Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 5505–5514
Marseille, 11–16 May 2020

Contextualized Embeddings based Transformer Encoder for Sentence Similarity Modeling in Answer Selection Task

Md Tahmid Rahman Laskar[1, 3], Jimmy Huang[2, 3], Enamul Hoque[2]
[1] Department of Electrical Engineering and Computer Science, York University
[2] School of Information Technology, York University
[3] Information Retrieval & Knowledge Management Research Lab, York University
Toronto, Ontario, Canada
tahmedge@cse.yorku.ca, jhuang@yorku.ca, enamulh@yorku.ca

Abstract

Word embeddings that consider context have attracted great attention for various natural language processing tasks in recent years. In this paper, we utilize contextualized word embeddings with the transformer encoder for sentence similarity modeling in the answer selection task. We present two different approaches (feature-based and fine-tuning-based) for answer selection. In the feature-based approach, we utilize two types of contextualized embeddings, namely the Embeddings from Language Models (ELMo) and the Bidirectional Encoder Representations from Transformers (BERT) and integrate each of them with the transformer encoder. We find that integrating these contextual embeddings with the transformer encoder is effective to improve the performance of sentence similarity modeling. In the second approach, we fine-tune two pre-trained transformer encoder models for the answer selection task. Based on our experiments on six datasets, we find that the fine-tuning approach outperforms the feature-based approach on all of them. Among our fine-tuning-based models, the Robustly Optimized BERT Pretraining Approach (RoBERTa) model results in new state-of-the-art performance across five datasets.

Keywords: Answer Selection, Transformer Encoder, Contextualized Embeddings, ELMo, BERT, RoBERTa, Deep Learning

1. Introduction

Measuring the similarity between question answering pairs (Yih et al., 2013) is a fundamental problem in the areas of Information Retrieval and Natural Language Processing (NLP). For example, in the answer selection task, a question along with list of candidate answers are given and then the answers are ranked based on their relevance with the question (see Table 1). Several neural models for the answer selection task use word embeddings like GloVe (Pennington et al., 2014) or Word2Vec (Mikolov et al., 2013) as input and measure the similarity between the vector representation of two sentences produced by the models (Chen et al., 2018a; Chen et al., 2018b). However, such embeddings provide fixed representation for each word and do not capture its context in different sentences.

Recently, contextualized word representation methods such as ELMo and BERT received a lot of attention. The ELMo model (Peters et al., 2018) learns the contextualized word representations from a deep bidirectional language model pre-trained on large text corpora. The BERT model (Devlin et al., 2019) can also generate contextual embeddings like ELMo via utilizing the encoder of transformer (Vaswani et al., 2017) and yields very good results on tasks such as named-entity recognition. Since these contextual embeddings can capture better representation of a sentence by generating embedding of each word based on its surrounding context, we are motivated to use them for sentence similarity modeling in answer selection task.

In this paper, we investigate how to utilize the contextualized word embeddings by integrating with the transformer encoder for the answer selection task. We also perform extensive experiments with both contextual and traditional embeddings and observe how their integration with the transformer encoder may impact the performance for answer selection. More concretely, we make the following contributions. First, we present two new approaches (feature-based and fine-tuning-based) by utilizing contextualized embeddings, namely, the ELMo, BERT, and RoBERTA (Liu et al., 2019) for the answer selection task. Second, we observe that combining contextual embeddings with transformer encoder improves performance from models where only contextual embeddings were used. We also empirically show that the integration of transformer encoder with ELMo or BERT outperforms its integration with GloVe for answer selection task which confirms the effectiveness of using the contextualized embeddings. Finally, we achieve new state-of-the-art results for the an-

Question:

- Which country won the FIFA world cup 2018?

List of Candidate Answers:

- England have won the Cricket World Cup 2019.
- **France have won the FIFA world cup 2018.**
- France have won the FIFA world cup 2014.

Potential Ranking:

- **France have won the FIFA world cup 2018.**
- France have won the FIFA world cup 2014.
- England have won the Cricket World Cup 2019.

Table 1: An example of Answer Selection Task. A question along with list of candidate answers are given. The text in bold font is the correct answer.

swer selection task in different datasets by fine-tuning the RoBERTa model. As a contribution to the Language Resources and Evaluation (LRE), we share the source codes, pre-processed[1] datasets[2] through the LRE Map[3] for future research and replicability of experiments.

2. Related Work

Early work in sentence similarity modeling relies on different feature engineering based approaches (Wan et al., 2006; Yih et al., 2013). For example, (Yih et al., 2013) utilized the WordNet based semantic features for question answering. However, these feature-based approaches have some critical limitations. For instance, the features which are used in one dataset may not perform well in another dataset (Chen et al., 2018b). Moreover, these approaches require lots of handcrafted rules and are often error-prone.

Recently, several deep learning based approaches for sentence similarity modeling showed good performance without requiring any handcrafted features (Chen et al., 2017; Bian et al., 2017; Chen et al., 2018a; Chen et al., 2018b; Madabushi et al., 2018; Tymoshenko and Moschitti, 2018; Sha et al., 2018; Kamath et al., 2019; Rao et al., 2019). Some researchers focused on extracting the common features in a sentence pair to improve the sentence similarity modeling (Wang et al., 2016; Chen et al., 2018a; Chen et al., 2018b). (Chen et al., 2018b) proposed a Collaborative and Adversarial Network in which the generator and the discriminator were utilized for extracting common features between sentences. (Chen et al., 2018a) detected the aligned words in a sentence pair and then absorbed aligned words' contexts for the hidden state generation of Recurrent Neural Network (RNN).

Various attention models based on RNN (Tan et al., 2015; Santos et al., 2016; Chen et al., 2017) and transformer model (Vaswani et al., 2017) were also proposed for improving sentence representation. While (Vaswani et al., 2017) utilized both the encoder and the decoder of transformer for sequence-to-sequence task, later some researchers have only used the encoder or the decoder of the transformer model for different NLP tasks (Cer et al., 2018; Radford et al., 2018). For example, (Cer et al., 2018) utilized the transformer encoder to generate embeddings from sentences for transfer learning to other tasks. (Radford et al., 2018) utilized the transformer decoder by pre-training it on large corpora for the language modeling task and then fine-tuned the model for downstream tasks.

The recently proposed BERT (Devlin et al., 2019) and ELMo (Peters et al., 2018) models are pre-trained on large text corpora and provide contextualized representations of words in a sentence. The former uses the transformer encoder (Vaswani et al., 2017) and the latter uses a deep, bi-directional long short term memory (LSTM) model (Hochreiter and Schmidhuber, 1997). BERT showed state-of-the-art results for several tasks including question answering in the Stanford Question Answering Dataset (SQuAD) by fine-tuning the pre-trained model. Later, various models based on BERT were proposed which improved the traditional BERT model. For example, the XLNet model (Yang et al., 2019) utilized the Transformer-XL (Dai et al., 2019) architecture and outperformed BERT in several NLP tasks by effectively learning contextual representations. The RoBERTa model (Liu et al., 2019) significantly improved the performance of the BERT model by utilizing new design choices and training strategies and exceeded the performance of XLNet and other models published based on BERT (Liu et al., 2019). More recently, the gated self attention network (Lai et al., 2019) was proposed which was combined with BERT along with transfer learning from a large-scale online corpus and provided improvement in the TREC-QA (Wang et al., 2007) and WikiQA (Yang et al., 2015) datasets for the answer selection task.

Despite being a promising approach, integrating contextualized word embeddings with transformer encoder model has rarely been evaluated for the answer selection task yet. A notable exception is the work from (Garg et al., 2019), where the pre-trained transformer encoder based models (BERT/RoBERTa) were first transferred into a model for a general task by fine-tuning it with a large dataset created from Wikipedia (Kwiatkowski et al., 2019). Then, another fine-tuning step was performed to adapt the transferred model in the target domain. Our fine-tuning approach differs from the above work as we used only one-step fine-tuning in the target domain without requiring any additional dataset. In addition, previous works based on pre-trained transformer encoder for answer selection (Garg et al., 2019; Lai et al., 2019; Laskar et al., 2019) were only evaluated on TREC-QA and WikiQA datasets, but these models were not evaluated on community answer selection datasets, such as SemEvalCQA (Nakov et al., 2015; Nakov et al., 2016; Nakov et al., 2017) and YahooCQA (Tay et al., 2017). In comparison to the previous work, we use six publicly available datasets for conducting a series of experiments to investigate the robustness of our approach.

3. Contextualized Embeddings based Transformer Encoder (CETE)

Let us assume that we want to measure the similarity between the two sentences $X = x_1, x_2, ..., x_m$ and $Y = y_1, y_2, ..., y_n$ for the answer selection task. We utilize the transformer encoder for this task based on two approaches: i) Feature-based approach, and ii) Fine-tuning-based approach.

In the feature-based approach, our framework works in the following steps as demonstrated in Figure 1(a). Our model first takes each token $x_i \in$ X and creates contextualized embedding representations $\mathbf{x_i}$ using pre-trained BERT/ELMo. It then combines the token embeddings with positional encodings (to track the order of the token sequence) and sends them to a randomly initialized transformer encoder. After the self-attention calculation, the resulting representations are passed through the feed-forward and pooling layers to obtain condensed vectors H_X and H_Y that represent sentence X and Y respectively. Finally, the cosine similarity

[1] As the question answering datasets are usually available in unstructured text format, we share the pre-processed version of each dataset for future reproduction of results.

[2] We did not share The YahooCQA dataset. It can be downloaded from `https://webscope.sandbox.yahoo.com/catalog.php?datatype=l&did=10`

[3] `http://lremap.elra.info/`

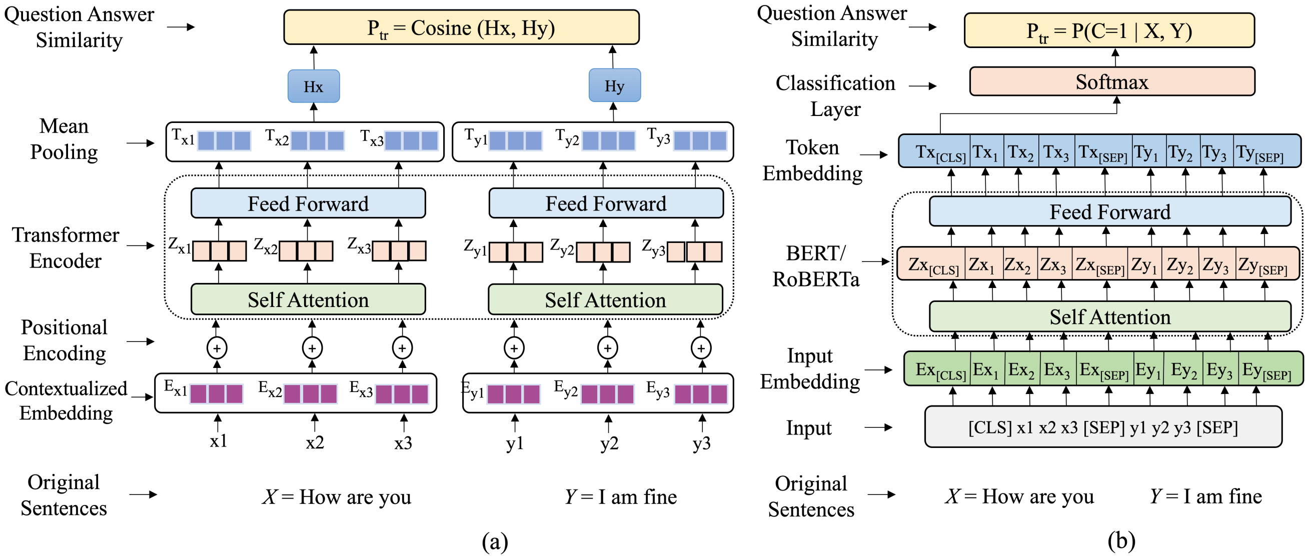


Figure 1: Our similarity modeling framework that applies contextualized embeddings: (a) Feature-based approach using the transformer encoder. (b) Fine-tuning-based approach using the BERT/RoBERTa model.

is calculated between H_X and H_Y.
In the fine-tuning approach, we adopt the pre-trained transformer encoder and then fine-tune it for question-answer similarity task for answer selection (see Figure 1(b)). In the following, we describe the two approaches in details.

3.1. Feature Based Approach

As shown in Figure 1(a), our feature-based approach adopts the encoder from the transformer model (Vaswani et al., 2017), which includes both encoder and decoder to perform machine translation task. We first extract the ELMo/BERT embeddings and feed them to the encoder. The encoder uses a self-attention layer to represent each token based on other relevant words in the sentence. This is done by creating three vectors for each token, namely a query vector Q, a key vector K, and a value vector V. These three vectors were created by multiplying the embedding vector $\mathbf{x_i}$ with three weight matrices (W_Q, W_K, W_V) respectively. These three weight matrices are updated during the training. Then, the output Z of self-attention for each word is computed as:

$$Z = softmax\left(\frac{Q \times K^T}{\sqrt{d_k}}\right) V \qquad (1)$$

Note that the transformer encoder uses multi-head attention mechanism to give attention on different positions. This is done by calculating the self attention eight times with eight different Query/Key/Value weight matrices to obtain eight Z matrices. It then concatenates the eight Z matrices into a single matrix and multiplies that matrix with an additional weight matrix and sends the resulting matrix to the feed-forward layer. We apply the mean pooling method (Wang and Nyberg, 2015) to obtain the sentence representation H. In order to obtain contextualized embeddings, we extract features from both ELMo and BERT models.

ELMo Embeddings: Instead of using fixed embedding representation, ELMo provides contextual embedding of a word based on its context in the entire sentence (Peters et al., 2018). Thus, it can capture multiple meanings of a word based on where it is used. ELMo uses a bidirectional LSTM (Hochreiter and Schmidhuber, 1997) to have a sense of both the next and the previous word. It is pre-trained on a vast amount of text data (Chelba et al., 2013) and provides three layers of representations for each word: one layer provides a character based word representation and the other two layers are the LSTM hidden states. The ELMo layer is the weighted sum of these three layers. We use the output of this layer as contextual word embeddings.

BERT Embeddings: The BERT model (Devlin et al., 2019) can also provide contextualized embeddings like ELMo. The model was originally pre-trained for masked language modeling and next sentence prediction task on the BooksCorpus (800M words) (Zhu et al., 2015) dataset along with the English Wikipedia (2,500M words). We use the token embeddings generated from the BERT model and feed them into the transformer encoder.

3.2. Fine Tuning Based Approach

Instead of using contextualized embeddings as input to a randomly initialized transformer encoder, we also experiment by fine-tuning the pre-trained BERT and RoBERTa models for the answer selection task as demonstrated in Figure 1(b).

BERT Fine-tuning for Answer Selection: In the BERT model, sentence pairs are combined together into a single sequence, separated by a special token $[SEP]$. The output of BERT is taken only for the first token ($[CLS]$), which is used as the aggregate representation of the sequence. During fine-tuning, parameters are added for an additional classification layer W. All the parameters of the pre-trained BERT model along with W are fine-tuned jointly to maximize the log-probability of the correct label. The label probabilities $P \in \mathbb{R}^K$ (where K is the total number of clas-

Dataset		# Questions			# Candidate Answers		
		Train	Dev	Test	Train	Dev	Test
TREC-QA	RAW	1229	82	100	53417	1148	1517
	Cleaned	1229	65	68	53417	1117	1442
WikiQA[5]	RAW	2118	296	633	20360	2733	6165
	Cleaned	873	126	243	8672	1130	2351
YahooCQA		50112	6289	6283	253440	31680	31680
SemEval-2015CQA		2600	300	329	16541	1645	1976
SemEval-2016CQA		4879	244	327	36198	2440	3270
SemEval-2017CQA		4879	244	293	36198	2440	2930

Table 2: Dataset Overview ('#' denotes 'Number of' and 'RAW' indicates the 'Original' version).

sifier labels) are calculated as follows:

$$P = softmax(CW^T) \tag{2}$$

In the answer selection task, there are two classifier labels (similar = 1, dissimilar = 0). In the original BERT model (Devlin et al., 2019), sentence pair classification task was done by predicting the correct label (1 or 0). But in our work, we modify the final layer by only considering the predicted score P_{tr} for the similarity label to rank the answers based on question-answer similarity.

$$P_{tr} = P(C = 1|X, Y) \tag{3}$$

RoBERTa Fine-tuning for Answer Selection: Since the BERT model was significantly undertrained (Liu et al., 2019), the RoBERTa model was proposed by modifying different hyperparameters in BERT along with new design choices. More specifically, RoBERTa used a much larger mini batches and learning rates compared to BERT. Also, the next sentence prediction task was removed from the pre-training stage. Five different datasets were used for pre-training, which in total consists of around 160GB of uncompressed text. These new parameter settings and objectives showed significant improvements in the BERT model in different NLP tasks[4]. To fine-tune it for the answer selection task, we followed the similar approach of BERT fine-tuning by modifying the final layer to utilize it for similarity modeling.

4. Experimental Setup

To evaluate the effectiveness of our approach, we ran experiments on six different datasets. In this section, we present the description of the datasets, evaluation metrics, the training procedure and parameter settings used in our experiments.

4.1. Datasets

We used six datasets for the answer selection task as shown on Table 2. Specifically, we used two widely used question answering (QA) datasets namely the TREC-QA and WikiQA as well as four community question answering (CQA) datasets, namely, the YahooCQA, SemEval-2015CQA, SemEval-2016CQA, and SemEval-2017CQA.

TREC-QA: This dataset is created from the QA track (8-13) of Text REtrieval Conference (Wang et al., 2007). It has two versions: RAW and Cleaned. The difference between the two versions is that the RAW version has some questions for which there is no answer or there are only positive/negative answers, whereas the Cleaned version removes those instances from the development and test sets. As a result, the RAW version contains 1148 QA pairs in the development set and 1517 QA pairs in the test set whereas the Cleaned version contains 1117 QA pairs in the development set and 1442 QA pairs in the test set.

WikiQA: This is an open domain QA dataset (Yang et al., 2015) in which the answers were collected from the Wikipedia. In this dataset, there are many questions that do not contain any answers. Only 873, 126, and 243 questions out of 2118, 296, 633 questions in the training, development, and test sets contain any answers, respectively[5].

YahooCQA: This dataset was prepared for answer selection task by (Tay et al., 2017) from the *Yahoo! Answers Manner Question*[6] dataset. It is a community-based question answering dataset and comparatively larger than TREC-QA or WikiQA. Each question in YahooCQA dataset is associated with at most one correct answer. The negative answers were generated by sampling 4 samples from the top 1000 hits obtained via Lucene[7]search. There are 253440, 31680, and 31680 QA pairs in the training, development, and test sets respectively.

SemEval-2015CQA: This CQA dataset is created from *Qatar Living Forums*[8]. We focus on subtask A, the question-comment similarity task. Each comment is tagged with "Good", "Bad" or "Potentially Useful". We consider "Good" as positive and other tags as negative examples by following the work of (Sha et al., 2018).

SemEval-2016CQA: This is another CQA dataset created from *Qatar Living Forums*. Though the task is similar to SemEval-2015CQA, the dataset used in SemEval-2016CQA is different.

SemEval-2017CQA: This one has the same training and development sets as SemEval-2016CQA. Only the test set is different in SemEval-2017CQA which contains 293 questions whereas the SemEval-2016CQA contains 327 questions.

[4] https://gluebenchmark.com/leaderboard

[5] For WikiQA, we used the original training data. But evaluation was done only on the cleaned test data where questions having no correct answers were removed.

[6] https://webscope.sandbox.yahoo.com

[7] https://lucene.apache.org/

[8] https://www.qatarliving.com/forum

4.2. Evaluation Metrics

Similar to the recent work on answer selection (Rao et al., 2019; Garg et al., 2019; Lai et al., 2019; Laskar et al., 2019), we used Mean Average Precision (MAP) and Mean Reciprocal Rank (MRR) as evaluation metrics to measure the performance of our models.

4.3. Training and Parameter Settings

In the feature-based approach, the dimensions of the input layer and the output layer as well as the inner feed forward layer were same. Specifically, when we used ELMo, the dimensions of hidden layers $\mathbf{d_{model}}$ and feed forward layers $\mathbf{d_{ff}}$ were both set to 1024. When we used the BERT$_{\text{Base}}$ model, the size of $\mathbf{d_{model}}$ and $\mathbf{d_{ff}}$ were set to 768. Like the original transformer (Vaswani et al., 2017), we set the number of attention heads **A** to 8. However, based on the performance in the development set, we used one encoder layer (**L** = 1) instead of six identical layers used in the original transformer model. We utilized the Adam optimization algorithm (Kingma and Ba, 2014) for the parameter update. The learning rate was set to 5×10^{-5} and the dropout value was set to 2×10^{-1}. We used the mean squared error as our loss function.

We fine-tuned with both Base and Large versions of BERT and RoBERTa models for the pairwise sentence classification task (Devlin et al., 2019). For training, we used cross entropy loss function to calculate the loss. The parameters of the BERT$_{\text{Base}}$ and RoBERTa$_{\text{Base}}$ models were: $\mathbf{d_{model}}$ = 768, $\mathbf{d_{ff}}$ = 3072, **A** = 12, **L** = 12. For the BERT$_{\text{Large}}$ and RoBERTa$_{\text{Large}}$ models, the parameters were: $\mathbf{d_{model}}$ = 1024, $\mathbf{d_{ff}}$ = 4096, **A** = 16, **L** = 24. For all models, the Adam was used as the optimizer, the batch size $\in \{8, 16, 24\}$, and the learning rates $\in \{1\text{e}{-5}, 2\text{e}{-5}\}$.

4.4. Implementation

We implemented our models using Pytorch. For fine-tuning, we used the Pytorch version[9] of the transformer model (Wolf et al., 2019) and ran our experiments in multi GPU setting with 4 Nvidia P100 GPUs. For the feature-based approach, experiments were run using a single Nvidia 1080 GPU. BERT contextual embeddings were generated using MXNet[10] library and the ELMo contextual embeddings were generated using AllenNLP[11] library.

5. Results and Analyses

We performed extensive experiments to compare our contextualized embeddings based transformer encoder (CETE) with the recent progress. To understand the effectiveness of our approaches, we also compared with several baselines.

For the feature-based approach, we used a baseline that had the transformer encoder but used GloVe word embeddings (Pennington et al., 2014) which do not consider the context. As the dimensions in the hidden layers and the feed forward layers in our feature-based approach were same as the dimension of the contextualized embeddings, we also used the dimensions in those layers for this baseline same as the dimension of GloVe: $\mathbf{d_{model}}$ = 300, $\mathbf{d_{ff}}$ = 300, **A** = 6, **L** = 1.

For the fine-tuning approach, we compared our models with the fine-tuned XLNet model (Yang et al., 2019). The XLNet model did not use the original transformer encoder (Vaswani et al., 2017). It utilized the ideas from the Transformer-XL (Dai et al., 2019) model by using the segments recurrence mechanism and the relative encoding scheme into pre-training. It also proposed the permutation-based language modeling to capture bidirectional context.

5.1. Effectiveness of Feature-based Approach

Performance on TREC-QA and WikiQA: Table 3 shows the performance of our models in TREC-QA and WikiQA respectively. We can see that integrating the transformer encoder with ELMo or BERT have outperformed the baseline where only the GloVe embedding was used with the transformer encoder. Specifically, in terms of MAP, our CETE model with ELMo achieves 13.56% improvement over the transformer encoder with GloVe in the WikiQA dataset. For the TREC-QA dataset, the improvement is 12.71% in the RAW version and 8.65% in the Cleaned version over the baseline. Though our best performing CETE model with ELMo could not outperform the state-of-the-art (Lai et al., 2019) in the WikiQA dataset, it outperformed or provided comparable performance with many recent work (Chen et al., 2018b; Chen et al., 2018a; Sha et al., 2018; Tymoshenko and Moschitti, 2018; Kamath et al., 2019). For both versions of TREC-QA dataset, the feature-based CETE models do not outperform the state-of-the-art models (Kamath et al., 2019; Garg et al., 2019). However, their performances are still comparable or better than many recent work (Tay et al., 2018; Tymoshenko and Moschitti, 2018; Chen et al., 2018a; Rao et al., 2019).

Performance on CQA datasets: We show the performance of our models in four CQA datasets in Table 4. We again notice that our proposed approach of integrating transformer encoder with ELMo or BERT have outperformed the baseline in all the CQA datasets. Specifically, in terms of MAP, our best performing feature-based approach BERT$_{\text{Large}}$ achieves 16.64% improvement over the transformer encoder with GloVe in the YahooCQA dataset. In the SemEvalCQA datasets, The CETE model with BERT$_{\text{Base}}$ performs the best with an improvement of 5.58%, 4.32%, and 6.19% over the baseline in terms of MAP in the SemEval-2015CQA, SemEval-2016CQA, and SemEval-2017CQA datasets respectively. Though none of our feature-based approaches outperform the current state-of-the-art models[12] (Nakov et al., 2017; Sha et al., 2018), they show comparable performance in each dataset.

Case study: To get deeper insights about why our CETE models with ELMo or BERT are more effective than the transformer encoder with GloVe, we randomly selected some question and candidate answer pairs and analyze the word-by-word similarity between them. Figure 2 shows the word-by-word similarity heatmap for a relevant question

[9] `https://github.com/huggingface/pytorch-transformers`

[10] `https://mxnet.apache.org/`

[11] `https://allennlp.org/`

[12] We did not report any recent progress for SemEval2015-CQA as we found that prior work used different evaluation metrics.

Model	**TREC-QA**				**WikiQA**	
	RAW		**Cleaned**			
	MAP	**MRR**	**MAP**	**MRR**	**MAP**	**MRR**
(Chen et al., 2017)	-	-	0.781	0.851	0.721	0.731
(Bian et al., 2017)	-	-	0.821	0.899	0.754	0.764
(Tay et al., 2018)	0.770	0.825	0.784	0.865	0.712	0.727
(Chen et al., 2018a)	-	-	0.823	0.889	0.736	0.745
(Chen et al., 2018b)	-	-	0.841	0.917	0.730	0.743
(Sha et al., 2018)	-	-	-	-	0.746	0.758
(Madabushi et al., 2018)	0.836	0.863	0.865	0.904	-	-
(Tymoshenko and Moschitti, 2018)	0.777	0.869	-	-	0.762	0.776
(Kamath et al., 2019)	0.852	0.891	-	-	0.700	0.716
(Rao et al., 2019)	0.774	0.843	-	-	-	-
(Lai et al., 2019)	-	-	0.914	0.957	0.857	0.872
(Garg et al., 2019)	-	-	**0.943**	0.974	-	-
Transformer Encoder + GloVe	0.708	0.764	0.728	0.812	0.671	0.686
CETE (ELMo Embeddings)	0.798	0.869	0.791	0.858	0.762	0.774
CETE (BERT$_{\text{Base}}$ Embeddings)	0.799	0.855	0.791	0.857	0.727	0.741
CETE (BERT$_{\text{Large}}$ Embeddings)	0.806	0.897	0.789	0.887	0.714	0.731
XLNet$_{\text{Base}}$ Fine Tuning	0.903	0.939	0.900	0.938	0.808	0.820
XLNet$_{\text{Large}}$ Fine Tuning	0.939	0.979	0.920	0.973	0.836	0.847
CETE (BERT$_{\text{Base}}$ Fine Tuning)	0.891	0.925	0.888	0.953	0.829	0.843
CETE (BERT$_{\text{Large}}$ Fine Tuning)	0.917	0.947	0.905	0.967	0.843	0.857
CETE (RoBERTa$_{\text{Base}}$ Fine Tuning)	0.927	0.962	0.905	0.950	0.847	0.860
CETE (RoBERTa$_{\text{Large}}$ Fine Tuning)	**0.950**	**0.980**	0.936	**0.978**	**0.900**	**0.915**

Table 3: Performance comparisons with recent progress on TREC-QA and WikiQA datasets.

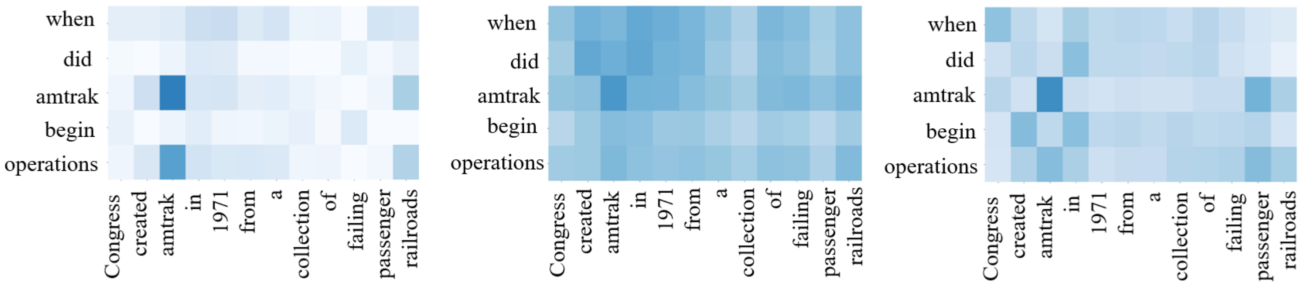
Model	**YahooCQA**		**SemEvalCQA**					
			SemEval-2015		**SemEval-2016**		**SemEval-2017**	
	MAP	**MRR**	**MAP**	**MRR**	**MAP**	**MRR**	**MAP**	**MRR**
(Nakov et al., 2017)	-	-	-	-	-	-	0.884	0.928
(Tay et al., 2018)	-	0.801	-	-	-	-	-	-
(Sha et al., 2018)	-	-	-	-	0.801	0.872	-	-
Transformer Encoder + GloVe	0.667	0.667	0.843	0.864	0.741	0.810	0.824	0.881
CETE (ELMo Embeddings)	0.762	0.762	0.875	0.909	0.767	0.824	0.860	0.914
CETE (BERT$_{\text{Base}}$ Embeddings)	0.776	0.776	0.890	0.924	0.773	0.835	0.875	0.922
CETE (BERT$_{\text{Large}}$ Embeddings)	0.778	0.778	0.883	0.923	0.765	0.831	0.867	0.922
XLNet$_{\text{Base}}$ Fine Tuning	0.939	0.939	0.929	0.960	0.849	0.912	0.902	0.934
XLNet$_{\text{Large}}$ Fine Tuning	0.945	0.945	0.945	0.969	0.860	0.912	0.930	0.962
CETE (BERT$_{\text{Base}}$ Fine Tuning)	0.948	0.948	0.923	0.949	0.843	0.906	0.904	0.942
CETE (BERT$_{\text{Large}}$ Fine Tuning)	0.951	0.951	0.935	0.961	0.866	0.927	0.921	0.963
CETE (RoBERTa$_{\text{Base}}$ Fine Tuning)	0.951	0.951	0.933	0.956	0.851	0.900	0.909	0.944
CETE (RoBERTa$_{\text{Large}}$ Fine Tuning)	**0.955**	**0.955**	**0.947**	**0.970**	**0.888**	**0.938**	**0.943**	**0.974**

Table 4: Performance comparisons with recent progress on YahooCQA and SemEvalCQA datasets.

and candidate answer pair. We observe that for the similar sentence pair, words between two sentences are more similar in the CETE model with ELMo/BERT embeddings than the transformer encoder with GloVe model. It indicates that for the similar sentence pair, the transformer encoder with contextualized embeddings effectively captures the overall context of the sentence, resulting in better performance for answer selection. For the dissimilar sentence pair, we observe from Figure 3 that the words between the sentence pair were more dissimilar in the CETE model than the transformer encoder with GloVe model, suggesting the effectiveness of integrating contextualized embeddings with the transformer encoder.

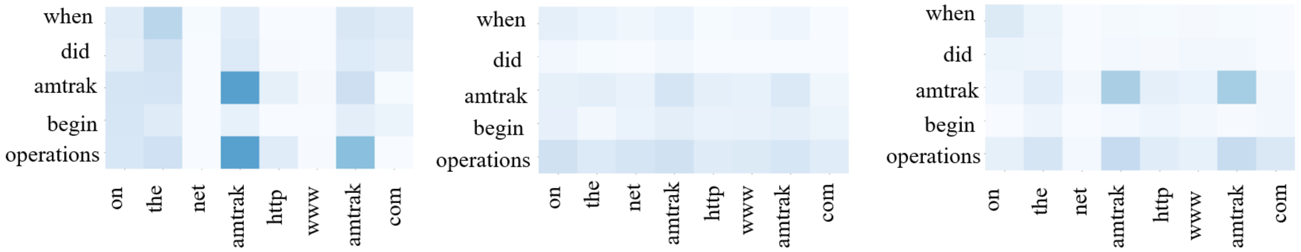
5.2. Effectiveness of Fine-tuning Approach

Performance on TREC-QA and WikiQA: We fine-tune the pre-trained BERT, RoBERTa, and XLNet models for answer selection. Among these three models, BERT and RoBERTa are transformer encoder based, whereas the XLNet model is based on Transformer-XL. We find that the Large versions of BERT, RoBERTa, and XLNet always outperform their respective Base versions in all datasets. While comparing XLNet with BERT, we find that XLNet outperforms BERT in both versions of the TREC-QA. However, in the WikiQA dataset, the BERT model achieves superior performance. In all datasets, the RoBERTa model outperforms both XLNet and BERT.



(a) GloVe + Transformer Encoder (b) ELMo + Transformer Encoder (c) $\text{BERT}_{\text{BASE}}$ + Transformer Encoder

Figure 2: Similarity between words of a question and a relevant candidate answer. (Darker color indicates more similarity)



(a) GloVe + Transformer Encoder (b) ELMo + Transformer Encoder (c) $\text{BERT}_{\text{BASE}}$ + Transformer Encoder

Figure 3: Similarity between words of a question and an irrelevant candidate answer. (Lighter color indicates less similarity)

In comparison to the prior work, we observe new state-of-the-art results in the RAW version of TREC-QA by fine-tuning with both Base and Large versions of BERT. The fine-tuned $\text{BERT}_{\text{Large}}$ and $\text{BERT}_{\text{Base}}$ models have an improvement of 7.63% and 4.58% respectively in terms of MAP over the previous state of the art (Kamath et al., 2019). However, they do not achieve the state-of-the-art results in the Cleaned version of TREC-QA (Garg et al., 2019) as well as in the WikiQA (Lai et al., 2019). With the fine-tuned RoBERTa models, we observe even more improvements in the WikiQA and the RAW version of TREC-QA compared to the fine-tuned BERT models. In the RAW TREC-QA, both the Base and Large versions of RoBERTa outperform the previous state-of-the-art (Kamath et al., 2019), with $\text{RoBERTa}_{\text{Large}}$ performing the best with MAP 0.950 and MRR 0.980. In the WikiQA dataset, our fine-tuned $\text{RoBERTa}_{\text{Large}}$ model sets a new state-of-the-art result with an improvement of 5.02% in terms of MAP and 4.93% in terms of MRR than the previous best performing model (Lai et al., 2019).

Though our approach of fine-tuning $\text{RoBERTa}_{\text{Large}}$ provides new state-of-the-results in TREC-QA (RAW) and WikiQA datasets, in terms of MAP it could not outperform the $\text{RoBERTa}_{\text{Large}}$ with Transfer Learning approach: the RoBERTa-TANDA model (Garg et al., 2019) in the Cleaned version of TREC-QA. It is to be noted that in the two-step fine-tuning-based RoBERTa-TANDA model, the first step of fine-tuning was done in a large dataset created from the Wikipedia (Kwiatkowski et al., 2019) which contains 57242 questions, along with more than 20 Millions candidate answers. Then the second step of fine-tuning was done in the target domain. In comparison to them, we only do a one-step fine-tuning in the target domain which contains 1229 questions with 53417 candidate answers in the training set (our training data size is only about 2% of total questions and 0.3% of total candidate answers used to train the RoBERTa-TANDA model). Without the leverage of large dataset, our fine-tuning approach provides almost similar result in terms of MAP with only 0.75% less than the RoBERTa-TANDA model. In terms of the MRR, we observe a new state-of-the-art result with an improvement of 0.41% compared to the RoBERTa-TANDA. We did not report the result of RoBERTa-TANDA for the WikiQA dataset as the number of questions and the candidate answers used in their test data were different than ours.

Performance on CQA datasets: In the CQA datasets, we again find that the RoBERTa model outperforms both BERT and XLNet. Among BERT and XLNet, we find that $\text{BERT}_{\text{Large}}$ outperforms XLNet in the YahooCQA and SemEval-2016 datasets, whereas in SemEval-2015 and SemEval-2017, $\text{XLNet}_{\text{Large}}$ outperforms BERT.

We observe new state-of-the-art results in all CQA datasets by fine-tuning both BERT and RoBERTa models. Though both Base and Large versions of BERT and RoBERTa provide state-of-the-art results across the CQA datasets, we find that the Large version outperforms the Base version in all of them. For the SemEval datasets, our best performing $\text{RoBERTa}_{\text{Large}}$ model has an improvement of 10.86% in SemEval-2016CQA and 6.67% in SemEval-2017CQA in terms of MAP than the state-of-the-art models (Sha et al., 2018; Nakov et al., 2017), respectively. For the YahooCQA dataset, the $\text{RoBERTa}_{\text{Large}}$ model again performs the best with an improvement of 19.23% in terms of MRR than the previous state-of-the-art result (Tay et al., 2018)[13].

[13] The authors did not report MAP in their paper.

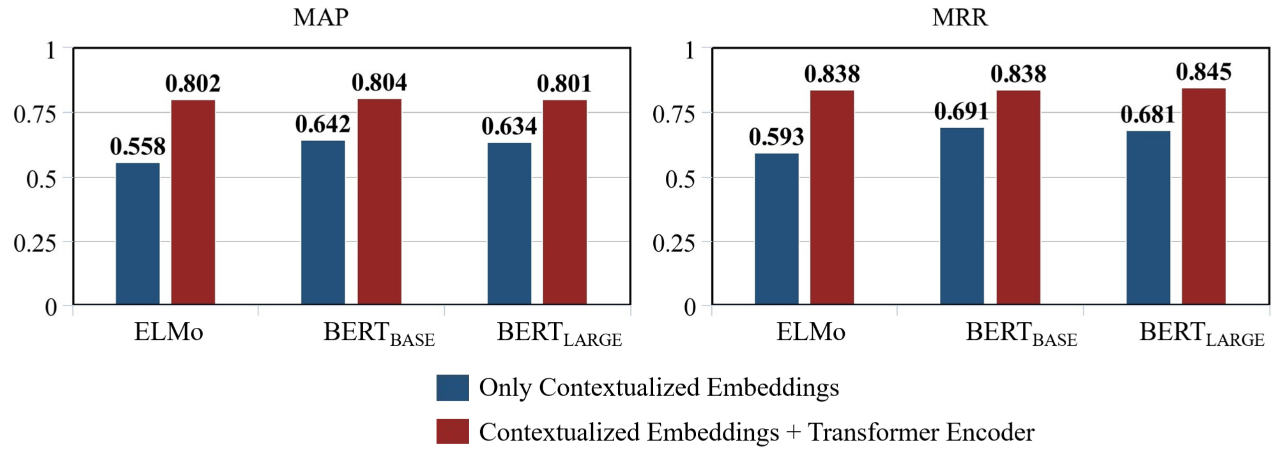


Figure 4: Performance comparisons based on the ablation test. Only the fixed contextual embeddings of each model (ELMo, BERT$_{\text{Base}}$, and BERT$_{\text{Large}}$) are compared with their integration with Transformer Encoder. MAP and MRR of each model are based on the average across all datasets.

Comparing Fine-tuning with Feature Extraction: In all datasets, we notice that based on paired t-test ($p \leq .05$), all the fine-tuning-based approaches perform significantly better than the feature-based approaches. It is worth noting that in our feature-based approach, features are extracted from the pre-trained model and they are fed to the transformer encoder model which is required to be trained from scratch. In contrast, in the fine-tuning approach, the pre-trained model is fine-tuned for a specific task by adding few additional randomly initialized parameters. As (Peters et al., 2019) suggest, the performance of the fine-tuning-based approach and the feature-based approach depends on the similarity between the pre-training and target tasks. They also observed that fine-tuning the BERT model significantly outperformed the feature-based approaches for the textual similarity task. This may explain why fine-tuning approach performs better than the feature-based approach for the answer selection task.

5.3. Ablation Study

In order to better investigate the effectiveness of integrating transformer encoder with contextual embeddings in the feature-based approach, we perform an ablation study. In the ablation study, following models are included:

- ELMo Embeddings: Only the feature-based ELMo Embeddings were used without any Transformer Encoder.
- BERT$_{\text{Base}}$ Embeddings: Only the feature-based BERT$_{\text{Base}}$ Embeddings were used without any Transformer Encoder.
- BERT$_{\text{Large}}$ Embeddings: Only the feature-based BERT$_{\text{Large}}$ Embeddings were used without any Transformer Encoder.

The above models simply measure the similarity between the question and the candidate answers based on the fixed contextualized embeddings generated from ELMo and BERT without sending them to the transformer encoder. We compare these models with our feature-based models that do include the transformer encoder. The results of our ablation study based on the average MAP and MRR scores across all datasets are given in Figure 4.

From the ablation study, we find that integrating transformer encoder with contextual embeddings improves the performance by 43.73%, 25.23%, and 26.34% in terms of MAP and 41.32%, 21.27%, and 24.08% in terms of MRR in ELMo, BERT$_{\text{Base}}$, and BERT$_{\text{Large}}$ respectively. These improvements are statistically significant based on paired t-test ($p \leq .05$). This shows the effectiveness of our proposed approach of integrating transformer encoder with the features extracted from the BERT or ELMo models.

6. Conclusions and Future Work

We present two approaches to utilize contextualized embeddings with the transformer encoder for the answer selection task. Our experiments on six datasets demonstrate that the performance of our feature-based approach is comparable with most of the prior work. More importantly, we find that our approach of fine-tuning the pre-trained transformer encoder models for answer selection is very effective even without the leverage of transfer learning from large corpora. We also observe that our fine-tuned RoBERTa model sets new state-of-the-art results on all six datasets in terms of MRR. Finally, we share the resources through LRE Map for further research and reproducibility of experiments. In future, we will investigate the performance of transformer based models on more tasks, such as information retrieval applications (Huang and Hu, 2009; Huang et al., 2003; Yin et al., 2013; Huang et al., 2005), sentiment analysis (Liu et al., 2007; Yu et al., 2012), learning from imbalanced datasets (Liu et al., 2006), named-entity recognition (Bari et al., 2019), and query focused abstractive summarization (Nishida et al., 2019; Nema et al., 2017).

7. Acknowledgements

This research is supported by the Natural Sciences & Engineering Research Council (NSERC) of Canada and an ORF-RE (Ontario Research Fund-Research Excellence) award in BRAIN Alliance. We would like to thank Dr. Qin Chen for helping us with the experiments.

8. References

Bari, M. S., Joty, S., and Jwalapuram, P. (2019). Zero-resource cross-lingual named entity recognition. *arXiv preprint arXiv:1911.09812*.

Bian, W., Li, S., Yang, Z., Chen, G., and Lin, Z. (2017). A compare-aggregate model with dynamic-clip attention for answer selection. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1987–1990.

Cer, D., Yang, Y., Kong, S.-y., Hua, N., Limtiaco, N., John, R. S., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., et al. (2018). Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

Chelba, C., Mikolov, T., Schuster, M., Ge, Q., Brants, T., Koehn, P., and Robinson, T. (2013). One billion word benchmark for measuring progress in statistical language modeling. *arXiv preprint arXiv:1312.3005*.

Chen, Q., Hu, Q., Huang, J. X., He, L., and An, W. (2017). Enhancing recurrent neural networks with positional attention for question answering. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 993–996.

Chen, Q., Hu, Q., Huang, J. X., and He, L. (2018a). CA-RNN: Using context-aligned recurrent neural networks for modeling sentence similarity. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*.

Chen, Q., Hu, Q., Huang, J. X., and He, L. (2018b). CAN: Enhancing sentence similarity modeling with collaborative and adversarial network. In *Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 815–824.

Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q., and Salakhutdinov, R. (2019). Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988.

Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2019). BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186.

Garg, S., Vu, T., and Moschitti, A. (2019). Tanda: Transfer and adapt pre-trained transformer models for answer sentence selection. *arXiv preprint arXiv:1911.04118*.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Huang, X. and Hu, Q. (2009). A bayesian learning approach to promoting diversity in ranking for biomedical information retrieval. In *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,*, pages 307–314.

Huang, X., Peng, F., Schuurmans, D., Cercone, N., and Robertson, S. E. (2003). Applying machine learning to text segmentation for information retrieval. *Information Retrieval*, 6(3-4):333–362.

Huang, X., Zhong, M., and Si, L. (2005). York university at TREC 2005: Genomics track. In *Proceedings of the Fourteenth Text REtrieval Conference, TREC*.

Kamath, S., Grau, B., and Ma, Y. (2019). Predicting and integrating expected answer types into a simple recurrent neural network model for answer sentence selection. In *20th International Conference on Computational Linguistics and Intelligent Text Processing*.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kwiatkowski, T., Palomaki, J., Redfield, O., Collins, M., Parikh, A., Alberti, C., Epstein, D., Polosukhin, I., Devlin, J., Lee, K., et al. (2019). Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Lai, T., Tran, Q. H., Bui, T., and Kihara, D. (2019). A gated self-attention memory network for answer selection. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 5955–5961.

Laskar, M. T. R., Hoque, E., and Huang, J. (2019). Utilizing bidirectional encoder representations from transformers for answer selection task. In *The V AMMCS International Conference: Extended Abstract*, page 221.

Liu, Y., An, A., and Huang, X. (2006). Boosting prediction accuracy on imbalanced datasets with SVM ensembles. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining, PAKDD*, pages 107–118.

Liu, Y., Huang, X., An, A., and Yu, X. (2007). ARSA: a sentiment-aware model for predicting sales performance using blogs. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 607–614.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Madabushi, H. T., Lee, M., and Barnden, J. (2018). Integrating question classification and deep learning for improved answer selection. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3283–3294.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Nakov, P., Màrquez, L., Magdy, W., Moschitti, A., Glass, J., and Randeree, B. (2015). SemEval-2015 task 3: Answer selection in community question answering. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval-2015)*, pages 269–281.

Nakov, P., Màrquez, L., Moschitti, A., Magdy, W., Mubarak, H., Freihat, A. A., Glass, J., and Randeree, B. (2016). SemEval-2016 task 3: Community question answering. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 525–545.

Nakov, P., Hoogeveen, D., Màrquez, L., Moschitti, A., Mubarak, H., Baldwin, T., and Verspoor, K. (2017). Semeval-2017 task 3: Community question answering. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 27–48.

Nema, P., Khapra, M. M., Laha, A., and Ravindran, B. (2017). Diversity driven attention model for query-based abstractive summarization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1063–1072.

Nishida, K., Saito, I., Nishida, K., Shinoda, K., Otsuka, A., Asano, H., and Tomita, J. (2019). Multi-style generative reading comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2273–2284.

Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing*, pages 1532–1543.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2227–2237.

Peters, M. E., Ruder, S., and Smith, N. A. (2019). To tune or not to tune? adapting pretrained representations to diverse tasks. In *Proceedings of the 4th Workshop on Representation Learning for NLP*, pages 7–14.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). Improving language understanding by generative pre-training.

Rao, J., Liu, L., Tay, Y., Yang, W., Shi, P., and Lin, J. (2019). Bridging the gap between relevance matching and semantic matching for short text similarity modeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 5373–5384.

Santos, C. d., Tan, M., Xiang, B., and Zhou, B. (2016). Attentive pooling networks. *arXiv preprint arXiv:1602.03609*.

Sha, L., Zhang, X., Qian, F., Chang, B., and Sui, Z. (2018). A multi-view fusion neural network for answer selection. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Tan, M., Santos, C. d., Xiang, B., and Zhou, B. (2015). LSTM-based deep learning models for non-factoid answer selection. *arXiv preprint arXiv:1511.04108*.

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 *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 695–704.

Tay, Y., Tuan, L. A., and Hui, S. C. (2018). Hyperbolic representation learning for fast and efficient neural question answering. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 583–591.

Tymoshenko, K. and Moschitti, A. (2018). Cross-pair text representations for answer sentence selection. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2162–2173.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Wan, S., Dras, M., Dale, R., and Paris, C. (2006). Using dependency-based features to take the'para-farce'out of paraphrase. In *Proceedings of the Australasian Language Technology Workshop 2006*, pages 131–138.

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

Wang, M., Smith, N. A., and Mitamura, T. (2007). What is the jeopardy model? a quasi-synchronous grammar for QA. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 22–32.

Wang, Z., Mi, H., and Ittycheriah, A. (2016). Sentence similarity learning by lexical decomposition and composition. *arXiv preprint arXiv:1602.07019*.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., and Brew, J. (2019). Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

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

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., and Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.

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

Yin, X., Huang, J. X., Li, Z., and Zhou, X. (2013). A survival modeling approach to biomedical search result diversification using wikipedia. *IEEE Transactions on Knowledge and Data Engineering*, 25(6):1201–1212.

Yu, X., Liu, Y., Huang, X., and An, A. (2012). Mining online reviews for predicting sales performance: A case study in the movie domain. *IEEE Transactions on Knowledge and Data Engineering*, 24(4):720–734.

Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., and Fidler, S. (2015). Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.