CMQA: A Dataset of Conditional Question Answering with Multiple-Span Answers

Yiming Ju*,1,2, Weikang Wang*,3, Yuanzhe Zhang1,2, Suncong Zheng2, Kang Liu1,2, Jun Zhao1,2,
1 National Laboratory of Pattern Recognition, Institute of Automation, CAS, Beijing, China
2 School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China
3 Machine Learning Platform Department, Tencent

{yiming.ju, yzzhang, kliu, jzhao}@nlpr.ia.ac.cn
{daxianwang, congzheng}@tencent.com

Abstract

Forcing the answer of the Question Answering (QA) task to be a single text span might be restrictive since the answer can be multiple spans in the context. Moreover, we found that multi-span answers often appear with two characteristics when building the QA system for a real-world application. First, multi-span answers might be caused by users lacking domain knowledge and asking ambiguous questions, which makes the question need to be answered with conditions. Second, there might be hierarchical relations among multiple answer spans. Some recent span-extraction QA datasets include multi-span samples, but they only contain unconditional and parallel answers, which cannot be used to tackle this problem. To bridge the gap, we propose a new task: conditional question answering with hierarchical multi-span answers, where both the hierarchical relations and the conditions need to be extracted. Correspondingly, we introduce CMQA, a Conditional Multiple-span Chinese Question Answering dataset to study the new proposed task. The final release of CMQA consists of 7,861 QA pairs and 113,089 labels, where all samples contain multi-span answers, 50.4% of samples are conditional, and 56.6% of samples are hierarchical. CMQA can serve as a benchmark to study the new proposed task and help study building QA systems for real-world applications. The low performance of models drawn from related literature shows that the new proposed task is challenging for the community to solve. CMQA can be accessed at https://github.com/juyiming/CMQA.

1 Introduction

Question answering (QA) is a challenging benchmark task, which can drive the development of natural language understanding (NLU) methods and has significant utility to users (Kwiatkowski et al., 2019). This research area has made significant progress with many sizable datasets and standard benchmarks. Notably, the span-extraction task is one of the most studied subtasks because of its wide applicability and easy evaluation characteristics.

Most existing span-extraction QA datasets (Rajpurkar et al., 2016; Yang et al., 2018; Choi et al., 2018; Kwiatkowski et al., 2019; Chen et al., 2020) only contain single-span samples, where the answer is a single text span in the context. However, limiting the answer to be a single span in the context might be restrictive since the answer of some questions can be multiple text spans in real-world QA scenarios. Take the case in Figure 1 for example, given the question asking about ‘drugs for face acne redness’, the corresponding drugs (green and blue highlighted text spans) appear in different parts of the context. A single span including all these drugs is extremely long, thus it is more suitable to use multi-span answers to answer this question. Moreover, when building the QA system for a real-world application, we found that the multi-span answers often appear with two characteristics:

First, multi-span answers can be caused by users lacking related domain knowledge and asking ambiguous questions. In such cases, questions might need multiple conditions to specify the circumstance. For example, in Figure 1, the user asked a brief question about ‘drugs for acne redness’. Since different drugs are needed according to the severity of symptoms: (‘A single appearance’ and ‘the acne is much and continuous into pieces’), the question is answered separately. It is misleading to give all answer spans without distinguishing the conditions. Moreover, answers of different conditions might be contradictory in some samples. For example, a user asked about the battery capacity of the iPhone. Since the battery capacity of the iPhone is different according to models, the multi-span an-
Question: What are the anti-inflammatory drugs for face acne redness?

Context: Acne redness is generally the acute stage of acne. A single appearance indicates that the symptoms are not serious and can be treated with anti-acne drugs containing antibiotics. External antibiotics treat acne by inhibiting or killing Propionibacterium acne, reducing the content of free fatty acids, inhibiting the production of inflammatory chemokines and cytokines. Clindamycin phosphate gel and mupirocin ointment can be used for treatment. Both are used once or twice a day. Gently apply a layer of film on the area to be treated, and continue to use it for 3-4 weeks to evaluate the effect. During the illness, you need to pay attention to avoid squeezing the acne by hand. Especially when acne grows in the dangerous triangle area, it is strictly forbidden to squeeze to avoid ascending bacterial infection and cause cavernous sinus thrombophlebitis. You can use topical iodophor solution for treatment if the acne is much and continuous into pieces. A large area of redness indicates that the fungal infection is more serious and iodophor can kill fungus quickly. It is necessary to keep the local skin clean and hygienic. Wash your hands frequently, and fungicidal liquid soap is recommended.

Text spans:
- condition: 1, 2; coarse: 3; fine: 4, 5, 6;
- Relations: condition-answer: 1-3, 2-6; coarse-fine: 3-4, 3-5

Structured Answer:
- A single appearance
- anti-acne drugs containing antibiotics
- Clindamycin phosphate gel
- mupirocin ointment
- the acne is much and continuous into pieces
- topical iodophor solution

Figure 1: An example of labels in CMQA. Text spans: condition (red), coarse (blue) and fine (green). Relations: condition-answer and coarse-fine. The example is translated from Chinese.

...swers will contradict each other without specifying the condition.

Second, multiple answers in a sample might belong to different granularities, and there are often hierarchical relations among them. For example, in Figure 1, the answer span ‘topical anti-acne drugs containing antibiotics’ is a type of drug, while ‘Clindamycin phosphate gel’ and ‘mupirocin ointment’ are two specific drug names of this type. If we only give all these answer spans in parallel, the user cannot get the granularity and hierarchical information of these drug names. In this case, we need to provide the answer granularity and the hierarchical relations for accuracy.

The analysis results in Table 4 show that about half of the multi-span samples in our data are conditional, and half of them are multi-granularity. The high proportion of these samples demonstrates that the summarized two characteristics might be widespread in some real-world QA scenarios. Though some recent span-extraction QA datasets include multi-span samples (Du et al., 2019; Dasigi et al., 2019; Zhu et al., 2020), they only contain unconditional and parallel answers, which cannot be used to tackle this problem. To bridge the gap, we propose a new task: conditional question answering with hierarchical multi-span answers, where both the hierarchical relations and the conditions need to be extracted.

In this paper, we introduce Conditional Multiple-span Chinese Question Answering dataset (CMQA) to track the new proposed task. Specifically, we pick out samples need to be answered with multiple text spans and use a new label strategy to annotate them. We labeled both answers and conditions if the sample is conditional. Moreover, the answer spans are labeled with different granularity: coarse and fine. fine means the answer is a specific thing, such as a specific time, person, to name a few. coarse means the answer span is a general term for a class of things, such as foods containing certain nutrients and people with certain characteristics. The hierarchical relations among answers of different granularities are also provided. As shown in Figure 1, labeled text spans in CMQA consist of three types: condition, coarse and fine (highlighted in the context). Labeled relations consist of two types: condition-answer and coarse-fine. A structured answer can be easily derived from these labels, which is clear and accurate. Furthermore, such labels are very helpful to reduce the burden of users if they want to read the full context.

The final release of CMQA consists of 7,861 multi-span QA pairs and 113,089 labels, where 50.4% of samples are conditional, and 56.6% are multi-granularity. In addition, we establish models as the baseline of CMQA. Traditional single-span QA models which search for the most likely token as the start/end of the answer are unsuitable for extracting multi-span answers, thus, the span extraction is cast as a sequence tagging problem (Segal et al., 2020). And we use competitive relation extraction methods drawn from related literature (Wu and He, 2019; Zhong and Chen, 2021) for predicting relations. Experimental results show that it is very difficult to extract all spans correctly in a sample. The error analysis shows that the main challenge is to judge whether a span of the correct entity type is an answer. Moreover, though the relation model can achieve high performance on some
traditional relation extraction datasets, the performance on our task is extremely poor, which makes giving structured answers like the case in Figure 1 difficult. The poor model performance demonstrates that the new proposed task is challenging for the community to solve.

Our contributions can be summarized as follows:

- We propose a challenge when solving questions from real-world users: conditional question answering with hierarchical multi-span answers.
- We introduce a new annotated Chinese question answering dataset: CMQA, which consists of questions that need to be answered with conditions and hierarchically answer spans.
- We establish models drawn from related literature as the baselines of CMQA. Experimental results show that the new proposed task is challenging.

2 Related Work

Most existing span-extraction QA datasets only contain single-span samples, which might be restrictive in some real-world QA scenarios. An important reason for this gap is the question collection methods when building the datasets. Question collection methods of QA datasets can generally be classified into two categories: creating questions by annotators and collecting questions from real-world users.

2.1 QA datasets with questions created by annotators.

Most span-extraction QA datasets consist of questions created by annotators. Some span-extraction QA datasets’ questions are written by annotators who have first read the context containing the answer. For example, SQuAD (Rajpurkar et al., 2016) tasks crowdworkers with asking up to 5 questions about each paragraph and highlighting the corresponding answer in the paragraph. HotpotQA (Yang et al., 2018) tasks crowdworkers with asking a question about two given paragraphs from different Wikipedia pages and providing the answers. Tasking annotators with questioning and answering at the same time is suboptimal, which might cause questioners to ask questions based on a text span in the context. The asked questions are often simple reformulations of sentences in the context.

<table>
<thead>
<tr>
<th>Field</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>healthcare</td>
<td>77.6%</td>
</tr>
<tr>
<td>education</td>
<td>13.5%</td>
</tr>
<tr>
<td>government affair</td>
<td>4.0%</td>
</tr>
<tr>
<td>food</td>
<td>1.4%</td>
</tr>
<tr>
<td>digital product</td>
<td>0.3%</td>
</tr>
<tr>
<td>planting</td>
<td>0.2%</td>
</tr>
<tr>
<td>others</td>
<td>2.1%</td>
</tr>
</tbody>
</table>

Table 1: The fields of questions in the QA community.

To avoid this problem and create more natural and challenging questions, NewsQA (Trischler et al., 2017) only provides news article’s headlines and its summary points to questioners. The full context is unseeable. Similarly, QuAC (Choi et al., 2018) also prevents the questioner from seeing the full context. Though effort has been made, there are inevitably differences between the human-created questions and the questions asked by users in real-world scenarios. The concerns are as follows: First, the user’s question is often not based on a certain context in real-world QA scenarios. Second, crowdworkers might already know the form of the answer (single text span), which hints that they should ask questions with a single text span answer. Last, the problems created are often high quality, while a real-world question might sometimes be ambiguous.

2.2 QA datasets with questions collected from real-world users.

Besides manually created questions, some QA datasets use questions collected from queries from search engines. The answer form of these datasets is usually free text instead of text spans. For example, MS Marco (Nguyen et al., 2016) contains queries sampled from the Bing search engine. DuReader (He et al., 2018), which contains queries from Baidu search logs, chooses the free-text answer form, both use the free-text answer form and BLEU (Papineni et al., 2002) score as the evaluation metric.

Though there are datasets with questions from search engines that use text span as the answer form, the span length is usually longer than phrase level. Moreover, a screening mechanism is often used to filter questions. For example, WIKIQA (Yang et al., 2015) uses a single sentence as the answer form, and questions that cannot be answered with a single sentence are abandoned during the
construction. **Natural Questions** (Kwiatkowski et al., 2019) filter out questions cannot be answered with an entity or explanation. Moreover, Natural Questions provide long answers (paragraphs, tables, list items, or whole lists) besides short answers. A question might have a long answer but no short answer. These datasets show that a single span, especially a phrase-level text span, is not enough to answer real-world questions.

### 3 Dataset Construction

In this section, we describe the data construction process of CMQA.

#### 3.1 Data Source

Samples in CMQA are collected from an authoritative Chinese QA community, where experts in related fields provide answers to questions from users. The community covers a lot of fields, as shown in Table 1, in which the healthcare field has received the most questions from users.

Since the provided answers often contain much secondary information such as descriptions and supplements, such as the example in Figure 1, the length of some answers is rather long (64.7% are longer than 100 words). The long answer texts are inconvenient to users who want to browse quickly and get information. Thus, we want to provide a concise answer (short text span) besides the full text. However, due to the existence of multi-span samples and the two characteristics summarized in Section 1, many questions are not suitable to be answered with a single text span. Thus, we let the annotators pick out samples that cannot be answered with a single text span to tackle the problem. We get 8,864 filtered samples from 25,000 samples and build our dataset based on these samples. Note that we have discard samples that may reveal the personal information of the questioner or containing offensive content during this phase.

#### 3.2 Annotation Scheme

The annotation process is realized by three experts and three full-time annotators. As shown in Figure 2, the annotation scheme consists of 4 steps. Experts first formulate the annotation guidelines after pilot annotation. Then we train the annotators until all annotators think they can work independently. Then, we conduct an inter-annotator agreement (IAA) study to examine the annotation quality. We decide whether to continue the training phase according to the IAA score. In the last phase, we assign different parts of the samples to annotators for annotation.

**Annotation Principles** The main principles of annotation are summarized as follows:

- The text span should be as concise as possible.
- There is a multi-level hierarchical relationship, only the last level is marked.
- The sample can be abandoned in the following situations:
  - If the context cannot answer the question.
  - All answers are adjacent and can be regarded as one.
  - Most of the words in the context are part of the answer. It is better to give the entire context as a long answer in this case. (Because the number of the answer spans of each sample is variable, we did not give a specific context length as the threshold to filter out samples. Experts give examples that should be abandoned, and the annotators make their judgments based on these examples.)
  - The question is a combination of multiple sub-questions.

Moreover, when the experts conduct pilot annotation, we find the mismatch in **condition** occurs mostly due to inclusion/exclusion some boundary words. Since these boundaries are difficult to determine sometimes, we decide to expand the boundary of the **condition** to its nearest stop words based on the annotation. In the final release, the average length of **condition** increased from 10.4 to 13.4 after expansion.

---

Footnote: We has obtained the license of using the data from the community, which will be published with the final release.
Table 2: Comparison of CMQA with other QA datasets containing multi-span samples. Questions in QUOREF and DROP are collected by Mechanical Turk while questions in MASH-QA and CMQA are collected from real-world users. Answer amount refers to the average number of answer span per sample, and answer length refers to the average length per answer span. Note that only multi-span samples are compared.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Language</th>
<th>#QA</th>
<th>Conditional</th>
<th>Hierarchical</th>
<th>Question source</th>
<th>Answer amount</th>
<th>Answer length</th>
<th>Answer granularity</th>
</tr>
</thead>
<tbody>
<tr>
<td>QUOREF (Dasigi et al., 2019)</td>
<td>English</td>
<td>2K</td>
<td>-</td>
<td>-</td>
<td>crowdsourcing</td>
<td>2.5</td>
<td>1.6</td>
<td>word/phrase</td>
</tr>
<tr>
<td>DROP (Dua et al., 2019)</td>
<td>English</td>
<td>5K</td>
<td>-</td>
<td>-</td>
<td>crowdsourcing</td>
<td>2.5</td>
<td>2.0</td>
<td>word/phrase</td>
</tr>
<tr>
<td>MASH-QA (Zhu et al., 2020)</td>
<td>English</td>
<td>29K</td>
<td>-</td>
<td>-</td>
<td>real-world users</td>
<td>4.2</td>
<td>19.3</td>
<td>sentence</td>
</tr>
<tr>
<td>CMQA</td>
<td>Chinese</td>
<td>8K</td>
<td>✓</td>
<td>✓</td>
<td>real-world users</td>
<td>7.5</td>
<td>4.5</td>
<td>word/phrase</td>
</tr>
</tbody>
</table>

Table 3: Inter-annotator agreement scores. all refers to calculating the micro F1-score of all label types.

<table>
<thead>
<tr>
<th>Label type</th>
<th>IAA score</th>
</tr>
</thead>
<tbody>
<tr>
<td>condition</td>
<td>95.1</td>
</tr>
<tr>
<td>coarse</td>
<td>86.1</td>
</tr>
<tr>
<td>fine</td>
<td>90.0</td>
</tr>
<tr>
<td>condition-answer</td>
<td>98.3</td>
</tr>
<tr>
<td>coarse-fine</td>
<td>97.6</td>
</tr>
<tr>
<td>all</td>
<td>92.7</td>
</tr>
</tbody>
</table>

Table 4: The division of CMQA according to text span types.

<table>
<thead>
<tr>
<th>Span type</th>
<th>Proportion</th>
</tr>
</thead>
<tbody>
<tr>
<td>condition</td>
<td>50.43%</td>
</tr>
<tr>
<td>coarse</td>
<td>58.46%</td>
</tr>
<tr>
<td>fine</td>
<td>98.13%</td>
</tr>
</tbody>
</table>

We choose F1-score instead of other conventional metrics such as the kappa coefficient (Cohen, 1960) because there is no one-to-one correspondence between annotations from different annotators in multiple-span scenarios. Some Named Entity Recognition (NER) datasets (Balasuriya et al., 2009; Srirangam et al., 2019) use kappa coefficient by examining tags of all tokens. However, since most tokens in the context are not parts of the answers in the Question Answering task, such a strategy will lead to a very high agreement score that fails to reflect the annotation quality.

4 Data Analysis

In this section, we analyze the label and question properties of the new dataset. Table 2 shows the comparison of CMQA with other QA datasets containing multi-span samples.

4.1 Label Properties

There are five label types in CMQA: condition, coarse, fine, condition-answer and coarse-fine. Since the only principle of picking out data to construct the dataset is that the question cannot be answered with a single text span, a sample in our dataset does not necessarily contain all five label types. We investigate the properties of different label types.

Text Span Properties Table 4 shows the division of CMQA according to the included text span types. As shown in Table 4, 50.4% of the samples in CMQA are conditional, and 56.6% are
multi-granularity, which shows the necessity and effectiveness of our labeling strategy. Contexts of CMQA have an average of 182.3 words, with 18.8% annotated as conditions and 11.3% annotated as answers. The average amount and length of text spans are shown in Table 5.

### Relation Properties
Samples containing condition also contain related conditional answer spans, which means condition-answer is also included. However, containing condition doesn’t mean all answers in the sample are conditional. There are samples where part of answers are conditional while the others are not, which accounts for 24.8% of condition-containing samples. There might be hierarchical relations between answer spans of different granularity. There are 12,458 coarse answers in CMQA, and 63.1% of them are connected to fine answers. A coarse answer can connect to several fine answers. The average amount of relations per sample is shown in Table 5.

### 4.2 Question Properties
Questions in CMQA are collected from users in the QA community. We heuristically identified question types for these questions. Specifically, we first manually observe the questions and summarize some frequently occurring question words. Then we assign question types based on whether these words are included in the question. We visualize the distribution of question types in Figure 3. As shown in Figure 3, the majority of questions in CMQA are about detailed information about specific facts, such as expense, duration, and food.

### 5 Experiments
This section establishes models using methods drawn from related literature as the baseline of CMQA and analyzes the experiment results.
use a pretrained language model to obtain contextualized representations $h_i$ for each input token $t_i$. The representation $h_i$ is then fed into a feedforward network to predict the probability distribution of the tag type. Two different modes are used to get the prediction: 1) **Separate Mode** uses different models to predict text spans of different types. These models share the same model structures but are trained independently. 2) **Merged Mode** uses one model to predict all text spans and using multiple begin tags ($B_{\text{condition}}$, $B_{\text{coarse}}$ and $B_{\text{fine}}$) to distinguish span types.

**Relation Model** Because the type of relationship depends on the span type, we only need to predict if there is a dependency between a pair of text spans: $s_a$ and $s_b$. We build the relation model following previous work (Wu and He, 2019; Zhong and Chen, 2021), which gets competitive results in several relation extraction benchmarks, such as ACE04 and SciERC (Luan et al., 2018). Concretely, we first obtain contextualized representations $h_i$ for each input token $x_i$. Then we concatenate the token representations of the start positions of two text spans and obtain the span-pair representation: $h_{(s_a, s_b)} = [h_{\text{start}(a)}; h_{\text{start}(b)}]$, where $\text{start}(a)$ and $\text{start}(b)$ are the indices of start tokens of $s_a$ and $s_b$. Finally, the span-pair representation $h_{(s_a, s_b)}$ will be fed into a feed forward network to make the binary classification. Moreover, we further investigate using span boundaries to enhance the relation model (**span assist**), where additional markers are used to highlight all conditions and answer spans in a sample. Concretely, we insert special markers (such as ‘<fine>’ and ‘</fine>’) at the span boundaries in the input layer, and the embeddings of these markers are trainable vectors.

### 5.2 Evaluation Metrics and Experimental Details

We adopt two evaluation metrics: Exactly Match (EM) and F1-score. **EM:** Prediction of a sample is considered as correct the prediction is equal to the annotation, formally as $\text{anno}_{\text{pre}} = \text{anno}_{\text{ref}}$.

**F1 score:** The exact steps of calculating F1 scores are the same as that in the IAA study, shown in Algorithm 1 in the Appendix.

In all experiments, we use bert-base-chinese as the encoder to get contextual representations. The implementation is based on Hugging-Face’s *Transformers* library (Wolf et al., 2019). We report the averaged test set results of 3 runs for all the experiments. The relation models are trained with ground-truth span labels.

### 5.3 Experimental Results and Analysis

**Span Extraction** As shown in the experimental results summarized in Table 6, the separate and merged models get similar performance on the span extracting task. We can see from the table that model performance in EM is really poor, while the performance in F1 is much higher. An important reason is that there exist many simple spans besides those difficult to extract in one sample. For example, there might be answers connected in a one-word interval (e.g., ‘ginger, pepper, garlic ...’).

We analyze samples on the development set and find that 47.1% of fine-grained answers and 8.6% of coarse-grained answers are connected to others. In this case, the model will easily extract the others if one is recognized as the answer. However, to maintain the authenticity of the dataset, we did not eliminate or adapt these simple labels. The poor EM score in Table 6 shows that extracting all text span in one sample correctly is rather difficult. We further analyze model performance on judging if a certain span type is included in a sample. The results in Table 7 show the model can achieve very high performance (around 90% accuracy) on this

<table>
<thead>
<tr>
<th>Label type/Model</th>
<th>EM</th>
<th>F1</th>
<th>EM</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>condition</td>
<td>42.3</td>
<td>71.8</td>
<td>47.3</td>
<td>72.6</td>
</tr>
<tr>
<td>coarse</td>
<td>32.8</td>
<td>65.0</td>
<td>33.1</td>
<td>65.1</td>
</tr>
<tr>
<td>fine</td>
<td>44.5</td>
<td>84.3</td>
<td>42.7</td>
<td>83.6</td>
</tr>
<tr>
<td>all</td>
<td>19.3</td>
<td>79.4</td>
<td>20.5</td>
<td>79.0</td>
</tr>
</tbody>
</table>

Table 6: Model performance on span extraction. **all** refers to evaluate all span types together. We use the micro F1-score to calculate F1-score of **all**. Note that we only evaluate samples that contain the related span type in EM metric (all samples are evaluated in **all**).

<table>
<thead>
<tr>
<th>Model/Label type</th>
<th>condition</th>
<th>coarse</th>
<th>fine</th>
</tr>
</thead>
<tbody>
<tr>
<td>Separate</td>
<td>89.2</td>
<td>93.6</td>
<td>98.9</td>
</tr>
<tr>
<td>Merged</td>
<td>88.3</td>
<td>92.8</td>
<td>99.1</td>
</tr>
</tbody>
</table>

Table 7: Model performance on judging if a certain span type is included in the sample.
Question: How long does it take to recover after a needle stick?

Context: ... After topical application of chlorotetracycline eye drops, erythromycin eye ointment and hot compress, you will usually recover after 3-4 days from mild local red nodules. If the pus is formed locally, it usually breaks in 2-3 days, and it will recover in 5-6 days ...

Answer: 3-4 days, 5 to 6 days

Prediction: 3-4 days, 2-3 days, 5 to 6 days

Table 8: An example the merged model’s wrong predictions on the development set.

<table>
<thead>
<tr>
<th>Model/Label type</th>
<th>coarse</th>
<th>fine</th>
</tr>
</thead>
<tbody>
<tr>
<td>Separate</td>
<td>81.8%</td>
<td>77.2%</td>
</tr>
<tr>
<td>Merged</td>
<td>76.0%</td>
<td>80.6%</td>
</tr>
</tbody>
</table>

Table 9: Percentage of samples with distinction error in wrong-predicted samples from the development set.

<table>
<thead>
<tr>
<th>Label type/condition-answer</th>
<th>coarse fine</th>
<th>EM</th>
<th>F1</th>
<th>coarse fine</th>
<th>EM</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td></td>
<td>3.9</td>
<td>47.9</td>
<td>16.2</td>
<td>54.6</td>
<td></td>
</tr>
<tr>
<td>Span-assist</td>
<td></td>
<td>6.5</td>
<td>51.1</td>
<td>22.1</td>
<td>57.2</td>
<td></td>
</tr>
<tr>
<td>Normal</td>
<td></td>
<td>0.4</td>
<td>21.6</td>
<td>7.3</td>
<td>34.9</td>
<td></td>
</tr>
<tr>
<td>Span-assist</td>
<td></td>
<td>1.8</td>
<td>24.0</td>
<td>13.0</td>
<td>36.5</td>
<td></td>
</tr>
</tbody>
</table>

Table 10: Results of relation models. ♠: results of using the ground-truth span labels. ♦: results of using the prediction of the merged span model.

Relation Extraction: We report experimental results of two settings for the relation extraction task: using the ground-truth label as the input and using the prediction of the span model as the input. The results are shown in Table 10.

Due to the limited relation type in CMQA, the relation extraction task seems to be simpler than traditional relation extraction tasks containing multiple relation types (Augenstein et al., 2017; Luan et al., 2018; Gábor et al., 2018). Surprisingly, results in Table 10 show that the competitive method on these tasks performs poorly on CMQA. We hypothesize that one reason is that the text span and relation amount per sample in CMQA are higher than many traditional relation extraction datasets, which often focus on intra-sentence relations. And another reason is that the span type plays a very marginal role in relation extraction in CMQA. In contrast, the entity type is an important feature for judging relationships in most traditional relation extraction tasks (Zhong and Chen, 2021). These differences indicate new approaches needed to be developed to solve relation extraction in CMQA. Results in Table 10 show that introducing the boundary information of other spans can improve model performance. However, the improved model performance is still far from satisfying, which makes providing structured answers as Figure 1 very difficult.

6 Conclusion

In this paper, we propose a new challenge: conditional question answering with hierarchical multi-span answers, which might be widespread in multi-span QA in real-world scenarios. Moreover, we introduce CMQA, which contains conditional and hierarchical samples to study the new proposed task. Data analysis and experimental results show the main characteristics and challenges of CMQA, and the poor model performance demonstrates that
the proposed task is challenging for the community to solve. We believe CMQA can serve as a benchmark to study the new proposed task and help build more reliable and sophisticated QA systems.

Acknowledgements

This work is supported by the National Key Research and Development Program of China (No.2020AAA0106400), the National Natural Science Foundation of China (No.61922085, No.61976211, No.61906196) and the Key Research Program of the Chinese Academy of Sciences (Grant NO. ZDBS-SSW-JSC006). This work is also supported by TencentKG, the Open Project of Beijing Key Laboratory of Mental Disorders (2019JSJB06), Yunnan Provincial Major Science and Technology Special Plan Projects (No.202103AA080015) and in part of the Youth Innovation Promotion Association CAS.

References


A Appendix

A.1 Training Annotators

The annotation process is realized by three full-time annotators. All annotators have at least a high school degree and more than one year of full-time annotation working experience. We let the annotators annotate different parts of data according to the guidelines. All annotators and experts are in an online communication group. We encourage annotators to ask experts questions about samples they are not very sure of. We guarantee that experts will answer annotators’ questions immediately in this phase. It takes about three days until all annotators think they can work independently. Each annotator has annotated over 100 samples in this phase. Note that this part of data is only used for training and will not be included in the final release.

A.2 Label Amount

We visualize the amount of each label type in Figure 4, where the first subgraph shows the total label amount of the dataset, and the rest of the subgraphs show that of each sample.

A.3 The exact steps of calculating F1 scores for IAA study and Experiment Evaluation

Algorithm 1 describes the exact steps in the evaluation procedure of IAA study and experiments.
Algorithm 1 Evaluation Algorithm

Input: Anno\textsubscript{ref}: the reference annotation; Anno\textsubscript{pre}: the prediction annotation;
Output: $F_1$ score;

1: function CAL\_F1(Anno\textsubscript{ref}, Anno\textsubscript{pre})
2: recall\textsubscript{mol} ← 0
3: recall\textsubscript{den} ← 0
4: precision\textsubscript{mol} ← 0
5: precision\textsubscript{den} ← 0
6: for anno\textsubscript{ref}, anno\textsubscript{pre} ∈ zip(Anno\textsubscript{ref}, Anno\textsubscript{pre})
7:     for a ∈ anno\textsubscript{ref} do
8:         if a ∈ anno\textsubscript{pre} then
9:             recall\textsubscript{mol} ← recall\textsubscript{mol} + 1
10:         end if
11:     end for
12:     for a ∈ anno\textsubscript{pre} do
13:         if a ∈ anno\textsubscript{ref} then
14:             precision\textsubscript{mol} ← precision\textsubscript{mol} + 1
15:         end if
16:     end for
17:     recall\textsubscript{den} ← recall\textsubscript{den} + len(anno\textsubscript{1}).
18:     precision\textsubscript{den} ← precision\textsubscript{den} + len(anno\textsubscript{2})
19: end for
20: recall ← recall\textsubscript{mol}/recall\textsubscript{den}
21: precision ← precision\textsubscript{mol}/precision\textsubscript{den}
22: $F_1$ ← $2 \times $ recall $\times$ precision / ($$recall + precision$)
23: return $F_1$

Anno\textsubscript{n} = \{anno\textsubscript{1}, anno\textsubscript{2}, ..., anno\textsubscript{n}\} refers to the annotation for n samples.