30.9	29.2	21.8	37.0	53.00 / 61.71
TF*IDF	32.4	18.7	39.9	35.1	32.2	26.7	41.9	39.92 / 46.38
Semantic Similarity Methods								
InferSent	23.0	14.8	27.0	21.3	22.5	22.8	29.3	43.62 / 50.53
p-mean Embeddings	25.7	17.0	32.1	29.3	24.3	19.6	31.7	42.82 / 50.44
CNN	25.9	24.4	36.9	25.9	22.5	20.2	25.3	27.33 / 31.48
BiLSTM	34.8	32.4	45.3	35.2	31.5	27.2	37.3	46.16 / 52.89
Att.-BiLSTM	34.5	37.9	43.0	36.2	31.2	24.7	33.9	47.04 / 54.36
AP-BiLSTM	31.3	31.9	38.8	32.2	27.3	22.9	34.5	46.98 / 55.20
LW-BiLSTM	34.1	36.9	43.2	32.3	30.2	23.4	38.5	47.56 / 54.33
Relevance Matching Methods								
Bigrams	18.3	19.4	19.3	16.7	19.8	13.0	21.5	39.84 / 47.55
CA-Wang	39.1	37.0	46.5	39.4	36.1	29.2	46.5	48.71 / 56.11
COALA	43.6	38.0	53.8	47.3	42.2	32.0	48.4	**60.58 / 69.40**
COALA p-means	**45.2**	**39.9**	53.4	46.5	**44.2**	**34.5**	**52.9**	59.29 / 68.48
COALA syntax-aware	44.3	39.5	**54.1**	**47.8**	43.5	32.7	48.3	60.48 / 68.75

Table 2: Accuracies of the different models on the cQA datasets and MAP/MRR on WikiPassageQA. Σ denotes the average accuracy over all cQA datasets.

swers with cosine similarity. We evaluate AP-BiLSTM (Dos Santos et al. 2016), Attentive-BiLSTM (Tan et al. 2016), and LW-BiLSTM (Rücklé and Gurevych 2017). Further, we also test the standard CNN and BiLSTM models.

Apart from supervised semantic similarity approaches, we also evaluate *universal sentence embeddings*. To score a candidate answer, we embed the question sentence and all answer sentences and compute the maximum cosine similarity between the question embedding and all answer sentence embeddings.[6] We test two recent models: supervised InferSent (Conneau et al. 2017) and unsupervised p-mean Embeddings (Rücklé et al. 2018).

(3) *Relevance matching methods*: We implement CA-Wang, which is the compare-aggregate architecture proposed by Wang and Jiang (2017) that constitutes the current state of the art on different text matching tasks. Since we use bigrams for extracting aspects, we also include a simple bigram model among our baselines where we count the number of bigrams of the question that appear in the answer.

Finally, we evaluate COALA and its extensions with power mean aggregation (COALA p-means) and syntax-aware aspects (COALA syntax-aware).

Training Procedure

To train the semantic similarity approaches we replicate the setup of Tan et al. (2016). Here we use triples of (question, answer, incorrect candidate) and train models with the max-margin hinge loss. During training, we obtain triples by randomly sampling 50 (incorrect) candidate answers and choosing the one with the highest similarity to the question according to the current trained model.

We train all other approaches on triples of (question, candidate answer, label), where the label is a binary class (correct/incorrect answer). For each question/answer pair, we sample one corresponding pair of question/incorrect answer during the training with the same method as described above. Here we minimize the cross-entropy loss.

For all approaches, we use SGD with Adam.

Neural Network Setup

We performed a random search for the hyperparameters of all models. This included the number of CNN filters, learning rate, batch size, and dropout rate. Random search for a model and dataset was stopped after 48 hours. We evaluate models with the hyperparameters that achieved the best validation score (values are given in our source code).

All models use 300d pre-trained GloVe embeddings.

Experiments

Results

We compare all approaches across the InsuranceQA and WikiPassageQA benchmarks as well as the five StackExchange datasets in Table 2. For the cQA answer selection datasets we measure the accuracy, which is the ratio of correctly selected answers, and for the passage retrieval in WikiPassageQA we report MAP/MRR.

The results show that COALA substantially outperforms all other relevance matching and semantic similarity approaches on all seven datasets. For instance, on the cQA datasets COALA improves by 4.5pp over CA-Wang and by 8.8pp over the best semantic similarity method on average.

[6] In preliminary experiments we found that this technique outperforms comparisons between the question embedding and the average over all answer sentence embeddings.

Our extended approach COALA p-means improves the performance of COALA on these datasets by an additional 1.6pp. The proposed power mean aggregation achieves a strong improvement on four datasets and results in a small performance decrease in the remaining three cases. This shows that it is often beneficial to capture more information during aggregation whereas for individual datasets the standard arithmetic mean can already sufficiently capture the most important information.[7]

The results of InferSent and p-mean Embeddings on the other hand show that universal sentence embeddings do not perform well in cQA answer selection, which indicates that the task requires more information beyond semantic similarity. This assumption is further supported by the results of the supervised semantic similarity approaches, which only perform slightly better than the IR baselines (on average).

Finally, on the passage retrieval dataset WikiPassageQA, COALA also achieves new state-of-the-art results with an improvement of 3.21 MAP and 1.48 MRR over the best reported results in (Cohen, Yang, and Croft 2018), which they obtained with a complex model, named Memory-LSTM-CNN-TF. This demonstrates that COALA can also serve as a strong baseline for other tasks in NLP and IR that deal with the retrieval of long texts.

Syntax-Aware Aspects

The results in Table 2 show that our proposed syntax-aware extension COALA syntax-aware, which incorporates syntactic roles of word sequences to learn syntax-aware aspect representations, improves the results in five out of seven cases. It thereby achieves an an average improvement of 0.7pp over COALA in our cQA datasets. Although the improvement is lower than the one obtained with power means on average, it is more consistent across datasets.[8] This suggests that while texts in the cQA domain are not necessarily grammatically well formed and the syntax information can be noisy, the integration of syntactic structure still leads to more informed decisions. This is consistent with the findings of previous work in cQA that used non-neural approaches that relied on structural information (Tymoshenko, Bonadiman, and Moschitti 2016).

Low-Resource cQA Answer Selection

While COALA achieves good results on a number of datasets, its network structure is shallow. All parameters are within the CNN for aspect identification. This can be beneficial when we only have access to a small number of training samples, e.g., in low-resource scenarios and within small-scale cQA platforms. This is an important practical scenario because there exist a large number of more specialized—and thus smaller—cQA platforms. For example, StackExchange contains 35 sites with less than 2,000 questions.

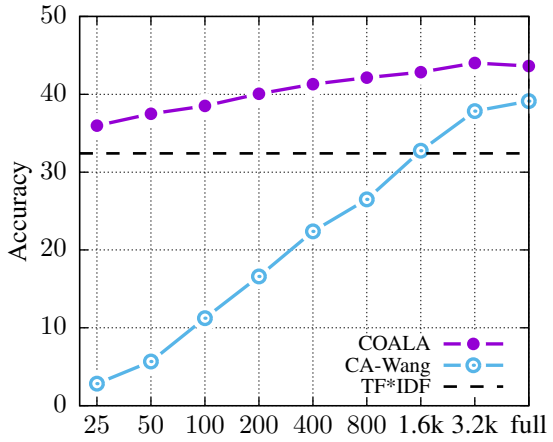

Figure 2: Model accuracies as a function of available training questions (averaged over the six cQA datasets).

To test and compare the effectiveness of our approach in such scenarios, we train COALA and CA-Wang on the cQA datasets from Table 2 with a reduced number of 25, 50, 100, ..., 3.2k training questions.[9]

Figure 2 contains the averaged accuracies over all six cQA datasets. The results show that COALA already performs better than the unsupervised baseline with 25 training questions. This is notable given that CA-Wang needs at least 1.6k training questions to achieve similar results. At the same time CA-Wang has a much steeper learning curve, which is, however, the expected behavior for a deep network and due to its lower initial performance. When both approaches are trained on the full datasets the learning curve finally flattens.

This demonstrates that COALA can be applied to a wide variety of different scenarios, e.g., to small-scale cQA platforms where only few questions exist. This is often the case for highly specialized cQA platforms and even more so for non-English platforms. Even if there exist no labeled question/answer pairs, it is still possible to use our approach because the manual annotation of 25 examples would suffice to train a good model.

Analysis

Answer Length

In Figure 3 we report the average accuracy over all cQA datasets for COALA, CA-Wang, and TF*IDF as a function of the length of the correct answers.

Here we observe that COALA performs better especially for very long answers. Our approach achieves an average accuracy of 57% for questions with correct answers that are longer than 250 words, which is substantially higher than the accuracy of 36% for CA-Wang. More importantly, we

[7] We also tested feeding $\mathbf{c}$ into a standard MLP without p-mean aggregation. Here we observed substantially decreased performances in most datasets. Therefore, the improvement of COALA p-means vs. COALA is not just due to the added complexity.

[8] We also experiment with a combination of the syntax-aware and power mean extensions, but the combined approach did not improve upon the power mean extension on average.

[9] For experiments with less than 200 questions we average over five runs with different random network initialization and for the rest we average over three runs.

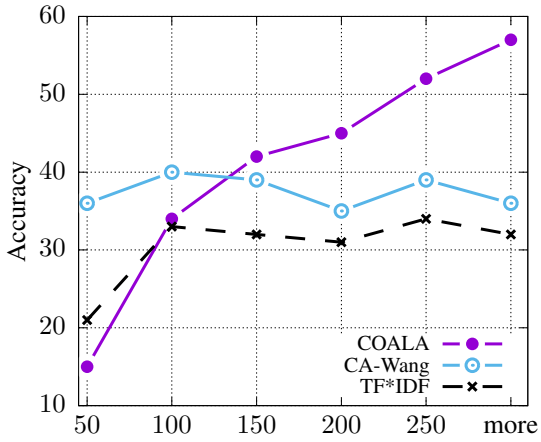

Figure 3: Model accuracies as a function of the length of the correct answers (averaged over the six cQA datasets). We group the results into six buckets: results for answers of length 0–50 (shown as "50"), 50–100 ("100"), . . . , and more than 250 ("more").

observe a steady increase in COALA's performance as the answer length increases. This is due to the coverage-based network architecture of our approach: longer answers are likely to contain more aspects of the question and COALA is able to retrieve all related aspects independent of the answer's complexity. CA-Wang on the other hand needs to process the full answer context, which is more difficult for long answers.

For answers that are shorter than 100 words, CA-Wang is more effective than COALA. This indicates that processing the full answer context is beneficial (and possible) in these cases. To further test this we also evaluated COALA on WikiQA (Yang, Yih, and Meek 2015), which is a well-known benchmark for factoid *short* answer selection (with an average answer length of 25 words). The results of COALA (69.7 MRR) are on the same level as strong semantic similarity methods such as AP-LSTM (Dos Santos et al. 2016) but below CA-Wang (75.5 MRR) as in (Wang and Jiang 2017).

This especially highlights that different tasks with different answer lengths, in particular cQA answer selection and factoid answer selection, require fundamentally different approaches to obtain optimal results.

Error Analysis

In most cases when COALA selects an incorrect candidate answer, the text either covers all aspects of the question or it covers more aspects compared to the correct answer. The aspects of the question then typically appear individually in the answer but within different contexts. The following question gives an example:

Does car insurance improve credit?

Here, COALA selects an incorrect candidate answer that covers all important aspects, i.e., car insurance and improving a credit score:

Bad credit can have an impact on the premium rates you are asked to pay for car insurance when first applying. Many auto insurers utilize credit scores to make underwriting decisions on new applications. If you have bad credit take steps to improve your score. Shop around for auto insurance from companies that use more traditional data in pricing policies. Or stick with your current carrier and drive safely. Of course you want to drive safely no matter what!

The contexts of the question and answer aspects are different and thus the answer should not be selected.

Such errors occur because COALA does not utilize the global answer context. On the other hand, models that do so often fail to properly recognize the coverage of aspects when the answers are long.

Overall, this suggest that it could be beneficial to combine COALA with other, context-aware approaches. For example, this could be done in a two-step ranking process in which COALA first selects a number of candidate answers that cover most aspects of the question, and afterwards a context-aware model chooses the answer that refers to these covered aspects in the right context. Alternatively, one could also use standard ensemble learning techniques. In both cases, however, it would be necessary to ensure that the global answer context is utilized correctly—e.g., only in cases where COALA finds multiple answers that cover all or most question aspects.

Conclusion

We proposed COALA, an efficient relevance matching approach for cQA answer selection based on the compare-aggregate framework with three important attributes: (1) our approach scales well to long answers—which are common in cQA—and outperforms the recent state of the art on six cQA datasets from different domains by a large margin; (2) it generalizes well from (very) small data and outperforms different unsupervised baselines already when trained with 25 questions—it is therefore suitable for low-resource scenarios, i.e., our approach can be applied to a large number of small-scale cQA platforms; (3) COALA can be efficiently enhanced, e.g., by incorporating syntactic information in the input layer and by using learned power means during aggregation. Both extensions lead to gains in our experiments.

In addition, our approach is not specific to cQA: it achieves state-of-the-art results on the passage retrieval benchmark dataset WikiPassageQA. It can therefore serve as a strong baseline for other tasks that rank or retrieve long texts.

Finally, our analysis revealed fundamental differences between the model's capabilities to deal with answers of different lengths. Whereas COALA can deal with long answers especially well, context-aware approaches are better suited to handle very short answers. This not only shows that different scenarios—e.g., cQA and factoid answer selection—require different types of approaches to achieve optimal results, but also that both types of approaches could be combined in future work to achieve improvements in both cases.

Our source code and data are publicly available.[10]

[10]https://github.com/UKPLab/aaai2019-coala-cqa-answer-selection

Acknowledgements

This work has been supported by the German Research Foundation (DFG) as part of the QA-EduInf project (grant GU 798/18-1 and grant RI 803/12-1) and by the DFG-funded research training group "Adaptive Preparation of Information form Heterogeneous Sources" (AIPHES, GRK 1994/1).

References

Bao, J.; Duan, N.; Zhou, M.; and Zhao, T. 2014. Knowledge-Based Question Answering as Machine Translation. In *ACL*.

Cohen, D.; Yang, L.; and Croft, B. W. 2018. WikiPassageQA: A Benchmark Collection for Research on Non-factoid Answer Passage Retrieval. In *SIGIR*, 1165–1168.

Conneau, A.; Kiela, D.; Schwenk, H.; Barrault, L.; and Bordes, A. 2017. Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. In *EMNLP*, 681–691.

Cui, Y.; Chen, Z.; Wei, S.; Wang, S.; Liu, T.; and Hu, G. 2017. Attention-over-Attention Neural Networks for Reading Comprehension. In *ACL*, 593–602.

Dos Santos, C.; Tan, M.; Xiang, B.; and Zhou, B. 2016. Attentive Pooling Networks. *arXiv preprint arXiv:1602.03609*.

Feng, M.; Xiang, B.; Glass, M. R.; Wang, L.; and Zhou, B. 2015. Applying Deep Learning to Answer Selection: A Study and an Open Task. In *ASRU*, 813–820.

Guo, J.; Fan, Y.; Ai, Q.; and Croft, B. W. 2016. A Deep Relevance Matching Model for Ad-hoc Retrieval. In *CIKM*, 55–64.

Hardy, G.; Littlewood, J.; and Pólya, G. 1952. *Inequalities*. Cambridge, England: Cambridge University Press.

He, H., and Lin, J. 2016. Pairwise Word Interaction Modeling with Deep Neural Networks for Semantic Similarity Measurement. In *NAACL*, 937–948.

Hu, B.; Lu, Z.; Li, H.; and Chen, Q. 2014. Convolutional Neural Network Architectures for Matching Natural Language Sentences. In *NIPS*, 2042–2050.

Lei, T.; Joshi, H.; Barzilay, R.; Jaakkola, T.; Tymoshenko, K.; Moschitti, A.; and Marquez, L. 2016. Semi-supervised Question Retrieval with Gated Convolutions. In *NAACL*, 1279–1289.

Lu, Z., and Li, H. 2013. A Deep Architecture for Matching Short Texts. In *NIPS*, 1367–1375.

Nakov, P.; Hoogeveen, D.; Màrquez, L.; Moschitti, A.; Mubarak, H.; Baldwin, T.; and Verspoor, K. 2017. SemEval-2017 Task 3: Community Question Answering.

Pang, L.; Lan, Y.; Guo, J.; Xu, J.; Xu, J.; and Cheng, X. 2017. DeepRank: A New Deep Architecture for Relevance Ranking in Information Retrieval. In *CIKM*, 257–266.

Parikh, A. P.; Täckström, O.; Das, D.; and Uszkoreit, J. 2016. A Decomposable Attention Model for Natural Language Inference. In *EMNLP*, 2249–2255.

Robertson, S. E., and Walker, S. 1994. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *SIGIR*, 232–241.

Romeo, S.; Da San Martino, G.; Barrón-Cedeño, A.; Moschitti, A.; Belinkov, Y.; Hsu, W.-N.; Zhang, Y.; Mohtarami, M.; and Glass, J. 2016. Neural Attention for Learning to Rank Questions in Community Question Answering. In *COLING*, 1734–1745.

Rücklé, A., and Gurevych, I. 2017. Representation Learning for Answer Selection with LSTM-Based Importance Weighting. In *IWCS*.

Rücklé, A.; Eger, S.; Peyrard, M.; and Gurevych, I. 2018. Concatenated Power Mean Embeddings as Universal Cross-Lingual Sentence Representations. *arXiv preprint arXiv:1803.01400*.

Schuster, S., and Manning, C. D. 2016. Enhanced English Universal Dependencies: An Improved Representation for Natural Language Understanding Tasks. In *LREC*.

Severyn, A., and Moschitti, A. 2015. Learning to Rank Short Text Pairs with Convolutional Deep Neural Networks. In *SIGIR*, 373–382.

Shen, Y.; Rong, W.; Sun, Z.; Ouyang, Y.; and Xiong, Z. 2015. Question / Answer Matching for CQA System via Combining Lexical and Sequential Information. In *AAAI*, 275–281.

Sidorov, G.; Velasquez, F.; Stamatatos, E.; Gelbukh, A.; and Chanona-Hernández, L. 2012. Syntactic Dependency-based N-grams as Classification Features. In *MICAI*, 1–11.

Tan, M.; Dos Santos, C.; Xiang, B.; and Zhou, B. 2016. Improved Representation Learning for Question Answer Matching. In *ACL*, 464–473.

Tay, Y.; Phan, M. C.; Tuan, L. A.; and Hui, S. C. 2017. Learning to Rank Question Answer Pairs with Holographic Dual LSTM Architecture. In *SIGIR*, 695–704.

Tymoshenko, K.; Bonadiman, D.; and Moschitti, A. 2016. Convolutional Neural Networks vs. Convolution Kernels: Feature Engineering for Answer Sentence Reranking. In *NAACL*, 1268–1278.

Verberne, S.; Boves, L.; Oostdijk, N.; and Coppen, P.-A. 2010. What Is Not in the Bag of Words for Why-QA? *Computational Linguistics* 36(2):229–245.

Wang, S., and Jiang, J. 2017. A Compare-Aggregate Model for Matching Text Sequences. *ICLR*.

Wang, S.; Yu, M.; Jiang, J.; Zhang, W.; Guo, X.; Chang, S.; Wang, Z.; Klinger, T.; Tesauro, G.; and Campbell, M. 2018. Evidence Aggregation for Answer Re-Ranking in Open-Domain Question Answering. In *ICLR*.

Wang, B.; Liu, K.; and Zhao, J. 2016. Inner Attention based Recurrent Neural Networks for Answer Selection. In *ACL*, 1288–1297.

Wang, M.; Smith, N. A.; and Mitamura, T. 2007. What is the Jeopardy Model? A Quasi-Synchronous Grammar for QA. In *EMNLP-CoNLL*, 22–32.

Yang, L.; Ai, Q.; Guo, J.; and Croft, B. W. 2016. aNMM: Ranking Short Answer Texts with Attention-Based Neural Matching Model. In *CIKM*, 287–296.

Yang, Y.; Yih, W.-t.; and Meek, C. 2015. WikiQA: A Challenge Dataset for Open-Domain Question Answering. In *EMNLP*, 2013–2018.

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

Zhang, X.; Li, S.; Sha, L.; and Wang, H. 2017. Attentive Interactive Neural Networks for Answer Selection in Community Question Answering. In *AAAI*, 3525–3521.