Original Content Is All You Need! An Empirical Study on Leveraging Answer Summary for WikiHowQA Answer Selection Task

Liang Wen[1,2] **Juan Li**[4] **Houfeng Wang**[1] **Yingwei Luo**[1,2]
Xiaolin Wang[1,2] **Xiaodong Zhang**[3] **Zhicong Cheng**[3] **Dawei Yin**[3]
[1]School of Computer Science, Peking University, China
[2]Peng Cheng Laboratory, Shenzhen, China [3]Baidu Inc., Beijing, China
[4]School of Chinese Language and Literature, Beijing Normal University, China
{yuco,wanghf,lyw,wxl}@pku.edu.cn {lijuan}@mail.bnu.edu.cn
{zhangxiaodong11,chengzhicong01}@baidu.com {yindawei}@acm.org

Abstract

Answer selection task requires finding appropriate answers to questions from informative but crowdsourced candidates. A key factor impeding its solution by current answer selection approaches is the redundancy and lengthiness issues of crowdsourced answers. Recently, Deng et al. (2020) constructed a new dataset, *WikiHowQA*, which contains a corresponding reference summary for each original lengthy answer. And their experiments show that leveraging the answer summaries helps to attend the essential information in original lengthy answers and improve the answer selection performance under certain circumstances. However, when given a question and a set of long candidate answers, human beings could effortlessly identify the correct answer without the aid of additional answer summaries since the original answers contain all the information volume that answer summaries contain. In addition, pretrained language models have been shown superior or comparable to human beings on many natural language processing tasks. Motivated by those, we design a series of neural models, either pretraining-based or non-pretraining-based, to check wether the additional answer summaries are helpful for ranking the relevancy degrees of question-answer pairs on WikiHowQA dataset. Extensive automated experiments and hand analysis show that the additional answer summaries are not useful for achieving the best performance.

1 Introduction

Answer selection task in community question answering (cQA) has been a popular research topic in both academy and industry due to its practical importance. In recent years, neural attention-based approaches for this task can be roughly categorized into two primary types. One type of them (Han et al., 2019; Rücklé et al., 2019) attempts to enhance the interactions of different granularity between question and candidate answer using the widely-adopted compare-aggregate framework (Wang and Jiang, 2017). The another focuses on incorporating additional input information, such as user metadata information (Xie et al., 2020), the subject-body relationship of community questions (Wu et al., 2018), etc. However, real-life cQA datasets that contain open-domain and non-factoid questions usually come along with long multi-sentence answer texts and noise. As a result, many previous neural answer selection approaches that were primarily designed to retrieve short answers fall short of expectations in such cases (Cohen et al., 2018; Rücklé et al., 2019).

Recently, Deng et al. (2020) propose to leverage answer summaries to tackle the redundancy and lengthiness issues of original answers in long answer selection task. To this end, they created *WikiHowQA*, the first large-scale open-domain cQA dataset that contains lengthy answers coupled with its summaries written by community users for non-factoid questions starting with "**How to**". Instead of relying on crowdsourcing, *WikiHowQA* was generated based on the WikiHow summarization dataset (Koupaee and Wang, 2018) and the online *WikiHow* knowledge base[1]. An example from the dataset is shown in Table 1. From it, we can see that the candidate answer details a method for decorating a school locker. And the answer summary well summarizes the key points of the answer. Based on this perspective, Deng et al. (2020) suggest that we could make use of answer summaries to alleviate the answer redundancy and noise issue in long answer selection task.

However, though answer summaries are always much shorter and more concise than the answer texts being summarized, they present text information in an abridged form and do not include details. Hence, the way that leverages the answer summaries to alleviate the answer redundancy and noise issues may also lead to the neglect of details

[1]http://www.wikihow.com

Proceedings of the 29th International Conference on Computational Linguistics, pages 1546–1555
October 12–17, 2022.

Question	How to decorate a school locker ?
Answer	Do you want your locker to be calm and relaxing , to relieve you from those stressful classes ? Or do you want your locker to be fun and exciting , colorful , or maybe you want it to show some of your rocker spirit. If you do n't want to express anything, pick a theme. A cool theme could be 70 's, giraffe 's, dogs ... (1112 words in total)
Summary	Think about the feeling you want to express. Acquire all of the materials you will need to make your locker look amazing. Hang up some pictures. Put a sign on the door ...
Label	1

Table 1: An example from the *WikiHowQA* dataset. Here we only list one candidate answer and its summary.

not covered by answer summaries. Besides, unlike user metadata information introduced by Xie et al. (2020) or subject-body relationship of community questions used by Wu et al. (2018), answer summaries are not supplementary to original answers and the utilization of them won't bring any additional information gain from an information entropy perspective. Last but not least, when given a question and a set of long candidate answers, Humans can easily figure out the correct answer without the need of additional answer summaries. Thus, it's unclear what role do answer summaries actually play in the WikiHowQA answer selection task and, indeed, whether it's beneficial to import additional answer summaries.

In this paper, we aim to conduct an in-depth and comprehensive analysis of this dataset and explore whether answer summaries could be helpful for "how-to" answer selection task. We demonstrate that, without the aid of answer summaries, simple, carefully designed LSTM-based models and pretraining-based models could obtain high, state-of-the-art MAP score of 72.91% and 82.74% on the dataset respectively. We carry out a meticulous qualitative analysis on randomly-sampled instances to provide data on their difficulty and quality, and whether the utilization of answer summaries could improve human performance. We conclude that: (i) This answer selection task is relatively easy though it contains long multi-sentence answer texts. (ii) Answer summaries do not convey additional information content and are not helpful for boosting both model and human performance. (iii) This dataset is noisy due to its method of data creation.

2 The Answer Selection Task

The *WikiHowQA* dataset introduced in (Deng et al., 2020) is made from an online wiki-style community website – WikiHow[2]. The questions contained in the dataset are all non-factoid and start with "**How to**". For a specific question, its accepted answers are considered as correct, and negative candidates were collected by retrieving the accepted answers to relevant questions. Each answer written by community users details multiple steps of doing a procedural task for a specific how-to question. In addition, every candidate answer is associated with a short reference summary. To be specific, the answer selection task could be formally defined as follows:

Given a question q_i and a set of lengthy candidate answers $A_i = \left\{a_i^{(1)}, \ldots, a_i^{(K)}\right\}$, the goal is to select all correct answers from the candidate answer set. In the training stage, for each candidate answer $a_i^{(k)}$, a corresponding reference summary $s_i^{(k)}$ and a label $y_i^{(k)}$ that denotes whether the answer $a_i^{(k)}$ can answer the question q_i are provided. However, during the testing procedures, the relevancy degree of question-answer pairs must be measured without access to their answer summaries. That is to say, answer summary is only accessible in the training stage. The reason we follow the constraint is that we want to make a fair comparison with previous methods[3].

Table 2 provides the detailed statistics of the dataset. From it, we can see that the answer texts is extremely long (with an average answer length of more than 520 words). Whereas, the well-known answer selection dataset InsuranceQA introduced by Feng et al. (2015) only has an average answer length of 112 words. Even in the recent *Long Answer Selection* (LSA) benchmark introduced by Rücklé et al. (2019), which are featured in con-

[2]http://www.wikihow.com

[3]The *WikiHowQA* dataset could also be used as an answer summary generation benchmark.

Statistics	*WikiHowQA*		
	Train	Dev	Test
Questions	76,687	8,000	22,354
QA pairs	904,460	72,474	211,255
Avg Qlen	7.20	6.84	6.69
Avg Alen	520.87	548.26	554.66
Avg Slen	67.38	61.84	74.42

Table 2: Statistics of the *WikiHowQA* Dataset. The average question length (Avg Qlen) is the average number of tokens in a question. The same applies to answer and summary.

taining long multi-sentence answer texts, we could only observe an average answer length of less than 290 words.

3 Methods

In this section, we describe two types of methods we implemented. The first class of methods directly models the relevancy degrees of question-answer pairs without using any answer summaries information. The second class of methods is partially inspired by Deng et al. (2020). They jointly learn answer selection and answer summary generation so as to leverage short answer summary to aid in picking out long multi-sentence answer. Different from them, we explicitly exploit the relevancy degrees of question-summary pair to aid in modeling the interaction between question and answer. For each class of methods, we from one side build our model based on bidirectional LSTMs so as to make a fair comparison with previous approaches and from the other we build our model based on a pretrained language model - ALBERT (Lan et al., 2020) to advance the state of the art.

3.1 LSTM-ASM

In this subsection, we aim at building a neural model that scores each answer in a pool of candidate answers according to its relevancy in regard to the given question. Our model adopts bidirectional LSTMs as text encoders, and we name it as LSTM-based Answer Selection Model(LSTM-ASM). The framework can be described in the following steps.

Given a question $q = \{w_q^1, \ldots, w_q^n\}$, an answer candidate $a = \{w_a^1, \ldots, w_a^m\}$ and the corresponding label y, we first map each word to its embedding. Then, the question embeddings E_q and the answer embeddings E_a are fed into a pair of Bi-LSTM encoders to generate contextual embeddings $\hat{E}_q$, $\hat{E}_a$ respectively.

Next, to capture the interactions between all aspects of question q and answer a, we feed the contextual embeddings $\hat{E}_q$ and $\hat{E}_a$ into a matching layer. Here, the matching Layer mainly define four different multi-perspective matching operations: Full-Matching, Maxpooling-Matching, Attentive-Matching, and Max-Attentive-Matching. Each matching operation describes a way to match each time-step of $\hat{E}_q$ against all time-steps of $\hat{E}_a$ and match each time-step of $\hat{E}_a$ against all time-steps of $\hat{E}_q$. We define the four matching operations as introduced by (Wang et al., 2017) and refer readers to it for details.

After applying a matching layer, we obtain the question-aware answer representations $R_a \in \mathbb{R}^{m \times d}$ and the answer-aware question representations $R_q \in \mathbb{R}^{n \times d}$, where d is the size of representations. Finally, we apply another bidirectional LSTM encoders on the R_a and R_q individually to generate the question-aware contextual answer representations $H_a \in \mathbb{R}^{m \times d}$ and the answer-aware contextual question representations $H_q \in \mathbb{R}^{n \times d}$. The last time-step of R_q and H_q are concatenated to form a sketch vector G_q, which outlines the matching result in the perspective of question. We also obtain another sketch vector G_a, which outlines the matching result in the perspective of answer. The final aggregation vector G_{qa} used for prediction is the concatenation of G_q and G_a.

To optimize our answer selection model LSTM-ASM, we use the cross-entropy loss function:

$$\mathcal{L}_{qa} = -\left[y \log p_{qa} + (1-y) \log (1-p_{qa})\right] \quad (1)$$

where p_{qa} is the predicted probability:

$$p_{qa} = \text{softmax}\left(W_{qa} G_{qa} + b_{qa}\right) \quad (2)$$

Here, W_{qa} and b_{qa} are trainable parameters.

3.2 LSTM-ASMSY

Different from LSTM-ASM, here we design another answer selection model that is capable of making use of reference answer summaries as additional information during training. Since the additional answer summary information is only available during the training period, our model is carefully designed to be able to make predictions without answer summaries as inputs. For simplicity, we name this model as LSTM-based Answer Selection Model with Summary (LSTM-ASMSY). As depicted in Figure 1, LSTM-ASMSY is composed of two modules: a long answer selection module

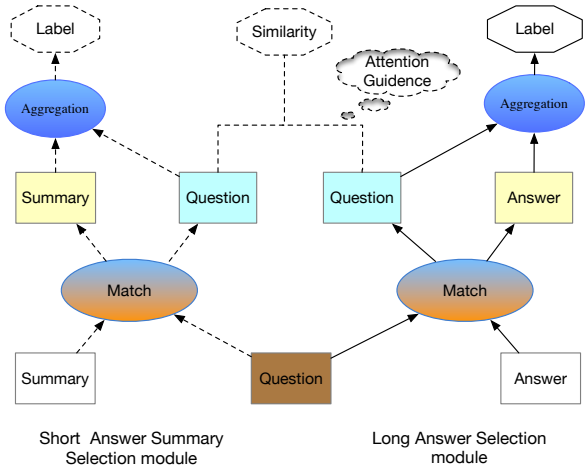

Figure 1: Architecture for LSTM-ASMSY.

and a short answer summary selection module. The short answer summary selection module adopts the same architecture as the LSTM-ASM model defined in Section 3.1. And it just plays a role in the training stage. Meanwhile, the long answer selection module also have a same architecture as the LSTM-ASM model but does not share any parameters with the short answer summary selection module. In the following, we detail how to train our LSTM-ASMSY model and in what way we can only use its long answer selection module to make predictions at inference time.

Given a question q, an answer a and its summary s, we first feed the long answer selection module with question q and answer a to obtain the sketch vector $G_q^{'}$ and the final aggregation vector $G_{qa}^{'}$ as in Section 3.1. Meanwhile, we feed the short answer summary selection module with the same question q and the corresponding answer summary s to get the sketch vector $\hat{G}_q$ and the aggregation vector G_{qs} in similar ways. Then, we induce the two modules to make the same prediction during training:

$$\mathcal{L}_{qa}^{'} = -\left[y \log p_{qa}^{'} + (1-y) \log \left(1 - p_{qa}^{'}\right)\right] \quad (3)$$

$$\mathcal{L}_{qs} = -\left[y \log p_{qs} + (1-y) \log \left(1 - p_{qs}\right)\right] \quad (4)$$

where $p_{qa}^{'}$ is the predicted probability output by the long answer selection module, and p_{qs} is the predicted probability output by the answer summary selection module. They are calculated as:

$$p_{qa}^{'} = \text{softmax}\left(W_{qa}^{'} G_{qa}^{'} + b_{qa}^{'}\right) \quad (5)$$

$$p_{qs} = \text{softmax}\left(W_{qs} G_{qs} + b_{qs}\right) \quad (6)$$

Where $W_{qa}^{'}$, W_{qs}, $b_{qa}^{'}$ and b_{qs} are trainable parameters, y is the gold label.

Next, in consideration of the sketch vector $G_q^{'}$ represents how well all the aspects of question are related to answer a while the sketch vector $\hat{G}_q$ represents how well all the aspects of question are matched in the perspective of the corresponding answer summary s, we encourage the sketch vector $G_q^{'}$ and $\hat{G}_q$ to be as similar as possible:

$$\mathcal{L}_{smi} = \left\| G_q^{'} - \hat{G}_q \right\|^2 \quad (7)$$

In such ways, we can not only leverage the answer summaries to provide implicit attention guidance during training, but also use the trained long answer selection module for making predictions without relying on the short answer summary selection module.

Finally, the overall loss function to optimize is:

$$\mathcal{L} = \lambda_1 * \mathcal{L}_{qa}^{'} + \lambda_2 * \mathcal{L}_{qs} + \lambda_3 * \mathcal{L}_{smi} \quad (8)$$

where λ_1, λ_2, λ_3 are tuneable hyper-parameters.

3.3 Extensions to ALBERT

ALBERT has achieved the state-of-the-art performance on sequence pair classification task but it can only process at most 512 tokens. However, on the *WikiHowQA* answer selection task, the average length of answer texts is more than 520 words, where each word could be broken down into more than one sub-word token. Hence, it prevents us from directly using ALBERT on this task. Here, we describe a simple way to extend the original ALBERT for handling long-form text matching.

Different from previous methods, like Longformer (Beltagy et al., 2020), ETC (Ainslie et al., 2020) and Big Bird (Zaheer et al., 2020), that require pre-training a new language model, our method does not need to train a language model from scratch. To be specific, we simply extend the original ALBERT to have a larger position vocabulary. And we reuse all the pretrained parameters within the original ALBERT model except the position embeddings. Besides, we initialize the first 512 position embeddings with original position embeddings and leave the rest random. The extended ALBERT is named as ALBERT-based Answer Selection Model (ALBERT-ASM). And we use the final hidden vector corresponding to the first input token ([CLS]) as the aggregate representation for measuring the relevancy of question-answer pair.

The way we build an ALBERT-based model that is capable of making use of reference answer summary during training is similar to Section 3.2. Specifically, we apply one extended ALBERT model to model the relevancy of question-answer pair and use another ALBERT model to model the relevancy of question-summary pair. And we encourage the hidden vectors corresponding to the question tokens of the two models to be as similar as possible. Similarly, we name this model as ALBERT-ASMSY.

4 Experiments

4.1 Training Details

For training our LSTM-based models, we use 300-dimensional GloVe word embeddings (Pennington et al., 2014) and apply a bidirectional GRU (Chung et al., 2014) to obtain a 100-dimensional character-level embedding for each word. The hidden size for all Bi-LSTM is set to 200 and the dropout ratio is set to be 0.1. We truncate candidate answer and its summary to 900 tokens and 100 tokens respectively. During training, we use the popular Adam optimizer (Kingma and Ba, 2015) and set its learning rate to be 0.0005. The batch size is set to 48 per gpu and the hyper-parameter λ_1, λ_2 and λ_3 are set to 1. We apply early stopping based on the evaluation result on the validation set. The maximum number of epochs is set to 20 and the patience is set to 5.

For training our ALBERT-based models, we initialize our models using the pretrained Albert-base-v1 model [4]. The maximum input sequence length of ALBERT-ASM is set to be 512, and the maximum input sequence length of ALBERT-ASMSY is set to be 1536, which is three times as long as the original maximum input length. We update our model using a batch size of 64 per gpu. And we adopt the AdamW (Loshchilov and Hutter, 2019) as our optimizer. Besides, we set the learning rate to be 3e-5 and the gradient clipping parameter to be 1.0. The maximum number of epochs is set to 3 for all the experiments. The hyper-parameter λ_1, λ_2 and λ_3 are set to 1.

4.2 Metrics

The performance of our models is measured in Mean Reciprocal Rank (MRR) and Mean Average Precision (MAP), which are standard metrics in Information Retrieval and Question Answering. The MRR measures the rank of *any* correct answer, while MAP examines the ranks of *all* the correct answers. Generally, the higher the scores, the better performance the model has.

4.3 Experiment Setups

We mainly compare our approaches against the following baselines:

(1) *Long answer selection methods*: CA is a widely-adopted compare-aggregate baseline for matching sequence pairs (Wang and Jiang, 2017). COALA is a recent baseline proposed by Rücklé et al. (2019), which has been proven to be effective in long answer selection task.

(2) *Two-Stage methods*: QPGN+AP-BiLSTM, QPGN+CA, and QPGN+COALA are three Two-Stage baselines, which first summarize the original lengthy answers and then conduct answer selection over the short generated answer summaries. Here, the QPGN is a question-driven pointer-generator network proposed by Deng et al. (2020), which is used to generate answer summaries for answer selection. AP-BiLSTM (Santos et al., 2016), CA (Wang and Jiang, 2017), COALA (Rücklé et al., 2019) are adopted answer selection models.

(3) *Joint Learning methods*: ASAS (Deng et al., 2020) is the recent state-of-the-art model that tackles the tasks of answer selection and answer summary generation in a joint manner.

For our own approaches, we evaluate the following models: LSTM-ASM, LSTM-ASMSY, ALBERT-ASM, and ALBERT-ASMSY. Besides, we also assess the performance of our models when they are fed with gold answer summary at test on purpose.

4.4 Main Results

Table 3 presents experiment results. In this table, we detail all model-specific inputs and their performance for each model. For example, our model LSTM-ASMSY jointly train an answer summary selection module and a long answer selection module during training and only adopt the trained long answer selection module for making predictions in consideration of the unavailability of answer summary at test. Hence, the input to LSTM-ASMSY during training is a question-answer-summary triplet while its input at test is a question-answer pair.

From Table 3, we mainly note the following observations: (1) LSTM-ASM achieve a new state-of-the-art result with an improvement of 18.67 MRR

[4] https://huggingface.co/albert-base-v1

Models	Inputs (training stage)	Inputs (test stage)	MAP	MRR
CA	Q, A	Q, A	50.22	52.14
COALA	Q, A	Q, A	50.03	51.96
QPGN+AP-BiLSTM	Q, A, S	Q, Generated S	52.37	53.43
QPGN+CA	Q, A, S	Q, Generated S	52.46	53.73
QPGN+COALA	Q, A, S	Q, Generated S	51.97	53.02
ASAS	Q, A, S	Q, A	55.22	56.86
Ours: LSTM-ASM	Q, A	Q, A	**72.91**	**75.53**
Ours: LSTM-ASMSY	Q, A, S	Q, A	72.40	74.97
Ours: ALBERT-ASM	Q, A	Q, A	**82.74**	**85.01**
Ours: ALBERT-ASMSY	Q, A, S	Q, A	82.65	84.97
Ours: LSTM-ASM†	Q, A	Q, S	64.99	67.75
Ours: LSTM-ASMSY†	Q, A, S	Q, S	64.42	67.20
Ours: ALBERT-ASM†	Q, A	Q, S	73.78	76.66
Ours: ALBERT-ASMSY†	Q, A, S	Q, S	73.69	76.62
Ours: LSTM-ASM†	Q, S	Q, S	65.77	68.50
Ours: ALBERT-ASM†	Q, S	Q, S	75.51	78.37

Table 3: Performance of different models on the *WikiHowQA* answer selection task. Here, *Q*, *A* and *S* denotes question, answer and summary respectively. Results marked † are the performances of our models when we purposely feed them with *gold answer summary* during test.

and 17.69 MAP over the best reported results in (Deng et al., 2020), which they obtained with a joint learning method, named ASAS. This demonstrates that LSTM-ASM can serve as a new strong baseline that uses LSTM as text encoders on this task. (2) LSTM-ASM has a better performance than LSTM-ASMSY and ALBERT-ASM also outperforms ALBERT-ASMSY, which show that making use of the additional answer summaries does not help to solve the answer selection problem. (3) From the last two rows of Table 3, we could see that, if we train our model LSTM-ASM and ALBERT-ASM with question-summary pairs and also test them using question-summary pairs [5], we could see a significant performance drop. This signifies that answer summaries are less informative than original lengthy answers when modeling the relevancy degrees of question-answer pairs. Also, it explains that leveraging additional answer summaries information may not result in a performance gain. (4) ALBERT-ASM and ALBERT-ASMSY have a much better performance than all LSTM-based models. This shows that better contextualized word representations brought by pretrained language models could be very effective in this long answer selection task. Meanwhile, these results also indicate that our method of extending ALBERT to do long answer selection is effective. (5) From row 11 to row 14, we can see that, when we train our models as usual but feed them with gold answer summary at test, we can observe severe performance degradation. This which further verifies that, question aspects covered by original answers are not always covered by their corresponding answer summaries to some extent.

Models	MAP	MRR
LSTM-ASM		
Full model	**72.91**	**75.53**
Max_tokens: 200	69.70	72.70
Max_tokens: 400	71.06	73.74
Max_tokens: 600	72.67	75.27
ALBERT-ASM		
Full model	**82.74**	**85.01**
Max_tokens: 512	81.79	84.05

Table 4: Model performances as a function of the length of input answers (averaged over three runs with different random seeds). "Max_tokens" denotes the maximum number of tokens of original lengthy answers to keep.

[5]Here, we do it on purpose for checking the informativeness of answer summary.

Annotation Source	Accuracy
question-answer pairs	**93%**
question-summary pairs	78%

Table 5: Human performances on the 100 sampled examples. The accuracy is the ratio of correct annotation.

In Table 4, we present the performances of our models as a function of the length of answers. From it, we can observe that the maximum length limit of answers has a big impact on the performance of our models. Both LSTM-ASM and ALBERT-ASM tend to exhibit better performance when we increase the maximum length limit of answers. This indicates that the *WikiHowQA* answer selection task does require the ability to deal with long answer.

5 Data Analysis

5.1 Human Performance

We randomly sampled 100 examples, namely 100 question-answer-summary triplets, from the test portion of the dataset for analysis. We first transform the 100 question-answer-summary triplets into 100 question-answer pairs and 100 question-summary pairs. Then, we divide our annotators into two groups, namely group 1, group 2. Each group consists of two annotators. Finally, we ask annotators of group 1 to label the question-answer pairs and ask annotators of group 2 to label the corresponding question-summary pairs [6]. Since the *WikiHowQA* dataset was created in an automatic and heuristic way, to make a fair evaluation, we also ask a graduate student who majors in linguistics to annotate the 100 examples and use these annotations as gold labels.

Table 5 show the annotation results. From it, we can see that annotators that use question-answer pairs as annotation source, are able to achieve an accuracy of 93% on this sampled subset. Whereas, when annotators use question-summary pairs as annotation source, they only obtain an accuracy of 78%, which is substantially lower than the previous one. This verifies that, on the *WikiHowQA* dataset, picking out the correct answers by measuring the relevancy degrees between question and its answer summary is much harder for human annotators.

In order to investigate whether additional answer summaries could help to boost the annotation performance, we also ask another two annotators to directly use the 100 question-answer-summary triplets as annotation source. After comparing these annotation results with the annotation results given by group 1, we find that the two results are almost exactly the same everywhere. This shows that additional answer summaries can not assist in improving the annotation performance either.

[6] We eliminate the annotation divergence by following the inter-annotator agreement.

Gold	Reference	G1	G2	Percentage
0	0	0	0	66%
0	0	0	1	16%
0	0	1	1	1%
1	0	1	1	5%
1	1	0	1	1%
1	1	1	1	11%

Table 6: The percentage of each annotation category. Here, we omit the categories with zero proportion.

Questions: How to change your name *in ohio*?
Answer: *In florida*, the name change process starts with checking your criminal history. In order to do this, you must have your fingerprints submitted for a state and national criminal records check. The fingerprints will be taken by the *florida department* of law enforcement …
Summary: Have a background check. Gather information for the petition. Fill out and sign the petition. File your petition. Attend your hearing. ..
Gold Label: **0** Reference Label: **0** Group 1: **0** Group 2: **1**

Questions: How to buy *affordable furniture*?
Answer: *Sales* happen all the time at retailers, especially around the holidays. While major holidays have their own sales, *furniture retailers have especially large sales* around president' s day, labor day, and memorial day. Take advantage of these sales to score larger furniture items and matching sets. January and July are also *good times to shop*…
Summary: *Shop seasonally.* Check retail store websites. *Use coupons. Search for clearance sales.*
Gold Label: **1** Reference Label: **0** Group 1: **1** Group 2: **1**

Figure 2: Some examples of annotation divergence.

5.2 Comprehensive Analysis

To get a comprehensive understanding of the above phenomenon, we further conduct an in-depth analysis. Specifically, for each example, we make a comparison among its gold label (Gold), its reference label (Reference, given by the dataset itself), its label annotated by group 1 (G1) and its label annotated by group 2 (G2). Table 6 provides our estimate of the percentage for each category.

From Table 6, we have the following observations: (i) For most of the examples, the gold labels, the reference labels, the labels annotated by group 1, and the labels annotated by group 2 are the same (All are zeros or ones), which indicates that the *WikiHowQA* answer selection task is relatively easy though it contains long multi-sentence answer texts. (ii) From the second row of table 6, we could see that 16 out of 100 samples are labeled correctly by annotators in group 1 but mislabeled by annotators in group 2. After carefully checking these examples, we observe an interesting finding from them. Specifically, most of these errors may have been caused by a lack of some specific details. A representative example of this category is presented at the top of Figure 2. From it, we can see that the answer summary explains how to change one's name in a general way while the answer explains "How to Change Your Name in Florida". This kind of annotation divergence is mainly due to the fact that the answer summary is relatively abstract and general while the answer contains all the clues. Therefore, we deem that the relevancy degree between question-summary pair is not enough for human annotators to make correct decisions in some cases. (iii) 5 examples of this sample set are labeled as '1' by annotators both in group 1 and group 2. And the reference labels of these examples are '0' but their gold label are '1'. By carefully checking these examples, we are confident that human annotators are correct. This clearly shows that the dataset is noisy to a certain extent. We present one of the representative examples at the bottom of Figure 2. From the example, we could see that both the answer candidate and the answer summary describe a way to shop for furniture on a reasonable budget, which exactly answer the question "How to buy affordable furniture ?". However, the reference label given by the dataset is '0', which largely may be due to that the dataset itself was created in an automatic and heuristic way.

6 Related Work

Long Answer Selection WikiPassageQA (Han et al., 2019) and LSA (Rücklé et al., 2019) are the two most well-known long answer selection benchmarks. On WikiPassageQA dataset, the state-of-the-art, non-pretraining-based method is proposed by Han et al. (2019). In their method, they derive contextualized uni-gram representation from n-grams and demonstrate that enabling multi-granular matches between question and answer n-grams are the key factors. On LSA benchmark, Rücklé et al. (2019) shows that a relevance matching approach based on the compare-aggregate framework with a coverage-based constraint works best among various LSTM-based methods.

In the midst of the pretraining-based methods, the best one is a self-supervised text matching model (Rücklé et al., 2020) which incorporates self-supervised with supervised multi-task learning on 140 source domains. It achieves state-of-the-art performances on both WikiPassageQA dataset and LSA benchmark.

Analysis of QA Tasks Several studies have investigated aspects of the design of QA datasets. Chen et al. (2016) conduct an examination of the CNN/Daily Mail reading comprehension dataset and conclude that this dataset is quite noisy and the required reasoning and inference level of this dataset is very simple. Sugawara et al. (2017) propose two classes of metrics (prerequisite skills and readability) to the quality of reading comprehension dataset. And they find that the readability of reading comprehension datasets does not directly affect the question difficulty. Yue et al. (2020) carry out a thorough analysis of the emrQA dataset (Pampari et al., 2018). And they discover that, though Pampari et al. (2018) claims that 39% of the questions may need knowledge to answer, their analysis shows that only a very small portion of the errors (2%) made by a state-of-the-art model might result from missing external domain knowledge.

In cQA, Liu et al. (2008) do a comprehensive analysis of questions and answers on cQA services and find that some questions usually have several best answers. And they show that customized question-type focused summarization techniques helps to improve cQA answer quality. Yang et al. (2011) analyze the not-answered questions in cQA and give a try on making predictions whether questions will receive answers.

7 Conclusion

In this paper, we carefully study the recent *WikiHowQA* answer selection task. Our models, either LSTM-based or ALBERT-based, outperform the previous state-of-the-art method by a large margin. More importantly, we do a careful hand-analysis of a small subset of the dataset. Overall, we think the *WikiHowQA* dataset is a valuable dataset, which provides a promising avenue for research on non-

factoid, long answer selection task. Nevertheless, we argue that: (i) this dataset is still noisy due to its method of data creation. (ii) For "how-to" answer selection task, the additional answer summaries can neither help to improve model performance nor can effect human annotation. (iii) the answer selection task is relatively easy though it contains long multi-sentence answer texts.

Acknowledgement

We thank all the reviewers for their valuable comments to improve this paper. The work is supported by National Natural Science Foundation of China (Grant No.62036001, No.62032001) and PKU-Baidu Fund (No. 2020BD021). The corresponding author of this paper is Houfeng Wang.

References

Joshua Ainslie, Santiago Ontanon, Chris Alberti, Vaclav Cvicek, Zachary Fisher, Philip Pham, Anirudh Ravula, Sumit Sanghai, Qifan Wang, and Li Yang. 2020. Etc: Encoding long and structured inputs in transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 268–284.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Danqi Chen, Jason Bolton, and Christopher D. Manning. 2016. A thorough examination of the cnn/daily mail reading comprehension task. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Daniel Cohen, Liu Yang, and W Bruce Croft. 2018. Wikipassageqa: A benchmark collection for research on non-factoid answer passage retrieval. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1165–1168.

Yang Deng, Wai Lam, Yuexiang Xie, Daoyuan Chen, Yaliang Li, Min Yang, and Ying Shen. 2020. Joint learning of answer selection and answer summary generation in community question answering. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7651–7658. AAAI Press.

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

Hojae Han, Seungtaek Choi, Haeju Park, and Seungwon Hwang. 2019. Micron: Multigranular interaction for contextualizing representation in non-factoid question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5892–5897.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Mahnaz Koupaee and William Yang Wang. 2018. Wikihow: A large scale text summarization dataset. *arXiv preprint arXiv:1810.09305*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Yuanjie Liu, Shasha Li, Yunbo Cao, Chin-Yew Lin, Dingyi Han, and Yong Yu. 2008. Understanding and summarizing answers in community-based question answering services. In *COLING 2008, 22nd International Conference on Computational Linguistics, Proceedings of the Conference, 18-22 August 2008, Manchester, UK*, pages 497–504.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Anusri Pampari, Preethi Raghavan, Jennifer J. Liang, and Jian Peng. 2018. emrqa: A large corpus for question answering on electronic medical records. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2357–2368. Association for Computational Linguistics.

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

Andreas Rücklé, Nafise Sadat Moosavi, and Iryna Gurevych. 2019. Coala: A neural coverage-based approach for long answer selection with small data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6932–6939.

Andreas Rücklé, Jonas Pfeiffer, and Iryna Gurevych. 2020. Multicqa: Zero-shot transfer of self-supervised text matching models on a massive scale. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 2471–2486. Association for Computational Linguistics.

Cicero dos Santos, Ming Tan, Bing Xiang, and Bowen Zhou. 2016. Attentive pooling networks. *arXiv preprint arXiv:1602.03609*.

Saku Sugawara, Yusuke Kido, Hikaru Yokono, and Akiko Aizawa. 2017. Evaluation metrics for machine reading comprehension: Prerequisite skills and readability. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 806–817. Association for Computational Linguistics.

Shuohang Wang and Jing Jiang. 2017. A compare-aggregate model for matching text sequences. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Zhiguo Wang, Wael Hamza, and Radu Florian. 2017. Bilateral multi-perspective matching for natural language sentences. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 4144–4150.

Wei Wu, Xu Sun, and Houfeng Wang. 2018. Question condensing networks for answer selection in community question answering. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 1746–1755. Association for Computational Linguistics.

Yuexiang Xie, Ying Shen, Yaliang Li, Min Yang, and Kai Lei. 2020. Attentive user-engaged adversarial neural network for community question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 9322–9329.

Lichun Yang, Shenghua Bao, Qingliang Lin, Xian Wu, Dingyi Han, Zhong Su, and Yong Yu. 2011. Analyzing and predicting not-answered questions in community-based question answering services. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2011, San Francisco, California, USA, August 7-11, 2011*. AAAI Press.

Xiang Yue, Bernal Jimenez Gutierrez, and Huan Sun. 2020. Clinical reading comprehension: A thorough analysis of the emrqa dataset. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 4474–4486. Association for Computational Linguistics.

Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big bird: Transformers for longer sequences. *arXiv preprint arXiv:2007.14062*.