A Review on Deep Learning Techniques Applied to Answer Selection

Tuan Manh Lai
Adobe Research
tlai@adobe.com

Trung Bui
Adobe Research
bui@adobe.com

Sheng Li
Adobe Research
sheli@adobe.com

Abstract

Given a question and a set of candidate answers, answer selection is the task of identifying which of the candidates answers the question correctly. It is an important problem in natural language processing, with applications in many areas. Recently, many deep learning based methods have been proposed for the task. They produce impressive performance without relying on any feature engineering or expensive external resources. In this paper, we aim to provide a comprehensive review on deep learning methods applied to answer selection.

1 Introduction

Answer selection is an active research field and has drawn a lot of attention from the natural language processing community. Given a question and a set of candidate answers, the task is to identify which of the candidates contains the correct answer to the question. For example, given the question "Who established the Nobel Prize?" and the following candidate answers:

1. The Nobel Prize was established more than 100 years ago.
2. The Fields Medal, established in 1936, is often described as the Nobel Prize of mathematics.
3. The Nobel Prize was established in the will of Alfred Nobel.

The third answer should be selected. This example shows that simple word matching is not enough. Even though, all of the sentences contain the keywords "established" and "Nobel Prize", only the third sentence answers the question. From this point, we assume that each candidate answer is a sentence although the discussion is also applicable to more general case such as each candidate answer is a passage.

Answer selection is an important problem in its own right as well as in the context of open domain question answering. Although details vary from system to system, a typical open domain question answering system can be considered as consisting of: (a) question analysis (b) retrieval of potentially relevant documents (c) ranking and selecting of the most promising sentences (or more generally, passages) within the retrieved documents; and optionally d) extracting the exact natural language phrase that answers the question (Prager, 2006; Ferrucci, 2012; Sequiera et al., 2017). Figure 1 depicts a typical question answering pipeline. In this setup, answer selection can be applied to identify the sentences that are most relevant to the question within the retrieved documents. Besides its application in open domain question answering, the techniques developed for answer selection can be potentially used to predict answer quality in community question answering (cQA) sites (Nakov et al., 2015).

Previous work on answer selection typically relies on feature engineering, linguistic tools, or external resources (Wang et al., 2007; Wang and Manning, 2010; Heilman and Smith, 2010; Yih et al., 2013; Yao et al., 2013). Recently, many deep learning based methods have been proposed for the task (Bian et al., 2017; Shen et al., 2017; Tran et al., 2018). They outperform traditional techniques. In addition, they do not need any feature-engineering effort or hand-coded resources beyond some large unlabeled corpus on which to learn the initial word embeddings, such as word2vec (Mikolov et al., 2013) or GloVe (Pennington et al., 2014).

Proceedings of the 27th International Conference on Computational Linguistics, pages 2132–2144
Santa Fe, New Mexico, USA, August 20-26, 2018.

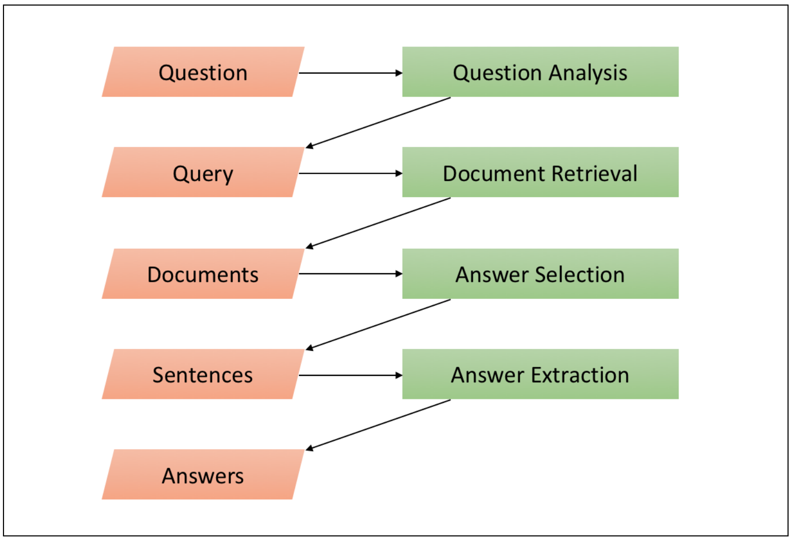

Figure 1: A typical question answering pipeline architecture, adapted from (Sequiera et al., 2017)

While previous work has recognized the increasing use of deep learning techniques in natural language processing (Young et al., 2017), no systematic survey of deep learning methods for answer selection to date has been published. Therefore, in this paper, we aim to give a comprehensive review of various deep learning methods that have been used to tackle the answer selection task.

In Section 2, we present a review of deep learning methods for answer selection. In Section 3, we examine the most popular datasets and the evaluation metrics for answer selection. We conclude this paper in Section 4 by discussing the potential future research directions.

2 Methods

Existing deep learning methods for answer selection can be examined along two dimensions: (i) learning approaches (ii) neural network architectures (Table 1).

2.1 Learning Approaches

Given a question and a set of candidate sentences, the task is to identify candidate sentences that contain the correct answer to the question. From the definition, the problem can be formulated as a ranking problem, where the goal is to give better rank to the candidate sentences that are relevant to the question. There are three most common approaches to learn the ranking function h_θ, namely, *pointwise*, *pairwise* and *listwise* (Liu, 2011).

In the *pointwise* approach, the ranking problem is transformed to a binary classification problem. More specifically, the training instances are triples (q_i, c_{ij}, y_{ij}), where q_i is a question in the dataset, c_{ij} is a candidate answer sentence for q_i, and y_{ij} is a binary value indicating whether c_{ij} contains the correct answer to q_i. It is enough to train a binary classifier: $h_\theta(q_i, c_{ij}) \to \hat{y_{ij}}$, where $0 \leq \hat{y_{ij}} \leq 1$. For example, in (Yu et al., 2014), the training objective is to minimize the cross entropy of all labelled question-candidate pairs in the training set. During inference, given a question, the trained classifier h_θ is used to rank every candidate sentence, and the top-ranked candidate is selected (i.e., $\text{argmax}_{c_{ij}}\, h_\theta(q_i, c_{ij})$ should be selected as the answer to q_i). Many work adopted this approach (Yu et al., 2014; Severyn and Moschitti, 2015; Wang et al., 2016b; Shen et al., 2017).

The second approach to ranking is the *pairwise* approach, where the ranking function h_θ is explicitly trained to score correct candidate sentences higher than incorrect sentences. Given a question, the approach takes a pair of candidate answer sentences and explicitly learns to predict which sentence is more relevant to the question. For example, in (Feng et al., 2015), the training instances are triples (q_i, c_i^+, c_i^-), where q_i is a question, c_i^+ is a correct sentence for q_i, and c_i^- is an incorrect sentence sampled from the

Method	Learning Approach	Model Architecture	MAP (Raw TrecQA)	MAP (Clean TrecQA)
TRAIN-ALL unigram+count (Yu et al., 2014)	Pointwise	Siamese	0.693	-
TRAIN-ALL bigram+count (Yu et al., 2014)	Pointwise	Siamese	0.711	-
QA-LSTM (Tan et al., 2015)	Pairwise	Siamese	-	0.682
QA-LSTM with attention (Tan et al., 2015)	Pairwise	Attentive	-	0.690
QA-LSTM/CNN (Tan et al., 2015)	Pairwise	Siamese	-	0.706
Attentive Pooling CNN (dos Santos et al., 2016)	Pairwise	Attentive	-	0.753
(Severyn and Moschitti, 2015)	Pointwise	Siamese	0.746	-
L.D.C Model (Wang et al., 2016b)	Pointwise	Compare-Aggregate	-	0.771
Pairwise Word Interaction Modelling (He and Lin, 2016)	Pointwise	Compare-Aggregate	0.758	-
Multi-Perspective CNN (He et al., 2015)	Pointwise	Siamese	0.762	0.777
HyperQA (Hyperbolic Embeddings) (Tay et al., 2018a)	Pairwise	Siamese	0.770	0.784
PairwiseRank+Multi-Perspective CNN (Rao et al., 2016)	Pairwise	Siamese	0.780	0.801
BiMPM (Shen et al., 2017)	Pointwise	Compare-Aggregate	-	0.802
Dynamic-Clip Attention (Bian et al., 2017)	Listwise	Compare-Aggregate	-	0.821
IWAN (Shen et al., 2017)	Pointwise	Compare-Aggregate	-	0.822
IWAN+CARNN (Tran et al., 2018)	Pointwise	Compare-Aggregate	-	0.829
MCAN (Tay et al., 2018b)	Pointwise	Compare-Aggregate	-	0.838

Table 1: Overview of existing deep learning methods to answer selection

whole candidate sentence space. And the hinge loss function is defined as follows:

$$L = max\{0, m - h_\theta(q_i, c_i^+) + h_\theta(q_i, c_i^-)\} \tag{1}$$

where m is the margin. If $h_\theta(q_i, c_i^+) - h_\theta(q_i, c_i^-) < m$ then L is positive. When this condition is satisfied, the implication is that the system ranks the positive sentence below the negative sentence, or does not sufficiently rank the positive answer above the negative answer. On the other hand, if the correct sentence has a score higher than the incorrect sentence by at least a margin m (i.e., $h_\theta(q_i, c_i^+) - h_\theta(q_i, c_i^-) \geq m$), and then the above expression has zero loss. In summary, the loss function is designed to encourage the correct answer to have a higher score than the incorrect answer by a certain margin. Similar to the *pointwise* approach, during testing, the candidate answer with the largest score is selected. (Tan et al., 2015; Yang et al., 2016; dos Santos et al., 2016; Tay et al., 2018a) also used the pairwise hinge loss function above.

The third method is the *listwise* approach (Cao et al., 2007). The *pointwise* approach and the *pairwise* approach ignore the fact that answer selection is a prediction task on list of candidate sentences. In the *listwise* approach, a single instance consists of a question and its list of candidates. For example, (Bian et al., 2017) adopted the approach. Concretely, during training, given a question q_i and its list of candidate sentences **C** $\{c_{i1}, c_{i2}, ..., c_{im}\}$ and the ground truth labels **Y** $\{y_{i1}, y_{i2}, ..., y_{im}\}$, the normalized score vector **S** is calculated as follows:

$$\begin{aligned} &Score_j = h_\theta(q_i, c_{ij}) \\ &\mathbf{S} = softmax([Score_1, Score_2, ..., Score_m]) \end{aligned} \tag{2}$$

Target labels also need to be normalized:

$$\mathbf{Y} = \frac{\mathbf{Y}}{\sum_{j=1}^{m} y_{ij}} \tag{3}$$

And the objective is to minimize the KL-divergence of **S** and **Y**.

Even though many work (Yu et al., 2014; Severyn and Moschitti, 2015; Wang et al., 2016b; Shen et al., 2017) adopted the *pointwise* approach, this approach is not close to the nature of ranking. The *pairwise* approach and the *listwise* approach exploit more information about the ground truth ordering of candidate sentences. (Rao et al., 2016) proposed a *pairwise* ranking approach that can directly exploit existing *pointwise* neural network models as base components. The approach outperforms many competitive *pointwise* baselines. (Bian et al., 2017) showed that the *listwise* approach performs better than the *pointwise* approach on public datasets such as TrecQA (Wang et al., 2007) and WikiQA (Yang et al., 2015). In the next section, we describe various neural network architectures for modeling the ranking function h_θ, which takes a question-candidate pair and returns a score indicating whether the candidate is relevant to the question.

2.2 Neural Network Architectures

There are three main types of general architectures for measuring the relevance of a candidate sentence to a question.

- **Siamese Architecture.** In a Siamese architecture (Bromley et al., 1993), the same encoder (e.g., a CNN or a RNN) is used to build the vector representations for the input sentences (i.e., the candidate answer and the question) individually. After that, the relevance score is determined solely based on the encoded vectors. There is no explicit interaction between the input sentences during the encoding process.
- **Attentive Architecture.** Instead of generating representations for the candidate answer and the question independently, attention mechanisms (Bahdanau et al., 2014; Hermann et al., 2015; Luong et al., 2015) can be used to allow the information from an input sentence to influence the computation of the other's representation (Tan et al., 2015; dos Santos et al., 2016). Even though the weakness of the Siamese models is alleviated, the interaction between the input sentences during the encoding process is still minimal in most Attentive architectures.
- **Compare-Aggregate Architecture.** The Compare-Aggregate architectures can capture more interactive features between input sentences than the Siamese architectures and the Attentive architectures, therefore typically have better performance when evaluated on public datasets such as TrecQA (Wang et al., 2007) and WikiQA (Yang et al., 2015). In a Compare-Aggregate architecture, vector representations of small units such as words of the sentences are first compared. After that, these comparison results are aggregated to calculate the final relevance score.

It is worth mentioning that the boundaries between the architecture types are not always crystal clear. For example, while many Siamese architectures typically capture less interactive features between the input sentences than the Attentive architectures, few recently proposed Siamese architectures have sophisticated comparison layer after the encoding layer (He et al., 2015; Rao et al., 2016). As a result, they even outperform some Attentive architectures. Even though the boundaries are not crystal clear, separating the existing different neural architectures into the three categories can provide the big picture more easily.

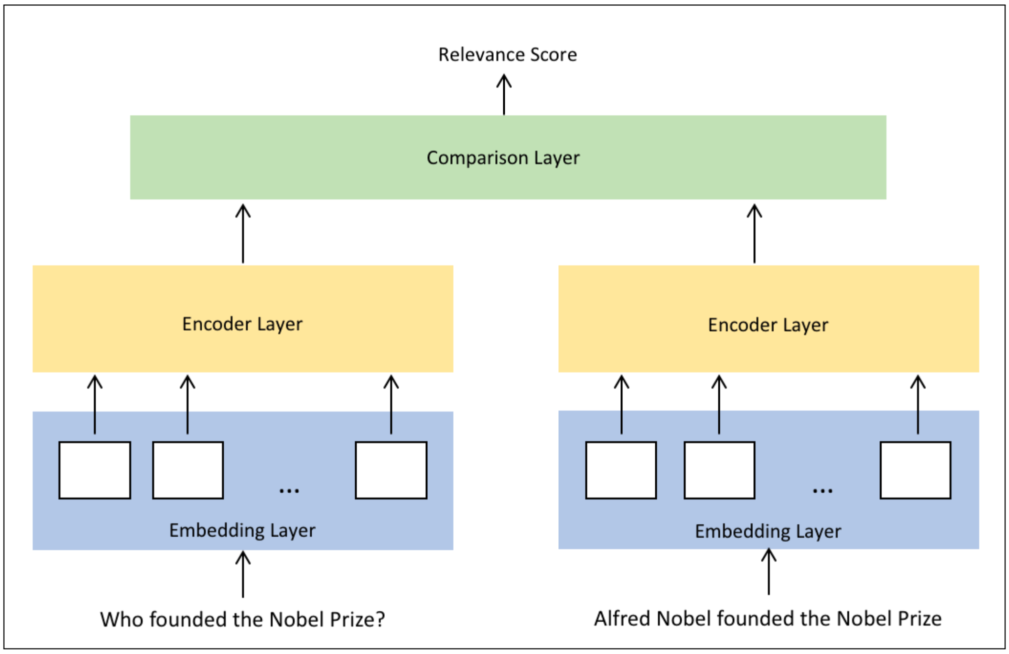

Figure 2: The general architecture of a Siamese model. The same encoder is used to generate the vector representations for the input sentences.

2.2.1 Siamese Architecture

Siamese neural networks have been proposed for a number of sentence pair modeling tasks, including semantic similarity (Mueller and Thyagarajan, 2016), paraphrase identification (Hu et al., 2014), and natural language inference (Conneau et al., 2017). Figure 2 shows the general architecture of a Siamese model. The vector representations of the input sentences are built separately by the encoder. Two input sentences have no influence on the computation of each other's representation. After that, the encoded vectors are compared using measures such as cosine similarity (Feng et al., 2015; Yang et al., 2015), element-wise operations (Tai et al., 2015), or neural network-based combination (Bowman et al., 2015). An advantage of this architecture is that applying the same encoder to each input sentence makes the model smaller. In addition, the sentence vectors can be used for visualization, sentence clustering and many other purposes (Wang et al., 2016a).

One of the first attempts at applying deep learning to answer selection was the bag-of-words model proposed by (Yu et al., 2014). The model generates the vector representation of a sentence by simply taking the average of all the word vectors in the sentence - having previously removed all the stop words. Integrating additional overlapping word count features with the model achieves performance better than many traditional techniques that require large numbers of hand-crafted features or external resources. Tan et al. (2015) proposed the QA-LSTM model that employs a bidirectional long short-term memory (biLSTM) network (Hochreiter and Schmidhuber, 1997) and a pooling layer to construct distributed vector representations of the input sentences independently. Then the model utilizes cosine similarity to measure the distance of the sentence representations. Severyn and Moschitti (2015) proposed a model that employs a convolutional neural network (CNN) to generate the representations of the input sentences. The CNN is based on an architecture that has previously been applied to many sentence classification tasks (Kalchbrenner et al., 2014; Kim, 2014). In (He et al., 2015), each input sentence is modeled using a CNN that extracts features at multiple levels of granularity and uses multiple types of pooling. The representations of the input sentences are then compared at several granularities using multiple similarity metrics. Finally, the comparison results are fed into a fully connected layer to obtain the final relevance score. The proposed model outperforms many other Siamese models. Tay et al. (2018a) proposed a simple but novel deep learning architecture that models the relationship between a question and a candidate sentence in Hyperbolic space instead of Euclidean space. It achieves highly competitive performance without employing any sophisticated neural encoder such as LSTM or CNN. Vector representations of the input sentences are generated independently in a bag-of-words manner.

2.2.2 Attentive Architecture

In a Siamese architecture, the input sentences are first encoded into fixed-length vector representations separately, and these representations are then compared. Despite its conceptual simplicity, a disadvantage is the absence of explicit interaction between the input sentences during the encoding process. A question is always mapped to the same vector regardless of the candidate answer in consideration, and vice versa. Attention mechanisms have been applied to alleviate the weakness. Tan et al. (2015) extended the basic QA-LSTM model with attention (Figure 3). The model employs a biLSTM network and a pooling layer to generate the question representation $\mathbf{o}_q$. The candidate representation $\mathbf{o}_c$ is calculated similarly, except that prior to the pooling layer, each biLSTM output vector will be multiplied by a weight, which is determined by the question representation $\mathbf{o}_q$. Conceptually, the attention mechanism gives more weights on certain words in the candidate answer, and the weights are computed according to the question information. In this case, the attention mechanism is performed only in a single direction. dos Santos et al. (2016) proposed a two-way attention mechanism called Attentive Pooling (AP). AP allows the information from the two input sentences to influence the computation of each other's representation.

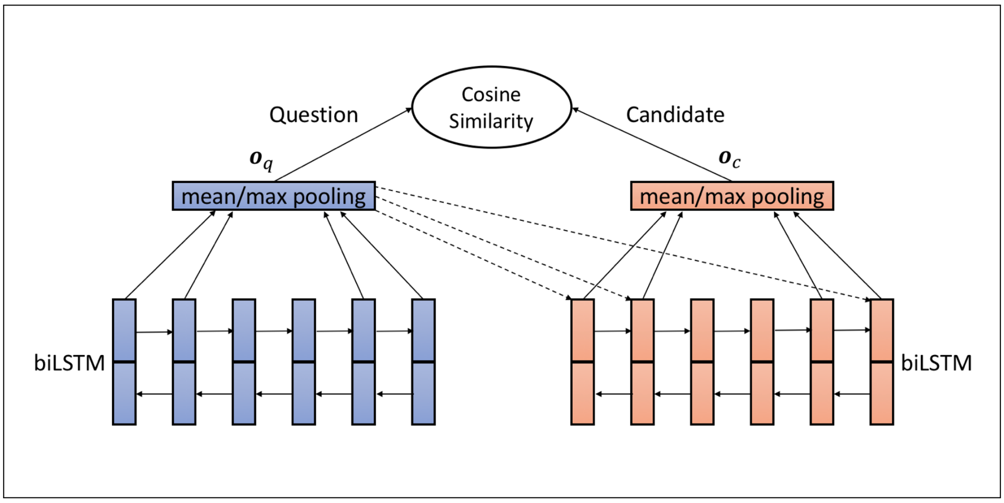

Figure 3: QA-LSTM with attention (figure adapted from (Tan et al., 2015))

2.2.3 Compare-Aggregate Architecture

A common trait of a number of recent state-of-the-art methods for answer selection is the use of the Compare-Aggregate architecture (Wang and Jiang, 2016; Wang et al., 2017; Bian et al., 2017; Shen et al., 2017; Tran et al., 2018). Under this architecture, smaller units (such as words) of the input sentences are compared. And then the comparison results are aggregated (e.g., by a CNN or a RNN) to make the final decision. Figure 4 shows the architecture of BiMPM, a Compare-Aggregate model proposed in (Wang et al., 2017). The model consists of five layers as follows.

Word Representation Layer. The goal of this layer is to represent each word in the input sentences with a d-dimensional vector. BiMPM constructs the d-dimensional vector with two components: a character-composed embedding and a word embedding pre-trained with GloVe (Pennington et al., 2014) or word2vec (Mikolov et al., 2013).

Context Representation Layer. The goal of this layer is to obtain a new representation for each position in the input sentences that captures some contextual information in addition to the word at the position. BiMPM employs a biLSTM to generate the contextual representations.

Matching Layer. The goal of this layer is to compare each contextual representation of one sentence against all contextual representations of the other sentence. The output of this layer are two sequences of matching vectors, where each matching vector corresponds to the comparison result of one position of a sentence against all the positions of the other sentence.

Aggregation Layer. The goal of this layer is to aggregate the comparison results from the previous layer. BiMPM employs another BiLSTM to aggregate the two sequences of matching vectors into fixed-length vectors.

Prediction Layer. The goal of this layer is to make the final prediction. BiMPM uses a two layer feed-forward neural network to consume the fixed-length vectors from the previous layer, and apply the softmax function to get the final score.

Even though specific details vary from model to model, other Compare-Aggregate models for answer selection (Bian et al., 2017; Shen et al., 2017; Tran et al., 2018) have a similar structure to the BiMPM model. The Compare-Aggregate architectures can capture more interactive features between input sentences than the Siamese architectures and the Attentive architectures, therefore typically have better performance when evaluated on public datasets such as TrecQA (Table 1)

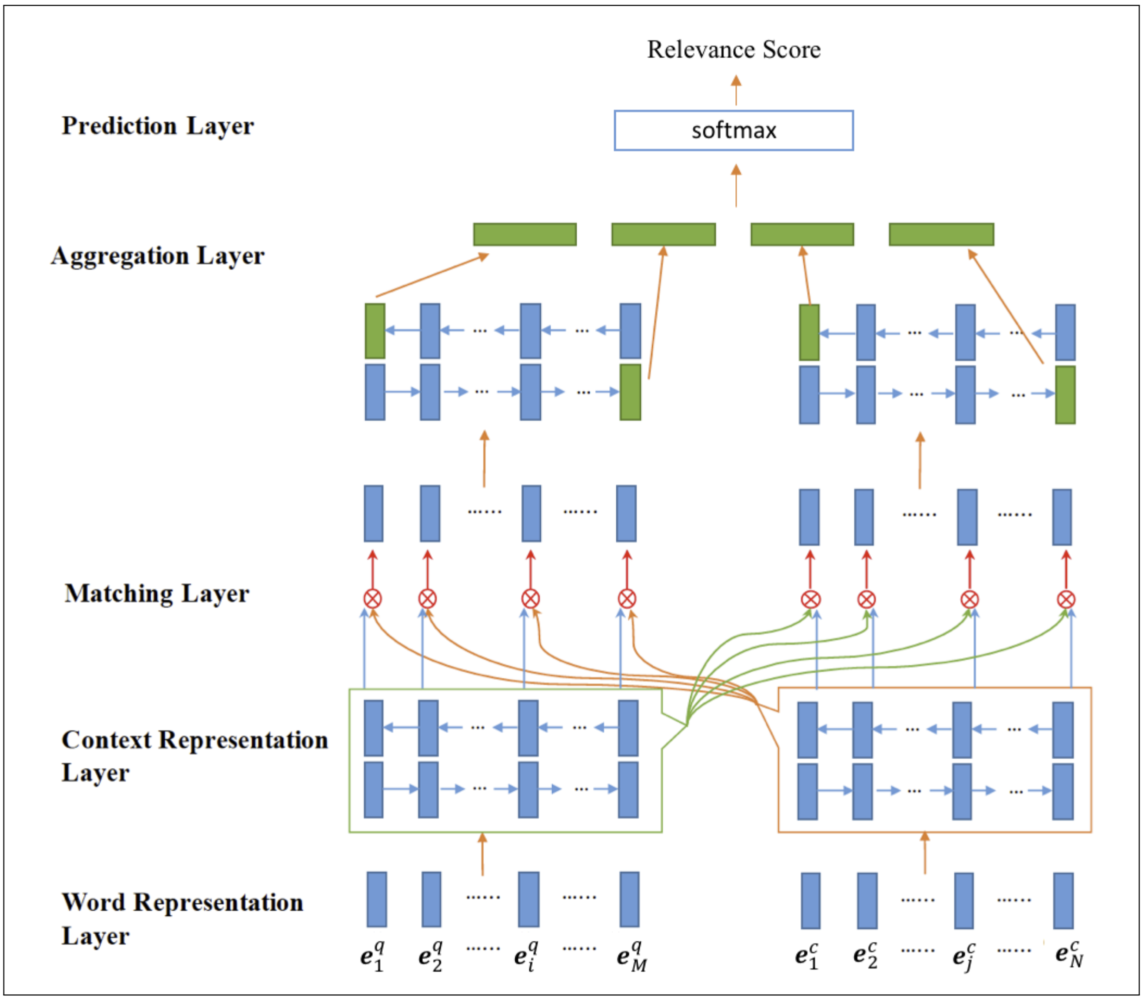
Relevance Score
softmax
Prediction Layer
Aggregation Layer
Matching Layer
Context Representation Layer
Word Representation Layer
e_1^q e_2^q ...... e_i^q ...... e_M^q
e_1^c e_2^c ...... e_j^c ...... e_N^c

Figure 4: The architecture of the BiMPM model (figure adapted from (Wang et al., 2017))

3 Datasets and Evaluation Metrics

TrecQA, WikiQA, and InsuranceQA datasets have been widely used for benchmarking answer selection systems.

- The TrecQA dataset (Wang et al., 2007) was created from the TREC Question Answering tracks. In the literature (Yih et al., 2013; Yu et al., 2014; Severyn and Moschitti, 2015; Wang et al., 2016b;

dos Santos et al., 2016), we observe two versions of TrecQA: both have the same training set but their development and test sets differ. The Raw version has 82 development questions and 100 test questions. The Clean version removed questions in development and test sets with no answers or only positive/negative answers, reducing the development and test set's sizes to 65 and 68 questions, respectively. Relevant statistics are shown in Table 2. The data provided for training come as two sets: a small set of 94 questions (TRAIN) that were manually judged and a more noisy set of 1229 questions (TRAIN-ALL) that comes with automatic judgements.

Split	# Questions	# QA pairs	% Correct
TRAIN	94	4718	7.4%
TRAIN-ALL	1229	53417	12.0%
Raw DEV	82	1148	19.3%
Raw TEST	100	1517	18.7%
Clean DEV	65	1117	18.4%
Clean TEST	68	1442	17.2%

Table 2: Statistics of the TrecQA dataset

- The WikiQA dataset (Yang et al., 2015) is an open-domain question answering dataset that was constructed from real queries of Bing and Wikipedia. Relevant statistics are shown in Table 3. There are 3047 questions and 29258 candidate answer sentences in the dataset, and 1473 sentences were labeled as correct answer sentences to their corresponding questions. In the WikiQA dataset, there are questions with only incorrect answers. All questions with no correct answers are usually removed when the dataset is used to train and evaluate answer selection systems (Yang et al., 2015; Bian et al., 2017; Shen et al., 2017). The excluded WikiQA has 873/126/243 questions and 8627/1130/2351 question-answer pairs for train/dev/test split.

	Train	Dev	Test	Total
# Questions	2118	296	633	3047
# Candidate Answers	20360	2733	6165	29258
# Correct Answers	1040	140	293	1473
# Questions w/o Correct Answers	1245	170	390	1805

Table 3: Statistics of the WikiQA dataset

- The InsuranceQA dataset (Feng et al., 2015) is a large-scale domain specific answer selection dataset in which all question and candidate pairs are in the insurance domain. The original dataset consists of four parts: train, development, test1 and test2. Relevant statistics are shown in Table 4. There could be multiple correct answers for some questions so that the number of correct answers is larger than the number of questions. The released dataset contains 24981 unique answers in total. For each question, the candidate answer pool size is set to be 500. These candidate pools were constructed by including the correct answer(s) and randomly selecting candidates from the complete set of unique answers.

Split	# Questions	# Correct Answers
TRAIN	12887	18540
DEV	1000	1454
TEST1	1800	2616
TEST2	1800	2593

Table 4: Statistics of the original InsuranceQA dataset

Community Question Answering (cQA) platforms such as Stack Overflow [1] and Yahoo Answers [2] have become an important resource of information for many Web users. A person posts a question on

[1] `http://stackoverflow.com`

[2] `https://answers.yahoo.com`

a specific topic and other users post their answers. The techniques developed for answer selection can be potentially used to improve various aspects of a cQA platform. For example, in a cQA platform, it is not unusual for a question to have hundreds of answers, the vast majority of which would not satisfy a user's information needs. Therefore, using answer selection techniques to find relevant answers can be very beneficial. In the literature, the SemEval-2016 cQA dataset (Nakov et al., 2016) has been widely used to test different answer selection systems.

- In the SemEval-2016 cQA challenge (Nakov et al., 2016), there are three subtasks for English. Subtask A (*Question-Comment Similarity*): Given a question and its first ten comments in the question thread, the goal is to rank these ten comments according to their relevance with respect to the question. Subtask B (*Question-Question Similarity*): Given a new question and the set of the first ten related questions from the forum retrieved by a search engine, the goal is to rank the related questions according to their similarity with respect to the new question. Subtask C (*Question-External Comment Similarity*): Given a new question and the set of the first ten related questions from the forum retrieved by a search engine, each associated with its first ten comments appearing in its thread, the goal is to rank the 100 comments according to their relevance with respect to the new question. The data of the three subtasks was extracted from the community-created Qatar Living Forums [3]. Answer selection techniques can be directly applied to each of the subtask. All the information related to the dataset can be found on the SemEval-2016 Task 3 website[4].

Table 5 shows examples from the datasets. Each example consists of a question with a positive answer and a negative answer. A major difference between the SemEval-2016 cQA dataset and the other answer selection datasets is the average length of a question. In WikiQA (Yang et al., 2015) and TrecQA (Wang et al., 2007), a question is typically a short sentence, while in the cQA dataset, the question body typically contains quite a few sentences.

Dataset	Example
TrecQA	**Question**: Who established the Nobel prize awards? **Positive Answer**: The Nobel Prize was established in the will of Alfred Nobel, a Swede who invented dynamite and died in 1896. **Negative Answer**: The awards aren't given in specific categories.
WikiQA	**Question**: How many albums has Eminem sold in his career? **Positive Answer**: He has sold more than 100 million records worldwide, including 42 million tracks and 49.1 million albums in the United States. **Negative Answer**: Eminem is one of the best-selling artists in the world and is the best-selling artist of the 2000s.
InsuranceQA	**Question**: Does Medicare cover my spouse? **Positive Answer**: If your spouse has worked and paid Medicare taxes for the entire required 40 quarters, or is eligible for Medicare by virtue of being disabled or some other reason, your spouse can receive his/her own medicare benefits. If your spouse has not met those qualifications, if you have met them, and if your spouse is age 65, he/she can receive Medicare based on your eligibility. **Negative Answer**: If you were married to a Medicare eligible spouse for at least 10 years, you may qualify for Medicare. If you are widowed, and have not remarried, and you were married to your spouse at least 9 months before your spouses death, you may be eligible for Medicare benefits under a spouse provision.
SemEval-2016 cQA	**Question**:Hi;Can any one tell me a place where i can have a good massage drom philipinies????? yesterday i had a massage in Bio-Bil they charged me 300qr for 01 hour bt it is totally waste... pls advice me if theres any philipinos.... **Positive Answer**: Try Magic Touch in Abu Hamour (beside Abu Hamour Petrol Stn)it will just cost you 60QR per hour and I've seen a lot of Qataris as their customers. **Negative Answer**: I dont know the name; you can call them. Do it fast; they have sooooo many reservations ;)

Table 5: Examples from different datasets

Typically, the performance of an answer selection system is measured in Mean Reciprocal Rank (MRR) and Mean Average Precision (MAP), which are standard metrics in Information Retrieval and

[3] `http://www.qatarliving.com/forum`

[4] `http://alt.qcri.org/semeval2016/task3`

Question Answering. Given a set of questions Q, MRR is calculated as follows:

$$\text{MRR} = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{\text{rank}_i} \tag{4}$$

where rank_i refers to the rank position of the first correct candidate answer for the i^{th} question. In other words, MRR is the average of the reciprocal ranks of results for the questions in Q. On the other hand, if the set of correct candidate answers for a question $q_j \in Q$ is $\{d_1, d_2, ..., d_{m_j}\}$ and R_{jk} is the set of ranked retrieval results from the top result until you get to the answer d_k, then MAP is calculated as follows:

$$\text{MAP} = \frac{1}{|Q|}\sum_{j=1}^{|Q|}\frac{1}{m_j}\sum_{k=1}^{|m_j|}\text{Precision}(R_{jk}) \tag{5}$$

When a relevant answer is not retrieved at all for a question, the precision value for that question in the above equation is taken to be 0.

Whereas MRR measures the rank of any correct answer, MAP examines the ranks of all the correct answers. In general, MRR is higher than MAP on the same list of ranked outputs, except that they are the same in the case where each question has exactly one correct answer. Table 1 shows the performance of several deep learning methods when trained and evaluated on the Raw TrecQA dataset or the Clean TrecQA dataset.

4 Discussion and Future Directions

Answer selection is an important problem in natural language processing, and many deep learning methods have been proposed for the task. In this paper, we give a comprehensive and systematic review of various deep learning methods for answer selection along two dimensions: (i) learning approaches (*pointwise*, *pairwise*, and *listwise*) (ii) neural network architectures (*Siamese architecture*, *Attentive architecture*, and *Compare-Aggregate architecture*). In addition, we examine the most popular datasets and the evaluation metrics for answer selection. Below we discuss several promising future research directions.

Transfer learning (Pan and Yang, 2010) has achieved success in domains such as speech recognition (Huang et al., 2013), computer vision (Razavian et al., 2014), and natural language processing (Zhang et al., 2017). Its applicability to question answering and answer selection has recently been studied (Min et al., 2017; Chung et al., 2017). Min et al. (2017) created SQuAD-T, a modification of the original large-scale SQuAD dataset (Rajpurkar et al., 2016) to allow for directly training and evaluating answer selection systems. Through a basic transfer learning technique from SQuAD-T, the state-of-the-art result in the WikiQA dataset can be improved. This demonstrates the potential of developing novel transfer learning techniques for the answer selection task.

Many deep learning methods for answer selection are applicable to other sentence pair modeling tasks such as natural language inference (He and Lin, 2016; Wang et al., 2017), paraphrase identification (He and Lin, 2016; Wang et al., 2017), or measuring semantic relatedness (Shen et al., 2017). Extending existing methods for answer selection to achieve state-of-the-art results on other sentence pair modeling tasks can be an interesting research direction, and vice versa.

In addition to its applications in open domain question answering and community question answering, answer selection has many other applications. Lai et al. (2018) formulated the task of answering questions related to product facts and specifications as an answer selection problem. Given a question and a set of candidate product specifications, an answer selection technique was used to identify the specification that is most relevant to the question. Applying answer selection techniques to real-world problems can be an interesting research direction. For example, answer selection techniques can be potentially used to enhance truth discovery methods (Li et al., 2017; Zhang et al., 2018).

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Weijie Bian, Si Li, Zhao Yang, Guang Chen, and Zhiqing Lin. 2017. A compare-aggregate model with dynamic-clip attention for answer selection. In *CIKM*.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *EMNLP*.

Jane Bromley, James W. Bentz, Léon Bottou, Isabelle Guyon, Yann LeCun, Cliff Moore, Eduard Säckinger, and Roopak Shah. 1993. Signature verification using a "siamese" time delay neural network. *IJPRAI*, 7:669–688.

Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to rank: From pairwise approach to listwise approach. Technical report, April.

Yu-An Chung, Hung-yi Lee, and James R. Glass. 2017. Supervised and unsupervised transfer learning for question answering. *CoRR*, abs/1711.05345.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *EMNLP*.

Cícero Nogueira dos Santos, Ming Tan, Bing Xiang, and Bowen Zhou. 2016. Attentive pooling networks. *CoRR*, abs/1602.03609.

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. *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 813–820.

David A. Ferrucci. 2012. Introduction to "this is watson". *IBM Journal of Research and Development*, 56(3):1.

Hua He and Jimmy J. Lin. 2016. Pairwise word interaction modeling with deep neural networks for semantic similarity measurement. In *HLT-NAACL*.

Hua He, Kevin Gimpel, and Jimmy J. Lin. 2015. Multi-perspective sentence similarity modeling with convolutional neural networks. In *EMNLP*.

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, Stroudsburg, PA, USA. Association for Computational Linguistics.

Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *NIPS*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9 8:1735–80.

Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2014. Convolutional neural network architectures for matching natural language sentences. In *NIPS*.

J. T. Huang, J. Li, D. Yu, L. Deng, and Y. Gong. 2013. Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 7304–7308, May.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *ACL*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *EMNLP*.

Tuan Lai, Trung Bui, Sheng Li, and Nedim Lipka. 2018. A simple end-to-end question answering model for product information. In *ECONLP@ACL*.

Yaliang Li, Nan Du, Chaochun Liu, Yusheng Xie, Wei Fan, Qi Li, Jing Gao, and Huan Sun. 2017. Reliable medical diagnosis from crowdsourcing: Discover trustworthy answers from non-experts. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 253–261. ACM.

Tie-Yan Liu. 2011. *Learning to Rank for Information Retrieval*. 01.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *EMNLP*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, pages 3111–3119, USA. Curran Associates Inc.

Sewon Min, Min Joon Seo, and Hannaneh Hajishirzi. 2017. Question answering through transfer learning from large fine-grained supervision data. In *ACL*.

Jonas Mueller and Aditya Thyagarajan. 2016. Siamese recurrent architectures for learning sentence similarity. In *AAAI*.

Preslav Nakov, Lluís Màrquez i Villodre, Walid Magdy, Alessandro Moschitti, Jim Glass, and Bilal Randeree. 2015. Semeval-2015 task 3: Answer selection in community question answering. In *SemEval@NAACL-HLT*.

Preslav Nakov, Lluís Màrquez i Villodre, Alessandro Moschitti, Walid Magdy, Hamdy Mubarak, Abed Alhakim Freihat, Jim Glass, and Bilal Randeree. 2016. Semeval-2016 task 3: Community question answering. In *SemEval@NAACL-HLT*.

S. J. Pan and Q. Yang. 2010. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, Oct.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*.

John M. Prager. 2006. Open-domain question-answering. *Foundations and Trends in Information Retrieval*, 1(2):91–231.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. In *EMNLP*.

Jinfeng Rao, Hua He, and Julie Qiaojin Lin. 2016. Noise-contrastive estimation for answer selection with deep neural networks. In *CIKM*.

Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. 2014. Cnn features off-the-shelf: An astounding baseline for recognition. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, CVPRW '14, pages 512–519, Washington, DC, USA. IEEE Computer Society.

Royal Sequiera, Gaurav Baruah, Zhucheng Tu, Salman Mohammed, Jinfeng Rao, Haotian Zhang, and Jimmy J. Lin. 2017. Exploring the effectiveness of convolutional neural networks for answer selection in end-to-end question answering. *CoRR*, abs/1707.07804.

Aliaksei Severyn and Alessandro Moschitti. 2015. Learning to rank short text pairs with convolutional deep neural networks. In *SIGIR*.

Gehui Shen, Yunlun Yang, and Zhi-Hong Deng. 2017. Inter-weighted alignment network for sentence pair modeling. In *EMNLP*.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *ACL*.

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

Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2018a. Hyperbolic representation learning for fast and efficient neural question answering. In *WSDM*.

Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2018b. Multi-cast attention networks. In *KDD*.

Quan H. Tran, Tuan Lai, Ingrid Zukerman, Gholamreza Haffari, Trung Bui, and Hung Bui. 2018. The context-dependent additive recurrent neural net. In *North American Chapter of the Association for Computational Linguistics (NAACL)*.

Shuohang Wang and Jing Jiang. 2016. A compare-aggregate model for matching text sequences. *CoRR*, abs/1611.01747.

Mengqiu Wang and Christopher D. 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 '10, pages 1164–1172, Stroudsburg, PA, USA. Association for Computational Linguistics.

Mengqiu Wang, Noah A. Smith, and Teruko Mitamura. 2007. What is the jeopardy model? a quasi-synchronous grammar for qa. In *EMNLP-CoNLL*.

Zhiguo Wang, Haitao Mi, and Abraham Ittycheriah. 2016a. Semi-supervised clustering for short text via deep representation learning. In *CoNLL*.

Zhiguo Wang, Haitao Mi, and Abraham Ittycheriah. 2016b. Sentence similarity learning by lexical decomposition and composition. In *COLING*.

Zhiguo Wang, Wael Hamza, and Radu Florian. 2017. Bilateral multi-perspective matching for natural language sentences. In *IJCAI*.

Yi Yang, Wen tau Yih, and Christopher Meek. 2015. Wikiqa: A challenge dataset for open-domain question answering. In *EMNLP*.

Liu Yang, Qingyao Ai, Jiafeng Guo, and W. Bruce Croft. 2016. anmm: Ranking short answer texts with attention-based neural matching model. In *CIKM*.

Xuchen Yao, Benjamin Van Durme, Chris Callison-burch, and Peter Clark. 2013. Answer extraction as sequence tagging with tree edit distance. In *North American Chapter of the Association for Computational Linguistics (NAACL)*.

Scott Wen-tau Yih, Ming-Wei Chang, Chris Meek, and Andrzej Pastusiak. 2013. Question answering using enhanced lexical semantic models. ACL Association for Computational Linguistics, August.

Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. 2017. Recent trends in deep learning based natural language processing. *CoRR*, abs/1708.02709.

Lei Yu, Karl Moritz Hermann, Phil Blunsom, and Stephen G. Pulman. 2014. Deep learning for answer sentence selection. *CoRR*, abs/1412.1632.

Yuan Zhang, Regina Barzilay, and Tommi S. Jaakkola. 2017. Aspect-augmented adversarial networks for domain adaptation. *CoRR*, abs/1701.00188.

Hengtong Zhang, Yaliang Li, Fenglong Ma, Jing Gao, and Lu Su. 2018. Texttruth: An unsupervised approach to discover trustworthy information from multi-sourced text data. In *Proceedings of the 24rd SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM.