Question-Context Alignment and Answer-Context Dependencies for Effective Answer Sentence Selection

Minh Van Nguyen¹, Kishan KC², Toan Nguyen², Thien Nguyen¹, Ankit Chadha², and Thuy Vu²**

¹Department of Computer Science, University of Oregon, OR, USA
²Amazon Alexa AI, CA, USA
{minhnv,thien}@cs.uoregon.edu
{ckshan,amztoan,ankitrc,thuyvu}@amazon.com

Abstract

Answer sentence selection (AS2) in open-domain question answering finds an answer for a question by ranking candidate sentences extracted from web documents. Recent work exploits answer context, i.e., sentences around a candidate, by incorporating them as an additional input string to the Transformer models to improve the correctness scoring. In this paper, we propose to improve the candidate scoring by explicitly incorporating the dependencies between question-context and answer-context into the final representation of a candidate. Specifically, we use Optimal Transport to compute the question-based dependencies among sentences in the passage where the answer is extracted from. We then represent these dependencies as edges in a graph and use Graph Convolutional Network to derive the representation of a candidate, a node in the graph. Our proposed model achieves significant improvements on popular AS2 benchmarks, i.e., WikiQA and WDRASS, obtaining new state-of-the-art on all benchmarks.

Index Terms: question answering, human-computer interaction, large language model

1. Introduction

Voice-based virtual assistants powered by open-domain question answering (ODQA) [1, 2] have gained significant commercial market in recent years, e.g., Google Assistant, Siri, or Alexa, thanks to the progress in question answering using large pre-trained language models (LLMs) such as BERT [3], RoBERTa [4], and GPT-3 [5]. Recent works address the task via generative models [6, 7]. However, a well-known issue that has been shown to occur with the generative models is hallucination [8, 9, 10], where the models generate statements that are plausible looking but factually incorrect. Additionally, if the answers are composed by a pretrained LLM without external knowledge, the information contained in the answers might be outdated and no longer valid, e.g., the answer for the question “Which country is the reigning World Cup champion?” will change through time. To avoid such problems, this work follows a typical ODQA pipeline involving two main stages: web retrieval and answer sentence selection (AS2); the latter selects the most relevant answer sentence from the retrieved documents to return to the user. This stage is typically implemented as a point-wise model that scores each sentence individually either without any additional information [2, 11], jointly with other top candidates [12], or with other contextual information [13, 14]. We focus on the latter in this paper.

Table 1: The example shows contextual sentences, i.e., prev and next in a passage p for an answer a. Their relation with the question q is critical to select the correct answer a.

Previous work on AS2 exploits contextual information for better performance, however, is limited to the concatenation approach. In particular, Lauriola et al. [13] concatenated contextual sentences, i.e., previous (prev) and next sentences (next), with an answer candidate (cand) for a given question (q) as an input sequence to a Transformer architecture, i.e., using the structure “[CLS] q [SEP] prev [SEP] cand [SEP] next [EOS]” instead of “[CLS] q [SEP] cand [EOS]”. Han et al. [14], similarly, uses a concatenation approach for contextual sentences and document titles. While both models outperform the vanilla baseline, i.e., using only cand, we hypothesize that the improvement results from using a longer sequence for higher predictive power.

From a modeling viewpoint, the concatenation approach is effective largely due to the self-attention mechanism in the Transformer architecture [15]. In particular, the mechanism allows weighing the relevance of each token with other tokens from the input, i.e., question, candidate, and contextual sentences. The approach, however, could be sub-optimal for AS2 for two reasons. First, the concatenation approach fails to ignore irrelevant information in the context sentences, which could introduce noise to the prediction of the model. Second, the approach fails to explicitly capture the alignment between the question and the answer/context sentences, which can be done to better reveal the correctness for the answer candidate.

We address these shortcomings in this paper. First, we propose to align the tokens of the question and the paragraph’s sentences to enhance the relevance computing. Specifically, we propose to employ Optimal Transport (OT) [16, 17] to solve the question-context alignment problem. Given two probability
distributions over two point sets and a cost function that measures the distance between any two points, the goal of OT is to find a mapping that moves probability from one distribution to another such that the total cost of transporting the probability mass between two point sets is minimized. We consider the question and a paragraph sentence (i.e., the answer candidate or context sentence) as two point sets, each word being a point. To measure the distance between two points, we employ the Euclidean distance between their semantic representations, which can be obtained from a pre-trained language model (PLM), e.g., RoBERTa [4]. A probability distribution is also defined over each point set via the frequencies of the words in training data. Intuitively, the optimal alignment between the question and the candidate sentence maps the words that are both statistically and semantically similar between the two sentences. In the end, the relevant context formed by the set of the candidate words aligned with the question is utilized to compute the representation for the paragraph sentence.

Second, we address the dependencies among sentences in the paragraph, i.e., the answer candidate and its contextual sentences, to further bolster the modeling of the answer candidate representation. In particular, we consider the paragraph sentences as nodes in a fully-connected graph and aim to learn a dependency weight between nodes. To compute the dependency weight for two sentences, we propose to employ their semantic representations and transportation costs with respect to the question, which are already obtained from the question-context alignment step. A feed-forward network with a sigmoid output function is then used to consume such information to estimate the dependency weight. Afterwards, the dependency weights are utilized to enhance the representations for the sentences via a Graph Convolutional Network (GCN) [18]. The output representation from the GCN for the answer candidate sentence can be directly sent to a binary classifier to obtain its correctness probability score. We finally exploit the mutual information (MI) between the GCN representations of the sentences to further encourage information sharing between them, i.e., by maximizing and minimizing the mutual information between the sentence pairs. To this end, we treat the GCN representations of the sentences as continuous random variables. The MI between the variables can then be optimized via the mutual information neural estimation (MINE) method [19, 20], which approximately estimates the lower bound of the MI via the binary cross entropy of a variable discriminator for optimization.

To demonstrate the effectiveness of our proposed model for AS2, we conduct experiments on a widely-used AS2 benchmark, i.e., WikiQA [21], used in previous work [13] and a recent large-scale dataset WDRASS [22]. Experimental results across the datasets show that our model achieves significant improvements compared to the previous work, obtaining new state-of-the-art performance for AS2.

2. Related work

AS2 is an important task in ODQA and is often solved by pointwise methods that rank answer sentences extracted from retrieved Web documents [23, 24, 25, 11]. Contextual information, e.g., neighboring sentences or document title, was recently incorporated to improve AS2 [13, 14]. Previous work, however, exploited the additional context by string concatenating [13], i.e., appending the input sequence. By contrast, we propose to explicitly model the dependencies between the answer sentence and its context. We propose to align words between the question and the answer/context sentences via Optimal Transport (OT), a popular alignment method used in machine translation [26], event argument extraction [27], and event coreference resolution [28]. Our work is the first to apply OT to perform the question-context alignment for AS2.

3. Architecture

Given a question \( q \) and a set of \( N \) answer sentence candidates \( C = \{c_1, c_2, \ldots, c_N\} \), the task of AS2 seeks to find correct answers \( A \subseteq C \) via learning a reranking function \( r : Q \times \phi(C) \rightarrow \phi(C) \), where \( Q \) is the question set and \( \phi(C) \) is the set of all permutations of \( C \), such that the answer sentences \( A \) are on top of the ranking produced by \( r \). The reranker \( r \) is often implemented as a pointwise network \( f(q, c_i) \), e.g., TANDA [11], which learns a correctness score \( p_i \in (0, 1) \) for each answer candidate \( c_i \) for ranking. Our work focuses on contextual AS2 [13], where additional context such as surrounding sentences is considered to better determine the correctness score for an answer candidate.

Our proposed model, called “CASSIE”, for contextual AS2 has four main components: i) Encoding, ii) Question-Context Alignment with OT, iii) Answer-Context Dependencies, and iv) Mutual Information Optimization.

3.1. Encoding

We are given a question \( q = [w^q_1, w^q_2, \ldots, w^q_{T_q}] \) with \( T_q \) words and a set of \( N \) answer candidates \( C = \{c_1, c_2, \ldots, c_N\} \) (retrieved by a search engine), where each candidate is a sentence \( c_i = [w^c_{i1}, w^c_{i2}, \ldots, w^c_{iT_i}] \) with \( T_i \) words. Following the previous work [13], we consider previous and next sentences \( s_{\text{prev}}, s_{\text{next}} \) as additional context for each answer candidate \( c \in C \).

1. The input for our model is then formed by concatenating the question, the answer candidate, and the context sentences to obtain a single input sequence: \( [q; c; s_{\text{prev}}, s_{\text{next}}] \). The resulting sequence is fed into a pre-trained language model (PLM),

\[1\] We employ padding sentences if any context sentence is missing.
e.g., RoBERTa [4], to obtain contextualized embeddings for the words. In addition, we employ different segment embeddings for the words in the question, the answer candidate, and the context sentences. These segment embeddings, which are randomly initialized and learnable during training, are added to initial embeddings for the words in the first layer of the PLM. For convenience, let $[w_1^q, w_2^q, \ldots, w_T^q]$ and $[w_1^r, w_2^r, \ldots, w_T^r]$ be the sequences of word representations obtained from the last layer of the PLM for the question $q$ and the answer candidate $c \in C$ respectively.

### 3.2. Question-Context Alignment with OT

Optimal Transport (OT) [16, 17] is an established method to move probability from one distribution to another by finding an alignment between two point sets. In a discrete setting, we are given two probability distributions $p_x$ and $p_y$ over two point sets $X = \{x_i\}_{i=1}^m$ and $Y = \{y_j\}_{j=1}^n$ respectively ($\sum_i p_{xi} = 1$ and $\sum_j p_{yj} = 1$). A function $D(x, y) : X \times Y \rightarrow \mathbb{R}^+$ is also provided to measure the distance between two points $x$ and $y$. OT aims to find the transportation cost of transporting the probability mass between the two point sets is minimized. Formally, the goal of OT is to find the optimal transportation matrix $\pi \in \mathbb{R}^{m \times n}$ such that minimizes the total cost of transporting the probability mass between the two point sets.

In our question-context alignment problem, we consider the question $q$ and a candidate/context sentence $c$ as two point sets: $X = \{x_i\}_{i=1}^m$ and $Y = \{y_j\}_{j=1}^n$ respectively (each word is a point). To obtain the probability distributions for the sets, we propose to measure the frequencies of the words and perform a sum normalization. In particular, the probability distribution for the question is computed as follows:

$$p_{xq} = \frac{freq(w_i^q)}{\sum_{j=1}^{n} freq(w_j^q)}$$

where $freq(w_i^q)$ is the number of times that the word $w_i^q$ appears in training data. Next, to estimate the distance between two words (points) $w_i^q \in q$ and $w_j^c \in c$, we measure their semantic divergence by computing the Euclidean distance of their contextualized representations obtained from the PLM: $D(w_i^q, w_j^c) = ||w_i^q - w_j^c||$. The optimal transportation matrix $\pi_{X,Y}$, i.e., $\pi_{xc}$ for the question $q$ and the sentence $c$ can then be solved efficiently using the Sinkhorn-Knopp algorithm [29, 17].

Finally, we obtain the relevant context $r_c$ for the sentence $c$ as:

$$r_c = \argmax_{\pi_{xc} \in CT, \pi_{xc} \geq 0} \sum_{j=1}^{n} \pi_{xc} (\argmax_{i=1}^{m} \pi_{xc})$$

In the end, we compute the representation for the sentence $c$ as the average sum over the word representations for the relevant context:

$$r_c = \frac{1}{|r_c|} \sum_{j \in r_c} w_j^c$$

### 3.3. Answer-Context Dependencies

For convenience, let $[r_1, r_2, r_3]$ be the representations obtained from Equation (1) for the answer candidate $s_1 \equiv c$, the previous sentence $s_2 \equiv s_{prev}$, and the next sentence $s_3 \equiv s_{next}$. To learn the dependencies among the sentences, we consider each sentence as a node in a fully-connected graph $G = (V, E)$, where $V = \{s_i\} (1 \leq i \leq 3)$ is the node set and $E = \{(s_i, s_j)\} (1 \leq i, j \leq 3)$ is the edge set. Our goal is to learn a weight $\alpha_{ij} \in (0, 1)$ for each edge $(s_i, s_j)$ to represent the dependency of $s_i$ on $s_j$. To this end, we propose to employ their semantic representations $r_i, r_j$, and transportation costs to the question $d_{qs}$, $d_{cs}$, to measure the dependency weight $\alpha_{ij}$ between the sentences $s_i$ and $s_j$.

1We exclude stopwords and punctuations from the two point sets before performing the alignment.
3.5. Training and Inference

All the components in our proposed model are jointly trained via minimizing the loss function:

\[ L = L_{AS2} + \gamma L_{MT} \]  \hspace{1cm} (4)

where \( \gamma \) is a hyper-parameter to balance the contributions of each component to the training of the model. Following the previous work [11, 13], we consider all answer candidates for each question for training and inference.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Train</th>
<th>Dev</th>
<th>Test</th>
<th>#A/Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>WikiQA</td>
<td>2,118</td>
<td>126</td>
<td>243</td>
<td>1.18</td>
</tr>
<tr>
<td>WDRASS</td>
<td>53,419</td>
<td>5,416</td>
<td>5,395</td>
<td>4.96</td>
</tr>
</tbody>
</table>

Table 2: The number of questions in training, development, and test data of the two datasets. The last column presents the average number of answers to a question.

4. Results

4.1. Experimental Setup

Datasets Following the previous works [11, 13], we use the same train/dev/test splits for the standard AS2 dataset, i.e., WikiQA [21]. In addition, we also experimented with a large-scale AS2 dataset called WDRASS [30] to investigate the models’ performance further.

- WikiQA is a QA dataset created by [21]. The dataset contains questions and answer candidates, manually annotated on Bing query logs over Wikipedia. Following the previous works, we conduct experiments with the clean version of the dataset, and combine the development and test sets to obtain a larger and more reliable set for model comparison.
- WDRASS is recently created by [30]. WDRASS is a large-scale dataset focusing on non-factoid questions requiring entire sentences to answer.

Statistics for all the datasets are shown in Table 2.

Hyper-parameters and Tools Following the previous work [13], we use a small portion of the WikiQA training data to tune hyper-parameters for our model and select the best hyper-parameters for all the datasets. We employ Adam optimizer to train the model with a learning rate of \( 1 \times 10^{-5} \) and a batch size of 64. We set 400 for the hidden vector sizes for all the feed-forward networks, \( L = 2 \) for the number of the GCN layers, and 0.3 for the trade-off weights \( \gamma \). To implement the models, we use PyTorch version 1.7.1 and Huggingface Transformers version 3.5.1. We use the NLTK library version 3.5 [31] to preprocess the data and remove stopwords. The model performance is obtained over three runs with different random seeds.

Evaluation Metrics Following the previous works, we measure the model performance using the entire set of answer candidate sentences for each question, using the three metrics: Precision-at-1 (P@1), Mean Average Precision (MAP), and Mean Reciprocal Rank (MRR) scores.

4.2. Performance Comparison

We compare our proposed model with TANDA [11] and LOCT [13], which are the current state-of-the-art models for AS2. Table 3 shows the performance comparison between the models on two settings: i) using a non-finetuned RoBERTa base encoder, and ii) using a finetuned RoBERTa base encoder. The non-finetuned RoBERTa Base is obtained from [4] while the other is produced by finetuning TANDA on the ASNQ dataset [11]. As can be seen from the table, all the models benefit from using the finetuned RoBERTa Base encoder. Across the two settings, our model outperforms the previous models by large margins, demonstrating its effectiveness for AS2.

Table 4 shows the performance of our proposed model compared to TANDA on the WDRASS test set. CASSIE significantly improves the performance for P@1 and MRR, however, decreases the performance for MAP. We attribute this to the fact that our model ranks the answer candidates individually. To deal with this, we explore another use case of our model where it can produce a joint reranking for multiple answer candidates ranked by TANDA, leading to significantly better performance for all the three metrics.

5. Conclusions and Future Work

In this work, we propose CASSIE, a novel LLM-based model that (i) efficiently learns the answer-context dependencies to improve representation learning for AS2 by (ii) leveraging relevant context in answer/context sentences captured via question-context alignments using Optimal Transport. Experimental results demonstrate the efficacy of our proposed model, resulting in significant improvements and new state-of-the-art performance across several widely-used AS2 benchmark WikiQA [21] and a recent large-scale AS2 dataset (WDRASS) [30].

Naturally, our proposed model CASSIE can approximate human judgement for the correctness of an answer for a given question. As a result, the model can be used to provide signals for assessing quality of answers produced by open-domain question answering (ODQA) systems. As such, we are exploring AS2 datasets and architecture designs for building better reward models for reinforcement learning from human feedback [7], which has shown impressive improvements for ODQA via training LLMs to generate answers for open-domain questions.

6. Acknowledgments

We would like to thank Zeyu Zhang for sharing the WDRASS dataset [22] and his helpful comments on the paper regarding the experiment setup and data preparation.
7. References


