Knowing-how & *Knowing-that*: A New Task for Machine Reading Comprehension of User Manuals

Hongru Liang[1] **Jia Liu**[2] **Weihong Du**[1]
dingnan jin[2] **Wenqiang Lei**[1*] **Zujie Wen**[2] **Jiancheng Lv**[1]
[1]College of Computer Science, Sichuan University
[2]Ant Group, China
{lianghongru, wenqianglei, lvjiancheng}@scu.edu.cn
duweihong@stu.scu.edu.cn
{jianiu.lj, dingnan.jdn, zujie.wzj}@antgroup.com

Abstract

The machine reading comprehension (MRC) of user manuals has huge potential in customer service. However, current methods have trouble answering complex questions. Therefore, we introduce the *Knowing-how* & *Knowing-that* task that requires the model to answer factoid-style, procedure-style, and inconsistent questions about user manuals. We resolve this task by jointly representing the sTeps and fActs in a gRAph (TARA), which supports a unified inference of various questions. Towards a systematical benchmarking study, we design a heuristic method to automatically parse user manuals into TARAs and build an annotated dataset to test the model's ability in answering real-world questions. Empirical results demonstrate that representing user manuals as TARAs is a desired solution for the MRC of user manuals. An in-depth investigation of TARA further sheds light on the issues and broader impacts of future representations of user manuals. We hope our work can move the MRC of user manuals to a more complex and realistic stage.

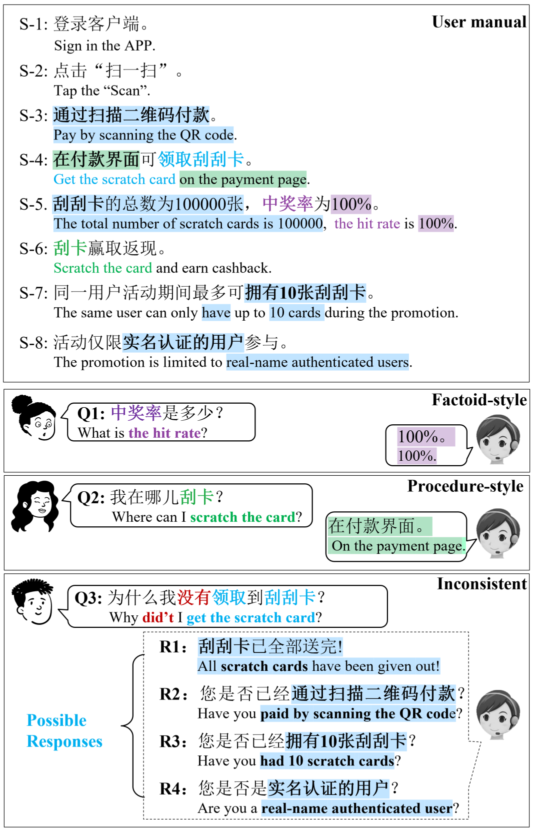

User manual
S-1: 登录客户端。
Sign in the APP.
S-2: 点击“扫一扫”。
Tap the “Scan”.
S-3: 通过扫描二维码付款。
Pay by scanning the QR code.
S-4: 在付款界面可领取刮刮卡。
Get the scratch card on the payment page.
S-5. 刮刮卡的总数为100000张，中奖率为100%。
The total number of scratch cards is 100000, the hit rate is 100%.
S-6: 刮卡赢取返现。
Scratch the card and earn cashback.
S-7: 同一用户活动期间最多可拥有10张刮刮卡。
The same user can only have up to 10 cards during the promotion.
S-8: 活动仅限实名认证的用户参与。
The promotion is limited to real-name authenticated users.
Factoid-style
Q1: 中奖率是多少？
What is the hit rate?
100%。
100%.
Procedure-style
Q2: 我在哪儿刮卡？
Where can I scratch the card?
在付款界面。
On the payment page.
Inconsistent
Q3: 为什么我没有领取到刮刮卡？
Why did’t I get the scratch card?
Possible Responses
R1：刮刮卡已全部送完!
All scratch cards have been given out!
R2：您是否已经通过扫描二维码付款？
Have you paid by scanning the QR code?
R3：您是否已经拥有10张刮刮卡？
Have you had 10 scratch cards?
R4：您是否是实名认证的用户？
Are you a real-name authenticated user?

Figure 1: A user manual and three QA cases about it

1 Introduction

User manuals are supposed to be helpful in using products, getting involved in promotions, or other goals of interest (Ryle, 2009; Chu et al., 2017; Bombieri et al., 2021). Though well-designed and instructive, they are seldom read by users because "*Life is too short to read manuals*" (Blackler et al., 2016). Towards high user satisfaction, professional customer service representatives (CSRs) are hired to do the reading and answer user questions about the manuals. The machine reading comprehension (MRC) of user manuals thus has huge potential, as it would not only reduce high labor costs but enable the service ready for customers 24/7.

Let's see how a CSR approaches various questions given the user manual[1] in Figure 1. Specifically, she can answer Q1 by looking up the arguments of facts, i.e., the value (100%) of the hit rate. For Q2, she needs to reason about the steps in the procedure as the answer is not explicitly exhibited in the user manual — the user manual directs the user to "Sign in" and "Scan", and eventually to the "payment page". After that, the subsequent description does not direct the user anywhere else. A user can infer from the procedure that one can scratch the card after immediately getting it on the payment page. Therefore, the location to scratch the card is on the payment page . The most difficult one is Q3, which raises a case inconsistent with the descriptions in the user manual — going along with the user manual (S-4), the user didn't get the scratch card. We call this type of question an inconsistent question. A possible response to

[*] Corresponding author

[1] We make a few modifications (e.g., removing the application name) to this real-world user manual for anonymity.

Findings of the Association for Computational Linguistics: ACL 2023, pages 10550–10564
July 9-14, 2023 ©2023 Association for Computational Linguistics

the inconsistent question may be an answer (R1) or a high-utility question leading to an answer (R2-R4). Towards such responses, the CSR needs to reason about not only the steps but the facts binding the steps — the inconsistency may be caused by incorrect operations in the previous step (e.g., the user didn't pay by scanning the QR code) or unsatisfiable constraints in the facts (e.g., the user already had 10 scratch cards). Therefore, aligned with the CSR, a practical MRC model must also be able to draw a unified inference across steps and facts to properly answer various user questions.

With a splendid library of studies (Rajpurkar et al., 2016; Zhu et al., 2020), current MRC models have shown strong power to answer factoid-style questions (e.g., Q1). More recent studies (Tandon et al., 2020; Goyal et al., 2021; Zhang et al., 2021) have mitigated the weakness of factoid-style MRC on procedure-style questions. However, these studies mainly focus on questions about the states of entities that can be answered by single spans. They have trouble answering more complex procedure-style questions with multiple-span answers or about the dynamics of actions (e.g., Q2), let alone inconsistent questions like Q3. We also notice that existing models are only designed for single types of questions. This doesn't meet the real-world scenarios where factoid-style, procedure-style, and inconsistent questions are all involved (cf., Table 4).

To address these limitations, we propose the ***Knowing-how* & *Knowing-that*** task[2] — given a user manual, the model is required to answer factoid-style, procedure-style or inconsistent questions about it. We resolve this task by jointly representing the sTePs and fActs in a heterogeneous gRAph (TARA), cf., Figure 2. This representation allows us to make the unified inference of various questions. We further benchmark the proposed task with a Heuristic method (HUM) to automatically represent User Manuals as TARAs and a densely-annotated dataset (OHO) to test the model's ability in answering real-world questions. Specifically, HUM is designed to be independent of labelled data with the ultimate goal of building industry-scale applications and particular attention to low costs and good generalization capability. The annotated dataset (OHO) is derived from Online Help dOcuments and real-world user questions from an e-commerce company. We provide the gold answers to user questions and gold TARAs of user manuals. Experiments demonstrate the superiority of TARA, the efficiency of HUM, and the significant challenges of OHO. With an in-depth investigation of TARA, we discuss the issues and broader impact. We expect our work can advance the MRC of user manuals and inspire future research on smart customer service. We highlight our work as follows:

- We introduce a new task and take the primary step to the MRC of user manuals in a more realistic setting, where various questions are involved.
- We resolve the task by jointly representing steps and facts in a graph. We also benchmark the task with an efficient method and densely annotate a testing set derived from real-world scenarios.
- The experiment reveals the superiority of the proposed representation and the significant challenges of the new task.
- We will release the dataset and codes to assist further studies once this paper is accepted.

2 Related Work

MRC for user manuals Earlier studies on the MRC of user manuals focus on factoid-style comprehension (Zhang et al., 2012; Mysore et al., 2019; Jiang et al., 2020; Nabizadeh et al., 2020). Recent research pays attention to the procedure-style comprehension of user manuals and aims to track the state values of pre-defined entities (Bosselut et al., 2017; Amini et al., 2020; Tandon et al., 2020; Goyal et al., 2021; Zhang et al., 2021). Although such models may answer some procedure-style questions, they are in trouble answering more complex questions like inconsistent ones. Several studies design structured representations for user manuals (Kiddon et al., 2015; Maeta et al., 2015; Vaucher et al., 2020; Kuniyoshi et al., 2020). However, they only represent the steps in user manuals. This largely limits the model's ability to answer questions that need inference about both steps and facts. Contrary to the above research, we study the MRC of user manuals that are suitable for factoid-style, procedure-style, and inconsistent questions and design a heterogeneous graph to represent both steps and facts. The representation supports the unified inference of various questions and can be constructed automatically by a heuristic method.

Datasets for user manuals We have done a thorough survey (cf., Table 1) about existing datasets

[2] We name it after Gilbert Ryle's thought that the possession of *Knowing-how* (i.e., procedural knowledge) and *Knowing-that* (i.e., factoid knowledge) is "a mark of intelligence" (Ryle, 2009).

Dataset	Types of user manuals	Number of user manuals	Number of annotations	QA pairs
Zhang et al. (2012)	recipes, technical manuals	74	1,979	✗
Mori et al. (2014)	recipes	266	19,939	✗
Kiddon et al. (2015)	recipes	133	Not Avaliable	✗
Yamakata et al. (2020)	recipes	300	Not Avaliable	✗
Jiang et al. (2020)	recipes	260	15,203	✗
Nabizadeh et al. (2020)	technical manuals	1,497	Not Avaliable	✗
Goyal et al. (2021)	technical manuals	1,351	6,350	✗
Zhong et al. (2020)	technical manuals	400 (sentences)	2,400	✗
Mysore et al. (2019)	scientific experiment	230	19,281	✗
Vaucher et al. (2020)	scientific experiment	1,764	∼4,755	✗
Kuniyoshi et al. (2020)	scientific experiment	243	23,082	✗
OHO	e-commercial helping documents	2,000	24,474	✓

Table 1: Comparisons between OHO with off-the-shelf datasets based on the statistics reported in the original paper

Term	Category	Explanation
Action	node	the action of a step performed by the user, presented as a verb, e.g., "pay".
Entity	node	the concepts in the user manual, e.g., "scratch card". Each user manual has a user Entity by default.
Action-ARG	argument	the modifier (MOD), time (TIME), location (LOC), and manner (MANN) arguments associated with an Action node.
Entity-ARG	argument	the arguments associated with an Entity node — footnote argument (FN), offering extra details; attribute argument (ATT), describing a specific aspect (e.g., the hit rate of the scratch card); state argument (STATE), describing a changeable state (e.g., the "have" state of the same user).
ARG-ARG	argument	the arguments associated with an Entity-ARG. The arguments of ATT are the same as Entity-ARG and the arguments of STATE is the same as Action-ARG.
NEXT	relation	the directed edge between two Action nodes, indicating the end action is the next step of the start one. It determines the order of the actions performed by the user.
AGT	relation	the directed edge from the user Entity to an Action, indicating the action is performed by the user.
PAT	relation	the directed edge from an Action to an Entity, indicating the action is performed on the entity.
SUB	relation	the directed edge between two Entity nodes, indicating the start entity is a sub-entity of the end one.
PATA	relation	the directed edge from a STATE to an Entity, indicating the entity is affected by the changing of STATE.

Table 2: Explanations about the nodes, relations, and arguments of TARA

for user manuals. The types of user manuals include recipes (Zhang et al., 2012; Mori et al., 2014; Kiddon et al., 2015; Yamakata et al., 2020; Jiang et al., 2020), technical manuals (e.g., device maintenance, surgical practices) (Zhang et al., 2012; Nabizadeh et al., 2020; Goyal et al., 2021; Zhong et al., 2020), and scientific experiment (Mysore et al., 2019; Vaucher et al., 2020; Kuniyoshi et al., 2020). There is a lack of a dataset about e-commerce scenarios where the MRC of user manuals is urgently needed due to the high labor costs for customer service. Besides, we have noticed that existing annotations are all about user manuals. None of them has offered real-world QA pairs to test the practical performance of MRC models. To address these limitations, we collect the OHO dataset with online helping documents from an e-commerce company. We then contribute annotations for data-driven evaluations of the model's ability to answer real-world questions and represent user manuals.

3 Representation of user manuals

Definitions We wish to design a representation for user manuals that supports the unified inference of various questions. It can be reduced to two fundamental tasks: how to extract actions (entities) and their arguments from the user manual, and what is the relation between an action and an entity (two actions or two entities). This motivates us to propose the TARA representation, which jointly describes steps and facts in a heterogeneous graph with two sets of nodes, three sets of arguments, and five sets of relations, as defined in Table 2. In this way, the user manual in Figure 1 is represented as a graph[3], part of which is displayed in Figure 2.

Unified inference We use TARA to represent Q1, Q2, and Q3, as shown in Figure 2. The inference of answers can be divided into two stages. First, we extract a sub-graph from the representation of the user manual that is most similar to the structure of the question. Second, we identify the answers by directly inferring the conflict arguments between the sub-graph and the question[4]. Specifically, if the

[3] Although the STATE of Entity shares the same arguments with an Action node, it cannot be upgraded into a node because it doesn't refer to an action and cannot be linked to other Action nodes. Similarly, the ATT of an Entity node cannot be a node because it can't be linked to other Entity nodes except for the parent one. Meanwhile, as the user dominates the whole user manual, it cannot be the target of any STATE.

[4] This operation reduces the cognition loads to identify Wh-word (e.g., when, what) and its variants in user questions.

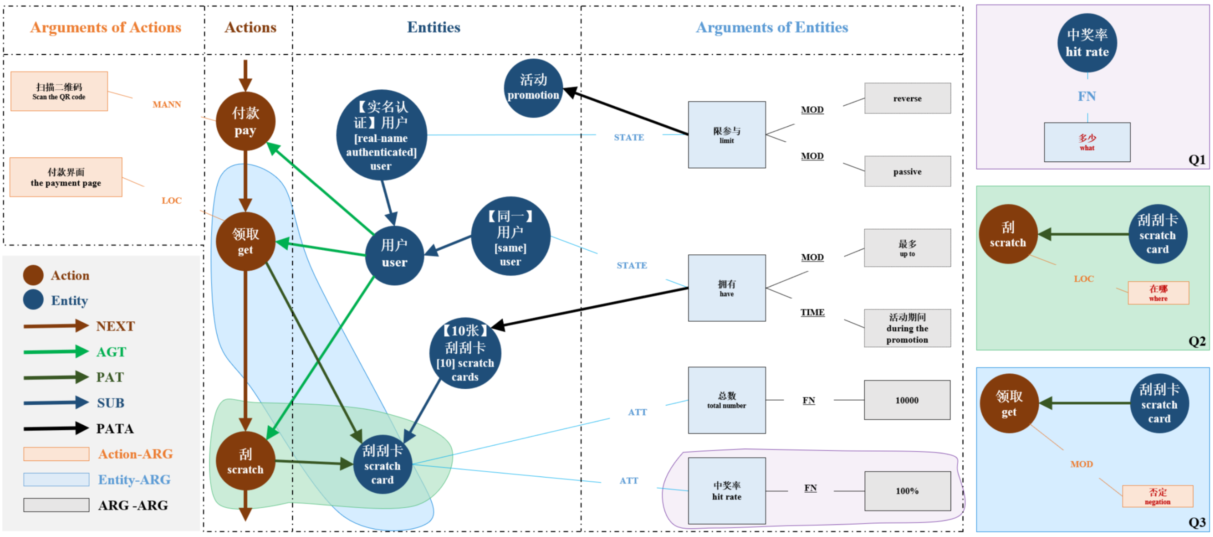


Figure 2: Screenshot of the graph of the user manual and the graphs of Q1, Q2, and Q3. The representation of a question and its most similar sub-graph are bounded in the same color and the conflict arguments are marked in red.

conflict is caused by the values of arguments, the answer is the value from the sub-graph. For example, the answer for Q1 is the FN value ("100%") of the hit rate. If the conflict is caused by the existence of Action-ARG, the answer is inferred from the nearest Action node, the same argument of which is not null, that can walk to the current Action node via NEXT relations. For example, the answer to Q2 is located in the LOC argument of the "get" node, which is linked to "scratch" via NEXT. The third type of conflict is the inconsistency of MOD arguments, namely, the Action node of the question has a MOD argument valued "negation". The answer is inferred from the nearest Action node and Entity nodes pointed to the sub-graph. For example, all possible responses to Q3 are obtained from the "pay" node, the "user" node, and the "scratch card" node. In summary, TARA allows the unified inference of factoid-style, procedure-style, and inconsistent questions by jointly representing steps and actions in the same graph.

4 The HUM Method

Towards practical industry-scale applications, it requires extra attention to labor costs and generalization. Thus, we choose to design an efficient method based on pre-trained semantic dependency parsing tools and domain-independent heuristics.

First, we segment the user manual sentence by sentence and feed the sentences into a pre-trained semantic dependence parsing tool. If a sentence is imperative, we leverage "user" as the subject of the sentence. For instance, "Sign in the APP" is modified as "User sign in the APP". Then, we eliminate the vertices to compose more meaningful phrases — the offsprings of a predicate's child are eliminated and combined with the child. An example[5] is shown in Figure 3. In addition, to maintain the leading role of the user, if the user is the patient of a predicate, we change its role with the agent and mark the predicate as "reverse".

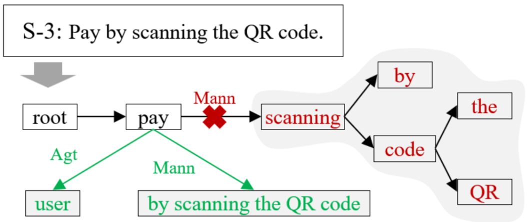


Figure 3: Modifications for the semantic dependency tree parsed from S-3. Red and green vertices are the eliminated and newly added vertices, respectively.

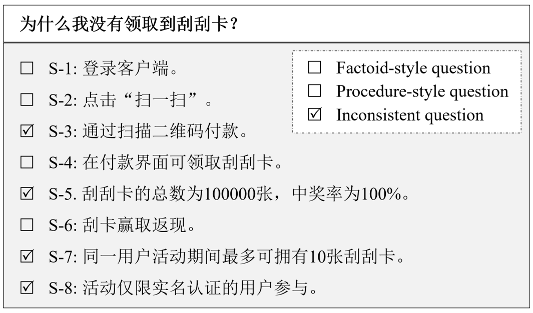
Second, we use the active verbs, whose agents are the user, as the Action nodes. These nodes are linked with the "NEXT" relations based on their order in the user manual and semantic dependencies. Besides the user node created by default, other Entity nodes are initialized as the unique patients with the user as agents[6]. The relation from the user Node to any Action node is "AGT". We also add an edge with the "PAT" relation from an Action node to the Entity node, whose value serves as the patient of the Action node. Particularly, if an entity has attributives, we create a new entity with the "SUB" relation to it by combining the attributives. The "same user" node is created in this

[5] To save space, some examples are only in one language.

[6] As the "scratch card" and "card" share the same semantics, we treat both of them as the "scratch card" entity. In addition, although "limit" is not an active verb, its patient ("promotion") is treated as an Entity node as its agent is the user.

ARGs	Tags
MOD	mDEPD, mTime, mRang, mDegr, mFreq, mDir, mNEG, mMod
TIME	Time, Tini, Tfin, Tdur, Trang
LOC	Loc, Lini, Lfin, Lthru, Dir
MANN	Mann, Tool, Matl, Accd
FN	LINK, Clas

Table 3: Alignment rules between the arguments and the tags from the semantic dependency parsing tools



为什么我没有领取到刮刮卡?

☐ S-1: 登录客户端。

☐ S-2: 点击“扫一扫”。

☑ S-3: 通过扫描二维码付款。

☐ S-4: 在付款界面可领取刮刮卡。

☑ S-5. 刮刮卡的总数为100000张，中奖率为100%。

☐ S-6: 刮卡赢取返现。

☑ S-7: 同一用户活动期间最多可拥有10张刮刮卡。

☑ S-8: 活动仅限实名认证的用户参与。

☐ Factoid-style question

☐ Procedure-style question

☑ Inconsistent question

Figure 4: Annotation of the answer to Q3

way. Most of Action-ARG, Entity-ARG, and ARG-ARG are generated by the alignment with the tags from off-the-shelf tools like LTP (Che et al., 2021), HanLP (He and Choi, 2020), etc. The alignment rules are defined in Table 3. Besides, if an entity acts as an attribute to an agent, we use the agent as an ATT argument to the entity. For example, “total number” is an ATT argument to the “scratch card” entity. We also use a predicate as the STATE argument of its agent when the agent is the user, but the predicate is a state verb (e.g., “have”) or when the agent is not the user.

Automated Inference for QA To infer the answers to user questions, we leverage the HUM method to represent the user manual and user question, simultaneously. Then, we extract the sub-graph that is most similar to the representation of the user question via the sub-graph matching algorithm (Zou et al., 2011). The extraction only considers the nodes in the graph. The similarity score between two nodes (x and y) is computed by

$$\mathcal{S}(x, y) = 1 - \mathcal{D}(x, y) / \max(|x|, |y|), \quad (1)$$

where $\mathcal{D}$ computes the Levenshtein distance, and $|*|$ is the length of $*$. After getting the sub-graph matched to the user question, we do the inference described in Section 3 to get the answer.

5 The OHO dataset

We gather the frequently asked questions (FAQs) summarized by the customer service department of an e-commerce company. We remove FAQs that aren’t attached to any documents and that are attached to documents with only images. Finally, we get a dataset of 2000 user manuals, each of which has an FAQ.

Type	Number	Question Words	Answer spans
Factoid-style	955 (47.75%)	11.86	1.17
Procedure-style	483 (24.35%)	12.07	1.14
Inconsistent	562 (28.10%)	11.36	1.16
All	2000	11.77	1.16

Table 4: The number (percentage) of questions, the average number of words in the question, and the average number of text spans in the answer

B1: What are the actions that should be performed by the user in the manual? (Action node)
B2: What are the entities in the user manual? (Entity node)
B3: What are arguments (modifier, time, location and manner) of the action? (Action-ARG)
B4: What are arguments (footnote, attribute and state) of the entity? (Entity-ARG)
B5: What are the arguments of the attribute/state? (ARG-ARG)
B6: Is Action-2 in the next step of Action-2? (NEXT)
B7: Does the entity act as the patient of the action? (PAT)
B8: Is Entity-2 a sub-entity of Entity-1? (SUB)
B9: Does the entity act as the patient of the state (PATA)

Table 5: Basic questions for representation annotation

Answer annotation Each annotator is shown an FAQ and its attached user manual, which is displayed sentence by sentence. A toy example for the annotation of Q3 is shown in Figure 4. The annotator is required to select which sentences in the user manual can be possible responses to the question and then select the type of the question. Each sentence is treated as a text span in the answer. The statistics of the annotations are described in Table 4. It reveals that the procedure-style and inconsistent questions take up more than half part (52.45%) of the real-world user questions about user manuals.

Representation annotation This annotation is done on 200 user manuals of OHO. As some user manuals describe more than one task, there are 346 graphs that need to be annotated. It is not practical for the annotator to draw a heterogeneous graph for each user manual. Following Dalvi et al. (2018), we reduce the task as the annotation of answers to nine basic questions derived from Table 2. The basic questions are defined in Table 5. Specifically, the annotations of B1 and B2 are completed first. This is because the annotations of B3-B4 and B6-B8 depend on the results of B1-B2. Similarly, the annotations of B5 and B9 are completed last as they depend on B4. The statistics are shown in Table 6. Although the basic questions are factoid-style questions, we can use them and gold answers to roughly estimate the maximum potential of models — if a model can correctly answer these basic questions,

	Average	Minimum	Maximum
Sentences / user manual	7.73	2	33
Words / sentence	40.91	10	205
Graphs / user manual	1.73	1	8
Actions / graph	3.47	2	14
Entities nodes / graph	2.95	1	6
Action-ARG / Action	1.13	0	3
Entity-ARG / Entity	1.40	0	3
ARG-ARG / graph	0.86	0	3
NEXT / graph	2.47	1	13
PAT / graph	0.80	0	2
SUB / graph	2.22	0	8
PATA / graph	0.63	0	2

Table 6: Statistics of the representation annotation

it can construct TARA for user manuals and thus have a chance to answer higher-level (procedure-style, inconsistent or more) questions via unified inferences.

Quality control We employ three CSRs with at least one-year working experience from the same department that summarizes the FAQs. We give a training session to the workers to help them fully understand our annotation requirements. The annotation tasks are released to the workers via an online annotation platform managed by the e-commerce company. On average, it takes about 3 minutes for an annotator to annotate the answer to an FAQ and about 2.8 hours to annotate all answers to basic questions of a user manual. An annotated result is accepted only if all three workers agree; otherwise, we invite two experts (one is the manager of the customer service department and the other is a post-doc working on computational linguistics) to make the final decision. The experts closely discuss with each other to ensure the consistency of results.

6 Experiments

To systematically benchmark the *Knowing-how &Knowing-that* task evaluation, we perform two experiments based on the OHO dataset. First, with the help of answer annotations, we compare the performances of HUM with seven baselines in answering real-world user questions, i.e., the FAQ-answering task. Second, with the help of representation annotations, we investigate the maximum potential of HUM, i.e., the B-answering task. Specifically, we use the LTP tool (Che et al., 2021) for word segmentation and semantic dependency parsing.

6.1 Experiment I: FAQ-answering Task

Baselines We compare HUM with seven baselines: 1) QA pattern matching (Peng et al., 2010; Jain and Dodiya, 2014), which utilizes a number of QA patterns written by experts; 2) lexical matching (Alfonseca et al., 2001; Yang et al., 2018), which computes the lexical similarity scores between the FAQ and candidate sentence via EQ (1); 3) semantic matching, which computes similarity scores of user questions and candidate sentences based on BERT embeddings (Devlin et al., 2018); 4) keyword matching (Moldovan et al., 2000), which measures the similarity between the keywords of the user question and candidate sentence; 5) pre-trained LM (i.e., PERT) (Cui et al., 2022), which has been trained with MRC tasks on other corpora; 6) self-supervised LM (Nie et al., 2022), which leverages the self-supervised strategy to train the LM with more than 360000 user manuals[7] with interrogative masks (Lewis et al., 2019); 7) fine-tuned LM, which fine-tunes the PERT models using a subset of OHO, the results are obtained from two-fold cross-validation. We also create a variant of HUM with oracle TARAs, named HUM-oracle. This method is tested on the 200 FAQs with user manuals that have gold TARAs.

Method	P / P@1	R / R@1	F1 / F1@1
QA pattern matching	0.77	0.18	0.30
Lexical matching	0.45 / 0.50	0.56 / 0.46	0.50 / 0.48
Semantic matching	0.36 / 0.38	0.60 / 0.36	0.45 / 0.37
Keyword matching	0.45 / 0.54	0.74 / 0.41	0.56 / 0.52
Pre-trained LM	0.24	0.24	0.24
Self-supervised LM	0.41	0.34	0.37
Fine-tuned LM	0.57 / 0.64	0.26 / 0.23	0.36 / 0.34
HUM	0.89 / 0.87	0.67 / 0.61	0.76 / 0.72
HUM-oracle	0.97 / 0.98	0.93 / 0.73	0.95 / 0.83

Table 7: Results of the FAQ-answering task. The values indicating the best performance and the worst performance are colored in green and red, respectively.

Evaluation metrics We use the precision (P), recall (R), and F1 scores to measure the performances. To make fair comparisons with methods designed for one-span answers (i.e., QA pattern matching, pre-trained LM, and self-supervised LM), we also report P@1, R@1, and F1@1 values.

Results Table 7 shows the performance of HUM and baselines on the FAQ-answering task. We can see that HUM-oracle significantly outperforms others on all metrics. This demonstrates that representing user manuals as TARAs is a desired resolution to answer various user questions. We also make the following observations. 1) The keyword matching method gets a higher recall score than HUM, but its

[7] They are online helping documents from the e-commerce company without FAQs and annotations.

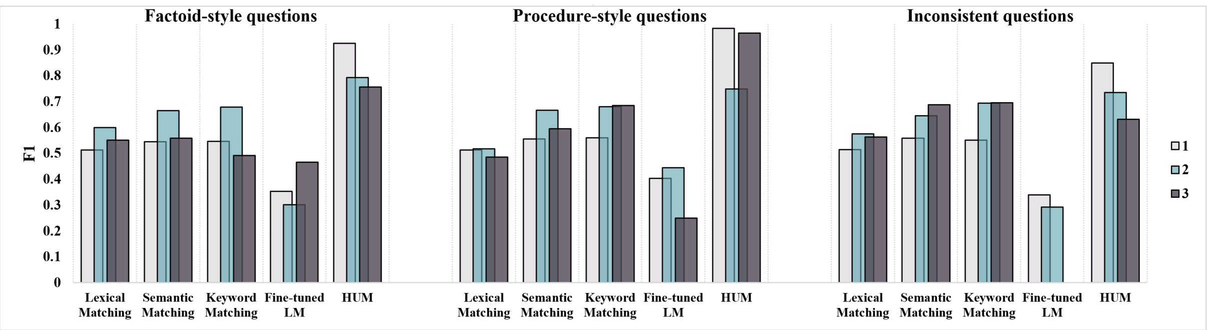


Figure 5: F1 scores w.r.t. different types of user questions and different number of answer spans

Basic Question	Biaffine	MFVI	HanLP	LTP	Human
B1	0.23 / 0.40 / 0.29	0.27 / 0.53 / 0.36	0.31 / 0.57 / 0.40	0.34 / 0.70 / 0.46	1.00 / 0.98 / 0.99
B2	0.10 / 0.08 / 0.09	0.20 / 0.16 / 0.18	0.17 / 0.15 / 0.16	0.24 / 0.25 / 0.25	0.90 / 0.98 / 0.94
B3	0.09 / 0.08 / 0.08	0.19 / 0.19 / 0.19	0.19 / 0.21 / 0.20	0.24 / 0.41 / 0.30	0.95 / 0.89 / 0.92
B4	0.04 / 0.02 / 0.03	0.08 / 0.09 / 0.09	0.06 / 0.06 / 0.06	0.12 / 0.16 / 0.14	0.88 / 0.80 / 0.84
B5	0.07 / 0.02 / 0.03	0.12 / 0.03 / 0.05	0.08 / 0.03 / 0.05	0.18 / 0.14 / 0.16	0.85 / 0.82 / 0.83
B6	0.05 / 0.08 / 0.06	0.13 / 0.21 / 0.16	0.14 / 0.22 / 0.17	0.18 / 0.35 / 0.24	0.98 / 0.98 / 0.98
B7	0.16 / 0.14 / 0.15	0.26 / 0.35 / 0.30	0.22 / 0.28 / 0.25	0.29 / 0.39 / 0.33	0.95 / 0.88 / 0.91
B8	0.04 / 0.06 / 0.05	0.09 / 0.11 / 0.10	0.06 / 0.08 / 0.07	0.08 / 0.13 / 0.10	0.95 / 0.88 / 0.91
B9	0.25 / 0.06 / 0.10	0.34 / 0.13 / 0.19	0.24 / 0.05 / 0.09	0.28 / 0.17 / 0.21	0.79 / 0.70 / 0.74
Average	0.11/ 0.10 / 0.10	0.19 / 0.20 / 0.18	0.16 / 0.18 / 0.16	0.22 / 0.30 / 0.24	0.92 / 0.88 / 0.90

Table 8: Performance of HUM using different semantic parsing tools and human w.r.t. the basic questions. For B1-B5, we report the P† / R† / F1† values. For B6-B9, we report the P / R / F1 values.

precision score is about half HUM. The matching of keywords between the user question and the user manual is similar to the extraction of the sub-graph from the heterogeneous graph of the user manual. So, this method can find all related spans to the user question. However, without the unified inference on TARA, it cannot reject the noise spans and thus get low precision scores. Similar results can also be found in the other matching methods. 2) The QA pattern matching has a high precision score because of sophisticated QA patterns (covering more than 80% of FAQs) but the worst recall score due to the high flexibility of answer expressions. 3) We are not surprised that the pre-trained LM and self-supervised LM rank last, as they are only exposed to factoid-style questions during the training phase thus they are insufficient to answer procedure-style and inconsistent questions. 4) The results of fine-tuned LM are slightly better than that of pre-trained LM and self-supervised LM but are still far behind HUM. We conjecture this is because of the lack of massive annotated data and the lack of unified inference. 5) After carefully checking HUM, we notice that about 5% of FAQs are not answered because HUM can't extract sub-graphs from user manuals. This suggests a possible future work is to eliminate such errors via more sub-graph matching algorithms, etc. 6) Except for HUM-oracle, the best F1 score is only 0.76, indicating the intrinsic challenge of the *Knowing-how* & *Knowing-that* task and there is substantial room for a model to represent user manuals as TARAs like an expert.

Auxiliary results We conduct a further study about methods that can produce multiple-span answers. As shown in Figure 5, in most cases, the performances decrease with the increase of answer spans. This suggests that it is more difficult to predict answers with more spans than answers with fewer spans. Compared with other methods, the fine-tuned LM rank last on all metrics and has poorer robustness when answering different types of user questions — the performance for inconsistent questions is significantly worse than that for procedure-style and factoid-style questions. It indicates that, despite the achievement of state-of-the-art results on many NLP tasks, the popular fine-tuned LM is not suitable for industry-scale MRC of user manuals without spending huge costs to annotate massive data. Notably, for all types of user questions, the F1 scores of HUM surpass other methods by a large margin on one-span and two-span answers and are slightly lower than the best value on three-span answers. This reveals the sufficiency and robustness of HUM to cope with the challenges raised by various types of user questions and multiple-span answers.

6.2 Experiment II: B-answering Task

We take a close look at HUM to investigate its ability to represent user manuals as TARAs and its maximum potential to answer higher-level questions. We create variants of HUM by replacing the backbone semantic parsing tool (i.e., LTP) with Biaffine (Dozat and Manning, 2018), MFVI (Wang et al., 2019), and HanLP (He and Choi, 2020). In addition, we employ five postgraduate students who major in computer science as annotators. They are asked to annotate the answers to basic questions. Before starting the annotation work, they are shown 10 samples annotated by experts without attending the training session. In this way, we obtain human (non-expert) upper bounds for this task.

Evaluation metrics Inspired by the evaluation of information extraction, we employ the precision, recall, and F1 scores based on BLEU scores (denoted as $P^\dagger$, $R^\dagger$, and $F1^\dagger$, respectively) (Tandon et al., 2020) to measure the performances of HUM on B1-B5. For B6-B9, we report the standard precision, recall, and F1 scores.

Results The results are reported in Table 8. The last column reports the human upper bounds. Although we have tried four state-of-the-art semantic parsing tools, the best-performing method only reaches ∼0.24 F1 scores, indicating the significant challenges to automatically constructing TARAs for user manuals. Specifically, the results of B6 are lower than that of B1, including the human results. This is because a worker's annotations to B6 are based on his annotations to B1, namely, the accumulative errors issue. The values of the human upper bound (∼0.92 precision score, ∼0.88 recall score and ∼0.90 F1 score) demonstrate that the task is feasible, well-defined and leaves plenty of room for more advanced semantic dependency parsing tools, more efficient heuristics, etc.

Issues and broader impact of TARA After investigating the bad cases, we find that, in addition to the accumulative errors, there are issues with co-reference problems, complex discourse parsing beyond sentences, etc. We leave the resolutions to these issues as future work. We observe that the performances of HUM on the B-answering task are much worse than that on the FAQ-answering task. Although the basic questions are factoid-style questions, they can compose all possible higher-level questions besides procedure-style and inconsistent ones. For example, Q2 can be decomposed into a combination of B1, B3, and B6. The FAQs in the B-answering task only take a small part of the questions composed from the basic questions. Thus, the B-answering task is much more challenging than the FAQ-answering task. This observation also strongly demonstrates that better performance on the B-answering task will largely improve HUM's ability to answer real-world questions. Besides, we'd like to talk about other potential benefits that can be gained from the TARAs. In addition to the unified inference of various questions, the joint representation of dynamic actions and entities also sheds light on the reasoning and planning of new tasks. Possible applications involve the arrangement of unordered actions, error detection of the draft user manuals, the automated composition of user questions, task-oriented information seeking, etc. The model, with the ability to correctly answer basic questions, can further be applied in downstream scenarios other than QA. For example, it is beneficial to build an intelligent training bot for new staff from customer service departments.

7 Conclusion and Future work

We propose the *Knowing-how* & *Knowing-that* task that requires the model to answer various questions about a user manual. We resolve it by representing the steps and facts of a user manual in a unified graph (TARA). To benchmark it, we design an efficient method and annotate a testing set derived from real-world customer service scenarios. Experiments reveal the superiority of TARA, the efficiency of HUM, and the significant challenges of OHO and the new task.

We take the primary step to study the MRC of user manuals in a more challenging setting, where various questions are involved. We hope our work can benefit further research on smart customer service. There are several directions for further work. First, to improve the performance of HUM, we will research resolutions to accumulative errors, co-reference problems, complex discourse parsing beyond sentences, etc. Second, to explore the potential of the proposed representation, we will introduce it to other tasks that need better interpretability, like task-oriented information seeking. Third, we will investigate the potential of unified inference for more complex user manuals, e.g., user manuals with multiple agents. Lastly, we plan to deploy HUM to online customer service to gain more insights for further improvements.

8 Limitations

We now explain the limitations and potential risks of our work. First, it seems the *Knowing-how & Knowing-that* task is a bit unfriendly to supervised learning methods as we only annotate the testing set. However, towards practical industry-scale applications, we encourage future work to utilize the current annotations and contribute to more efficient heuristic, unsupervised, self-supervised, or weakly supervised methods, etc. Second, each user manual in `OHO` only contains one user (agent). However, there are a number of user manuals involving more than one agent, e.g., "invite your friend as a new user and get cash back". This motivates us to explore multi-agent user manuals in our future work. Third, in addition to the textual content, many user manuals contain visual information like images and GIFs. Hence, it will be more desirable to add such user manuals and study the *Knowing-how & Knowing-that* task in multi-modal settings.

9 Ethics Statement

This paper presents a new task for machine comprehension of user manuals. Although the user manuals and FAQs involved in the task are collected from an e-commerce company, they are designed for normal users and have been widely used by the public for some time. We also have carefully checked these data to make sure they don't contain any personally identifiable information or sensitive personally identifiable information. Thus, we believe there are no privacy concerns.

All user manuals and FAQs are reviewed at least three times by the company's staff before being released to the public. Besides, we have been authorized by the company to make `OHO` publicly available for academic research. Thus, we believe the dataset doesn't contain any harmful information and is qualified for distribution.

The annotators of `OHO` consist of CSRs, a postdoc, and undergraduate students. As the dataset is about user manuals and the job is to answer questions about the user manuals, we believe there are no physical or mental risks to the annotators.

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China (No. 62206191 and No. 62272330); in part by the China Postdoctoral Science Foundation (No.2021TQ0222 and No. 2021M700094); in part by the Natural Science Foundation of Sichuan (No. 2023NSFSC0473), and in part by the Fundamental Research Funds for the Central Universities (No. 2023SCU12089 and No. YJ202219).

References

Enrique Alfonseca, Marco De Boni, José-Luis Jara-Valencia, and Suresh Manandhar. 2001. A prototype question answering system using syntactic and semantic information for answer retrieval. In *TREC*.

Aida Amini, Antoine Bosselut, Bhavana Dalvi Mishra, Yejin Choi, and Hannaneh Hajishirzi. 2020. Procedural reading comprehension with attribute-aware context flow. In *Automated Knowledge Base Construction*.

Pratyay Banerjee Tejas Gokhale Chitta Baral. Unsupervised question answering: Challenges, trends, and outlook.

Alethea L. Blackler, Rafael Gomez, Vesna Popovic, and M. Helen Thompson. 2016. Life is too short to rtfm: How users relate to documentation and excess features in consumer products. *Interacting with Computers*, 28(1):27–46.

Marco Bombieri, Marco Rospocher, Diego Dall'Alba, and Paolo Fiorini. 2021. Automatic detection of procedural knowledge in robotic-assisted surgical texts. *International Journal of Computer Assisted Radiology and Surgery*, 16(8):1287–1295.

Antoine Bosselut, Omer Levy, Ari Holtzman, Corin Ennis, Dieter Fox, and Yejin Choi. 2017. Simulating action dynamics with neural process networks. *arXiv preprint arXiv:1711.05313*.

Wanxiang Che, Yunlong Feng, Libo Qin, and Ting Liu. 2021. N-LTP: An open-source neural language technology platform for Chinese. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 42–49, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Cuong Xuan Chu, Niket Tandon, and Gerhard Weikum. 2017. Distilling task knowledge from how-to communities. In *Proceedings of the 26th International Conference on World Wide Web*, pages 805–814.

Peter Coates and Frank Breitinger. 2022. Identifying document similarity using a fast estimation of the levenshtein distance based on compression and signatures. *Digital Investigation*.

Yiming Cui, Ziqing Yang, and Ting Liu. 2022. Pert: Pre-training bert with permuted language model. *arXiv e-prints*, pages arXiv–2203.

Bhavana Dalvi, Lifu Huang, Niket Tandon, Wen-tau Yih, and Peter Clark. 2018. Tracking state changes in procedural text: a challenge dataset and models for process paragraph comprehension. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1595–1604.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina N Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding.

Timothy Dozat and Christopher D Manning. 2018. Simpler but more accurate semantic dependency parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 484–490.

Saransh Goyal, Pratyush Pandey, Garima Gaur, Srikanta Bedathur, Maya Ramanath, et al. 2021. Tracking entities in technical procedures–a new dataset and baselines. *arXiv preprint arXiv:2104.07378*.

Mark A Greenwood and Robert Gaizauskas. 2003. Using a named entity tagger to generalise surface matching text patterns for question answering. In *Proceedings of the Workshop on Natural Language Processing for Question Answering (EACL03)*, pages 29–34.

Han He and Jinho Choi. 2020. Establishing strong baselines for the new decade: Sequence tagging, syntactic and semantic parsing with bert. In *The Thirty-Third International Flairs Conference*.

Sonal Jain and Tripti Dodiya. 2014. Rule based architecture for medical question answering system. In *Proceedings of the Second International Conference on Soft Computing for Problem Solving (SocProS 2012), December 28-30, 2012*, pages 1225–1233. Springer.

Yiwei Jiang, Klim Zaporojets, Johannes Deleu, Thomas Demeester, and Chris Develder. 2020. Recipe instruction semantics corpus (risec): Resolving semantic structure and zero anaphora in recipes. In *AACL-IJCNLP 2020, the 1st Conference of the Asia-Pacific Chapter of the Association Computational Linguistics and 10th International Joint Conference on Natural Language Processing*, pages 821–826. Association for Computational Linguistics (ACL).

Chloé Kiddon, Ganesa Thandavam Ponnuraj, Luke Zettlemoyer, and Yejin Choi. 2015. Mise en Place: Unsupervised Interpretation of Instructional Recipes. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 982–992, Lisbon, Portugal. Association for Computational Linguistics.

Fusataka Kuniyoshi, Kohei Makino, Jun Ozawa, and Makoto Miwa. 2020. Annotating and extracting synthesis process of all-solid-state batteries from scientific literature. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 1941–1950.

Patrick Lewis, Ludovic Denoyer, and Sebastian Riedel. 2019. Unsupervised question answering by cloze translation. *arXiv preprint arXiv:1906.04980*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Hirokuni Maeta, Tetsuro Sasada, and Shinsuke Mori. 2015. A framework for procedural text understanding. In *Proceedings of the 14th International Conference on Parsing Technologies*, pages 50–60.

Dan Moldovan, Sanda Harabagiu, Marius Pasca, Rada Mihalcea, Roxana Girju, Richard Goodrum, and Vasile Rus. 2000. The structure and performance of an open-domain question answering system. In *Proceedings of the 38th annual meeting of the Association for Computational Linguistics*, pages 563–570.

Shinsuke Mori, Hirokuni Maeta, Yoko Yamakata, and Tetsuro Sasada. 2014. Flow graph corpus from recipe texts. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2370–2377.

Sheshera Mysore, Zachary Jensen, Edward Kim, Kevin Huang, Haw-Shiuan Chang, Emma Strubell, Jeffrey Flanigan, Andrew McCallum, and Elsa Olivetti. 2019. The materials science procedural text corpus: Annotating materials synthesis procedures with shallow semantic structures. In *Proceedings of the 13th Linguistic Annotation Workshop*, pages 56–64.

Nima Nabizadeh, Dorothea Kolossa, and Martin Heckmann. 2020. Myfixit: an annotated dataset, annotation tool, and baseline methods for information extraction from repair manuals. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 2120–2128.

Yuxiang Nie, He-Yan Huang, Zewen Chi, and Xian-Ling Mao. 2022. Unsupervised question answering via answer diversifying. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1732–1742.

Li Peng, Teng Wen-Da, Zheng Wei, and Zhang Kai-Hui. 2010. Formalized answer extraction technology based on pattern learning. In *International Forum on Strategic Technology 2010*, pages 236–240. IEEE.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Deepak Ravichandran and Eduard Hovy. 2002. Learning surface text patterns for a question answering system. In *Proceedings of the 40th Annual meeting of the association for Computational Linguistics*, pages 41–47.

Gilbert Ryle. 2009. *The concept of mind*. Routledge.

Martin M Soubbotin and Sergei M Soubbotin. 2001. Patterns of potential answer expressions as clues to the right answers. In *TREC*.

Niket Tandon, Keisuke Sakaguchi, Bhavana Dalvi, Dheeraj Rajagopal, Peter Clark, Michal Guerquin, Kyle Richardson, and Eduard Hovy. 2020. A dataset for tracking entities in open domain procedural text. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6408–6417, Online. Association for Computational Linguistics.

Alain C Vaucher, Federico Zipoli, Joppe Geluykens, Vishnu H Nair, Philippe Schwaller, and Teodoro Laino. 2020. Automated extraction of chemical synthesis actions from experimental procedures. *Nature communications*, 11(1):1–11.

Xinyu Wang, Jingxian Huang, and Kewei Tu. 2019. Second-order semantic dependency parsing with end-to-end neural networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4609–4618.

Yoko Yamakata, Shinsuke Mori, and John A Carroll. 2020. English recipe flow graph corpus. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5187–5194.

An Yang, Kai Liu, Jing Liu, Yajuan Lyu, and Sujian Li. 2018. Adaptations of ROUGE and BLEU to better evaluate machine reading comprehension task. pages 98–104.

Zhihan Zhang, Xiubo Geng, Tao Qin, Yunfang Wu, and Daxin Jiang. 2021. Knowledge-aware procedural text understanding with multi-stage training. In *Proceedings of the Web Conference 2021*, pages 3512–3523.

Ziqi Zhang, Philip Webster, Victoria Uren, Andrea Varga, and Fabio Ciravegna. 2012. Automatically extracting procedural knowledge from instructional texts using natural language processing. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 520–527.

Botao Zhong, Xuejiao Xing, Hanbin Luo, Qirui Zhou, Heng Li, Timothy Rose, and Weili Fang. 2020. Deep learning-based extraction of construction procedural constraints from construction regulations. *Advanced Engineering Informatics*, 43:101003.

Ming Zhu, Aman Ahuja, Da-Cheng Juan, Wei Wei, and Chandan K Reddy. 2020. Question answering with long multiple-span answers. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3840–3849.

Lei Zou, Jinghui Mo, Lei Chen, M Tamer Özsu, and Dongyan Zhao. 2011. gstore: answering sparql queries via subgraph matching. *Proceedings of the VLDB Endowment*, 4(8):482–493.

A Settings of the baselines for the FAQ-answering task

The detailed settings of the baselines models of the FAQ-answering task are as follows.

- QA pattern matching (Peng et al., 2010; Jain and Dodiya, 2014) uses handwritten rules to match questions and answers with corresponding syntactic formats (Ravichandran and Hovy, 2002; Soubbotin and Soubbotin, 2001; Greenwood and Gaizauskas, 2003). For example, given Q1 following the pattern of what be <entity>?, the answer is the sentence in the user manual following the pattern of <entity> be <value>.
- Lexical matching (Alfonseca et al., 2001; Yang et al., 2018). We use EQ (1) to calculate the lexical similarity between question and candidate sentence (Coates and Breitinger, 2022). Then we choose the top two sentences with the highest score as the final answer.
- Semantic matching (Devlin et al., 2018). We represent a sentence by averaging the token embeddings in this sentence. Then we compute the cosine similarity between question representation and candidate representation. The top two sentences with the highest score are used as the final answers.
- Keyword matching (Moldovan et al., 2000). We use the TF-IDF algorithm to extract ten keywords for each user manual. Meanwhile, we use the keywords extraction API of the iFLYTEK open platform to obtain the keywords of the question text. We calculate the matching score between each candidate answer and question according to the following formula:

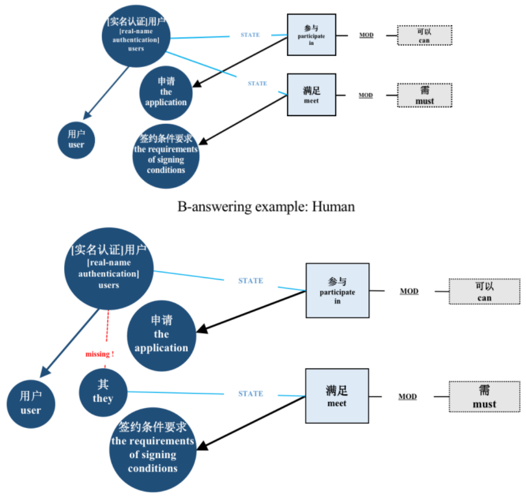
$$\begin{aligned}\mathcal{S} =& 16 \times |K_Q| + 16 \times |K_A| + 16 \times |K_Q \cap K_A| + \\ & 16 \times |\Gamma_{K_Q} \cap \Gamma_{K_A}| - 4 \times \sqrt{\mathcal{D}_{\max(K_{\{Q\}})}}\end{aligned} \tag{2}$$

  where $|K_Q|$ represents the number of candidate answer keywords existing in the question text, $|K_A|$ represents the number of question keywords existing in the candidate answers, $|K_Q \cap K_A|$ represents the number of words that are both question keywords and answer keywords, $\Gamma_{K_Q} \cap \Gamma_{K_A}$ represents how many question keywords are on the same sub_tree of the candidate answer parsing tree, where the parsing tree of the candidate answer is obtained from the semantic dependency parsing of LTP (Che et al., 2021), $\mathcal{D}_{\max(K_{\{Q\}})}$ represents the distance between the two question keywords that are farthest apart in the candidate answer text.
- Pre-trained LM. We use the Chinese machine reading comprehension model based on PERT-large (Cui et al., 2022), which has been fine-tuned on a mixture of Chinese MRC datasets. It is highly competitive in many tasks of machine reading comprehension and sequence labeling. It obtained 90.8, 95.7, and 79.3 F1-score on CMRC 2018 Dev, DRCD Dev, and SQUAD-Zen Dev (Answerable) data set respectively. We concatenate the question text and the user manual and then feed it to the model to predict the start and end positions of answer spans.
- Self-supervised LM (Nie et al., 2022). Self-supervised learning is the approach that trains the model using the constructed data sets of potential QA data mined from a large amount of corpus (Baral). It can obtain a quantity of data without manual annotation to fine-tune the pre-training model. Inspired by previous work (Lewis et al., 2019), we mask actions, entities, and corresponding arguments and relations in the source user manuals and replace them with interrogative words (when, where, why, how, etc.) to construct QA data. We finally constructed 800,000 QA pairs from 360,000 user manuals as the training dataset. For model training, we still choose the PERT-large model, whose pre-training paradigm is also self-supervised learning. Following the pre-training paradigm, we concatenate the question text and the original user manual as the input of the model and then predict the start and end positions of the answer span. We choose the AdamW (Loshchilov and Hutter, 2017) as the optimizer, the learning rate is set to 1e-5, and a total of three epochs are trained.
- Fine-tuned LM. The labelled dataset `OHO` is divided into two parts. We fine-tune the pre-training model on half of the data and then evaluate the performance of the model on the other half of the data. To ensure that the model can output multiple possible answer spans for each question, we splice the question text and candidate sentences in the source user manual into the model in turn and then predict whether each candidate sentence is the answer to the question, and we still choose PERT-large as the pre-training model. The GPU we used for model training is Tesla P100, the max length of the model input is set to 512, the batch size is set to 4, the learning rate is set to 1e-5, and a total of six epochs are trained. The results are obtained from two-fold cross-validation.

B Case Study of the B-answering task

We here discuss the issues of HUM in generating TARAs for user manuals. Figure 6 presents a bad case caused by the co-reference problem — the model fails to identify that "they" refers to "real-name authentication users", leading to a missing STATE argument of the "real-name authentication users" node. Figure 7 presents a bad case caused by accumulative errors — the model generates noise relations in TARA after it wrongly treats "pass" as an Action node.

S: 仅实名认证用户可以参与申请，其需满足签约条件要求。
Only real-name authentication users can participate in the application,
and they need to meet the requirements of signing conditions

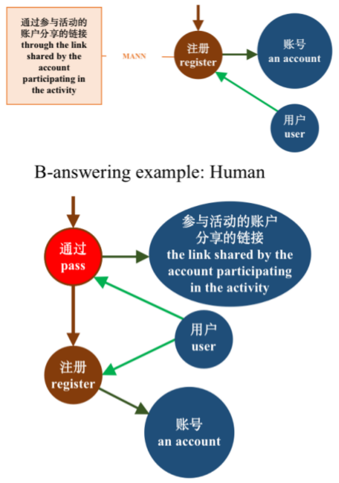


B-answering example: Human

B-answering example: HUM

Figure 6: Bad case caused by the co-reference issue

S: 通过参与活动的账户分享的链接注册账号。
Register an account through the link shared by
the account participating in
the activity.


通过参与活动的账户分享的链接 through the link shared by the account participating in the activity
MANN
注册 register
账号 an account
用户 user
通过 pass
参与活动的账户分享的链接 the link shared by the account participating in the activity

B-answering example: Human

B-answering example: HUM

Figure 7: Bad case caused by accumulative errors

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
8

☑ A2. Did you discuss any potential risks of your work?
8

☑ A3. Do the abstract and introduction summarize the paper's main claims?
1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☒ Did you use or create scientific artifacts?

Left blank.

□ B1. Did you cite the creators of artifacts you used?
No response.

□ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
No response.

□ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
No response.

□ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
No response.

□ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
No response.

□ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
No response.

C ☑ Did you run computational experiments?

6

□ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Not applicable. Left blank.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
6

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
6

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
6

D ☑ Did you use human annotators (e.g., crowdworkers) or research with human participants?
5

☑ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
5

☑ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
5

☑ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
5

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.