Exploring LLMs for Personal Knowledge Graph Population from Conversation

Anonymous ACL submission

Abstract

Although large language models (LLMs) have made significant advancements, they still lack the ability to personalize responses. However, manually inputting personal information into LLMs can be tedious and may never be completed. Since conversations contain a wealth of personal information, we propose to extract personal information and populate a personal knowledge graph (PKG) from conversation. We explored finetuning and prompting LLMs, but found that they still struggle with generating desired PKGs. Our analysis shows that GPT-3.5 cannot generate knowledge triples with desired relations and T5 often fails to identify the correct subject. Furthermore, GPT-3.5 struggles with extracting in-context subjects, recognizing negation expressions, and differentiating between questions and statements. By highlighting these limitations, we aim to inspire future research on PKG population from conversation and the development of personalized dialogue systems.

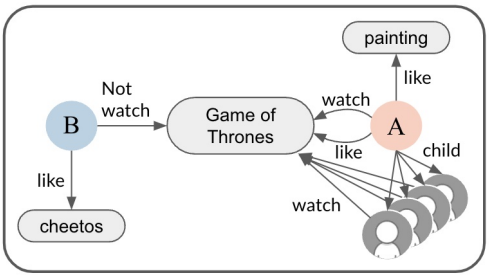
1 Introduction

A personal knowledge graph (PKG) is a structured knowledge source that stores personalized information. In contrast to a general-purpose knowledge graph, the entities in a PKG may only be relevant to the user and not globally important (Balog and Kenter, 2019). For example, "Rose is pregnant" might be a significant event only for Rose's friends but not the general public. Thus, the entity "Rose" would not exist in a general-purposed knowledge graph (unless Rose is a celebrity), nor would her personal information, such as her pregnancy status.

A PKG can function as an external personal memory for tasks such as a personal memory assistant. It can also be integrated with other systems, such as chatbots or recommendation systems, to produce personalized results. Some might argue that personal information can be added from dialogue histories, as many of current conversational systems do (Roller et al., 2021). However, there are limitations to the number of tokens that can be included in the inputs of large language models (LLMs), and studies have shown that LLMs still struggle with remembering memories from a long time ago or maintaining their knowledge across multiple turns in the interaction. (Xu et al., 2022; Bang et al., 2023) By storing information in the form of a PKG, it can increase the interpretability of how machines make decisions and allow users to manage their information independently.

Documenting personal knowledge can be a tedious and overwhelming task. Requesting users to input all range of their personal information covering their interests, preferences, etc., can be nearly impossible. However, conversations, as a means of human communication, provide a rich source of personal information. Therefore, we propose to automatically construct a PKG from conversations instead of relying on users to manually supply their personal information.

As shown in Fig.1, our goal is to generate the



A: My children and I were just about to watch Game of Thrones.
B: Nice! How old are your children?
A: I have four that range in age from 10 to 21. You?
B: I do not have children at the moment.
A: That just means you get to keep all the popcorn for yourself.
B: And Cheetos at the moment!
A: Good choice. Do you watch Game of Thrones?
B: No, I do not have much time for TV.
A: I usually spend my time painting: but, I love the show.

Figure 1: An example of personal knowledge graph population from conversation.

PKG of the two conversational partners given a snippet of their dialogue. Prior studies on personal knowledge extraction from conversation have mainly focused on the **utterance level of a single speaker**, which entails extracting attributes of the speaker (Li et al., 2014; Tigunova et al., 2021, 2020; Wang et al., 2022). In contrast, our work operates at a more comprehensive **conversational level of both speakers**. In conversations, relations between entities may only become valid after confirmation by the other speaker. For instance, "A: Do you like Japanese food? B: I love it!", the information that B likes Japanese food cannot be extracted without considering the entire conversation.

The process of populating a knowledge graph typically involves several subtasks, including entity detection or named entity recognition (NER), relation extraction, and entity linking. In this work, we tackle the task of populating a personal knowledge graph from conversation by fine-tuning generative T5 models (Raffel et al., 2020) and prompting GPT-3.5 (Radford et al., 2018). We extract personal knowledge from the Life Event Dialog dataset (Chen et al., 2023) in the form of (Subject, Relation, Object) triples. Our results suggest that GPT-3.5 model enumerates all subtle activities but not outputs the desired relation types, while T5 model struggles to predict the right subject even it correctly captures the relation and object.

Our contributions can be summarized as follows: (1) We extend the task of personal knowledge graph population from conversation beyond the utterance level to the conversational level, identifying the personal knowledge not only for a single speaker. (2) We explore finetuning and prompting LLMs for the task of personal knowledge graph population from conversation and provide a comprehensive analysis of how current LLMs understand conversations.

2 PKG Population from Conversation

2.1 Challenges

In this section, we elaborate the challenges and the main differences of populating a PKG from conversations versus conventional information extraction tasks. Identifying spans and their types is crucial in conventional information extraction task. Both relation extraction and event extraction tasks predicts the types of a span in the given text of a single-person narrative. For instance, given a sentence "Steve became CEO of Apple in 1997," relation extraction task focuses on classifying a predefined relation type ("work for") from two mentions ("Steve" and "Apple"). Event extraction task identifies spans of trigger ("became") and arguments ("Steve" and "Apple") from the given text, and classifies these spans into predefined types ("start position", "employee", and "employer"). The mention-entity relationship is usually one-to-one or many-to-one mapping. In a conversation, there are only two entities that we are focusing on, i.e., S1 and S2. However, the surface form of mentions from different entities are highly overlapped (Chen et al., 2023). For example, the same surface form "I" might refer to the Speaker 1 (S1) and the Speaker 2 (S2) in different utterances. The mention itself is not important; instead, we are solely concerned with the information pertaining to the entity (S1 or S2). Nevertheless, current IE models predict relations given mentions and their locations or predict mentions and their types based on the context, and when the surface form is the same, the model get confused about what to predict.

2.2 Approach

Given a conversation, our goal is to output the personal knowledge in the form of (Subject, Relation, Object) triple. Following the PKG definition from (Balog and Kenter, 2019), we limit the Subject to either one of the speakers in the dialogue.

We examine finetuning T5 models with different templates, and prompting GPT-3.5. The templates for T5 are shown in Appendix C.2. We utilize the GPT-3.5-turbo model from OpenAI's API to extract personal knowledge in the form of triples by the prompt shown in Appendix C.1. To guide the output, we supplied GPT-3.5 with a list of relations and two examples, which are tailored to generate triples based on the given conversation.

3 Experiment

3.1 Setting

Our experiment is conducted on the Life Event Dialog dataset (Chen et al., 2023), a collection of speakers' daily life events annotated on DailyDialogue (Li et al., 2017). More details about the dataset is described in Appendix B.

Most end-to-end information extraction works adopt the *strict* evaluation (Nayak and Ng, 2020), in which a triple is considered correct only if all its elements are correct (Ye et al., 2022).

However, in the case of personal knowledge from conversations, the mentions of objects often

Strict	P	R	F1	BERTScore (BS)
T5	**0.265**	**0.221**	**0.241**	**0.944**
GPT	0.069	0.151	0.094	0.897

Sbj-first	Sbj P	Sbj R	Sbj F1	Rel P	Rel R	Rel F1	Obj BS
T5	**0.644**	0.827	**0.723**	**0.558**	0.428	**0.483**	**0.964**
GPT	0.575	**0.974**	**0.723**	0.317	**0.540**	0.399	0.933

Sbj-Rel	P	R	F1	Obj BS
T5	**0.362**	0.428	**0.391**	**0.919**
GPT	0.211	**0.540**	0.303	0.882

Rel-first	Sbj P	Sbj R	Sbj F1	Rel P	Rel R	Rel F1	Obj BS
T5	0.608	0.388	0.474	**0.644**	0.567	**0.603**	**0.967**
GPT	**0.840**	**0.540**	**0.658**	0.253	**0.633**	0.362	0.933

Table 1: Result of automatic evaluation metrics described in Section 3.1

	Output Triples	Valid Ratio
T5	206	54.4%
GPT	511	**83.8%**

Table 2: GPT generates more triples than T5 for all dialogues in test set, and most of them are valid.

consist of multi-word descriptions instead of a single word. Consequently, we propose three additional evaluation modes tailored to the task of PKG population from conversation.

- In *Sbj-Rel*, we relaxed the strict evaluation by only evaluating F1 of (Subject, Relation), and evaluated Object by BERTScore (BS) (Zhang et al., 2020).
- In *Sbj-first*, we turned triples into a hierarchy tree {Subject: {Relation: Object}}. We first calculated the F1 of Subject. For example, if the ground truth only contains information of S2, but the model predicts triples for both S1 and S2, then the precision, recall, and F1 of Subject is 0.5, 1, and 0.67, respectively. Then, for the correctly predicted Subject, we calculated the Relation F1; for the correctly predicted relation, we calculated the Object BERTScore.
- *Rel-first* is similar to *Sbj-first*, except that we first evaluated Relation, then Subject, and then Object. The hierarchy tree is like {Relation: {Subject: Object}}.

In addition to the four automatic metrics (*Strict*, *Sbj-Rel*, *Sbj-first*, *Rel-first*), we conducted a human evaluation to check the *Valid Ratio* for both T5 and GPT-3.5 outputs. The *Valid Ratio* measures the accuracy of a triple in relation to the given dialogue. The authors manually examine each triple to determine whether the model hallucinated non-existent triples or if the triples were correct solely based on the provided dialogue. This evaluation was performed without comparing the generated triples to the ground truth.

3.2 Result

The results of automatic evaluations comparing the GPT-3.5 (GPT) [1] and T5 models are shown in Table 1. The finetuned T5 consistently outperforms GPT in most automatic metrics by a substantial margin. Despite being provided with a relation list in the prompt, GPT still generates many relations not included in the given list, leading to poor precision in *Strict* and *Sbj-Rel* metrics and low relation precision (Rel P) in *Sbj-first* and *Rel-first* metrics.

Table 2 indicates that most GPT-generated triples are valid, achieving a higher *Valid Ratio* than the T5 model. Although 84% of GPT-generated triples are accurate based on the dialogue, most of them do not appear in the ground truth triples, causing the discrepancy between Table 1 and Table 2. In comparison to the ground truth, which has a total of 237 triples, the GPT model generates more than twice as many triples. These additional triples predominantly enumerate trivial information from the conversation rather than extracting significant personal knowledge. On the other hand, we observe that T5 often struggles to identify the correct subject for a triple, although it can output the predefined relations after fine-tuning. The choice between the T5 and GPT models may involve a trade-off between controllability and coverage (variety).

3.3 GPT Error Analysis

We analyzed the 16% of invalid triples generated by the GPT model and summarized our observations with corresponding examples in Table 3.

GPT Error-1: Fail to predict in-context subjects. The majority of GPT errors result from incorrect subject prediction, especially when the subject is mentioned in the context. GPT often outputs triples with subjects that are neither S1 nor S2, even though our ground truth personal knowledge focuses exclusively on the two speakers. In these

[1] All "GPT" in this paper refers to the GPT-3.5 model.

Error Type		Example Dialogue	Triple
In-Context Subject	28.2%	S2: "They're on special offer today."	GPT: (S2, have special offer, today) Correct: they
		S2: "... bring my cat, Mr. Twinkles."	GPT: (S2, name, Mr Twinkles) Correct: (S2's) cat
Speaker Subject	27.1%	S1: "Can you recommend some?" S2: "I think Pond's is the best."	GPT: (S1, recommend, Pond's) Correct: S2
Negation	18.8%	S1: "So have you accepted offers from other companies?" S2: "No, I haven't got one by now."	GPT: (S2, accept, offers from other companies)
		S1: "Did you go to the concert last weekend?" S2: "No, I didn't. And you?"	GPT: (S2, go, concert last weekend)
		S1: "I can't find the book you lent me."	GPT: (S1, find, book)
		S1: "I've never tasted anything better."	GPT: (S1, taste, anything better)
Questions as Statements	14.1%	S1: "So have you accepted offers from other companies ?"	GPT: (S2, accept, offers from other companies)
Others	11.8%		

Table 3: GPT error analysis and examples.

instances, GPT tends to incorrectly identify the subject as either S1 or S2, when it actually refers to a third party mentioned in the context.

GPT Error-2: Mess up the speaker and subject. Another common error occurs when GPT mess up the speaker and the subject. There is no apparent reason for these inaccuracies, but we did observe that about half of these examples contain questions in the dialogue. As we discuss later, GPT seems to struggle with interpreting questions.

GPT Error-3: Hard to capture negation. Of the 15 dialogues containing negation, GPT incorrectly predicts 8 of them as positive. These dialogues sometimes include a question followed by a negative answer, or simply feature the speaker expressing negation. However, GPT tends to interpret them as positive triples.

GPT Error-4: Treat questions as true statements. As shown in the examples from Table 4, GPT occasionally predicts a triple given merely a question and before the other speaker has even provided an answer.

GPT doesn't hallucinate many personal knowledge from conversation. We investigated whether GPT hallucinates personal knowledge from conversations, given that hallucination is one of the most notorious challenges faced by current LLMs. From our observations, most of the imagined personal knowledge inferred from dialogues is either correct or remains unverified. Moreover, one of our goals in populating a PKG is to enable users to manipulate or correct any potentially false inferred knowledge by themselves, regardless of whether LLMs generate hallucinated personal knowledge.

Our investigation highlights potential weaknesses in GPT and analyzes the causes of errors. We found GPT still does not fully understand the conversational context, and further inspections on how LLMs process negative expressions and questions is needed.

4 Conclusion

We introduce the task of personal knowledge graph population from conversations and highlight the challenges of directly applying conventional information extraction approaches to this task. We explore fine-tuning T5 and prompting GPT-3.5 models for this purpose. While the fine-tuned T5 consistently outperforms GPT in automatic evaluation metrics, GPT frequently generates more valid triples based on the dialogue. The outputs from the two models suggest that T5 often fails to identify the correct subject, while the GPT model produces numerous trivial relations not present in the ground truth personal knowledge. Our error analysis further reveals that GPT struggles with subject prediction, interpreting questions, and handling negation in this task. We hope this work can facilitate the automatic construction of PKG from conversations, assist users in managing their own personal data when interacting with LLMs, and contribute to the development of personalized dialogue systems.

Limitations

Some limitations of this work should be acknowledged. First, our investigation focuses on two specific LLMs, GPT-3.5 and T5, which may not fully represent the broader landscape of large language models. Second, the extraction process may inadvertently introduce biases or inaccuracies into the personal knowledge graphs. Lastly, our dataset is limited to English conversations about daily life and our approach may not be generalized to all types of dialogues or personal knowledge extraction scenarios, as the quality of the extracted information may vary depending on the content and context of the conversations.

Besides, there are several potential risks associated with this work, which should be carefully considered. The primary concern for many people may be the privacy issues arising from extracting personal knowledge from conversations. While we circumvent this problem by using publicly available data in our study, privacy concerns could emerge when adopting this approach to real-world applications. Users may not be fully informed, provide consent, or feel comfortable with having their personal information stored in such a manner, particularly if it is accessible to third parties. Also, the extracted personal information could potentially be misused by unauthorized individuals or entities, which might lead to identity theft, targeted advertising, or other malicious activities.

References

Krisztian Balog and Tom Kenter. 2019. Personal knowledge graphs: A research agenda. In *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval*, pages 217–220.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, et al. 2023. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *arXiv preprint arXiv:2302.04023*.

Yi-Pei Chen, An-Zi Yen, Hen-Hsen Huang, Hideki Nakayama, and Hsin-Hsi Chen. 2023. Led: A dataset for life event extraction from dialogs. In *The 17th Conference of the European Chapter of the Association for Computational Linguistics*.

Xiang Li, Gokhan Tur, Dilek Hakkani-Tür, and Qi Li. 2014. Personal knowledge graph population from user utterances in conversational understanding. In *2014 IEEE Spoken Language Technology Workshop (SLT)*, pages 224–229. IEEE.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. Dailydialog: A manually labelled multi-turn dialogue dataset. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 986–995.

Tapas Nayak and Hwee Tou Ng. 2020. Effective modeling of encoder-decoder architecture for joint entity and relation extraction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8528–8535.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Eric Michael Smith, Y-Lan Boureau, and Jason Weston. 2021. Recipes for building an open-domain chatbot. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 300–325, Online. Association for Computational Linguistics.

Anna Tigunova. 2020. Extracting personal information from conversations. In *Companion Proceedings of the Web Conference 2020*, pages 284–288.

Anna Tigunova, Paramita Mirza, Andrew Yates, and Gerhard Weikum. 2021. PRIDE: Predicting Relationships in Conversations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4636–4650, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Anna Tigunova, Andrew Yates, Paramita Mirza, and Gerhard Weikum. 2020. Charm: Inferring personal attributes from conversations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5391–5404.

Zhilin Wang, Xuhui Zhou, Rik Koncel-Kedziorski, Alex Marin, and Fei Xia. 2022. Extracting and inferring personal attributes from dialogue. In *Proceedings of the 4th Workshop on NLP for Conversational AI*, pages 58–69.

Jing Xu, Arthur Szlam, and Jason Weston. 2022. Beyond goldfish memory: Long-term open-domain conversation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5180–5197, Dublin, Ireland. Association for Computational Linguistics.

Hongbin Ye, Ningyu Zhang, Hui Chen, and Huajun Chen. 2022. Generative knowledge graph construction: A review. *arXiv preprint arXiv:2210.12714*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

A Works on Personal Information Extraction

Earlier work like Li et al. (2014) constructed personal knowledge graph by a pipeline approach. Given private query logs from Microsoft Cortana, they first determined whether the input utterance contains personal information, then classified the utterance into one predefined relation types, followed by slot filling the attributes of relations. Tigunova (2020); Tigunova et al. (2020) identify attribute-related keywords and rank relevant documents to predict a person's hobby and profession from Reddit.

These previous works focused only on one speaker and on utterance-level instead of dialogue-level. That is, they all only detect personal knowledge from the speaker's own utterance and only the speaker's relations. Therefore, they didn't encounter the challenge of subject detection or entity linking. Not to mention that their data were not real conversational data but natural sentences crawled from social media or single-person utterance, except the private data in (Li et al., 2014). Besides, they only detect a few relation types, ranging from 2 to 39.

Our work, in contrast, is conducted on real conversation of two speakers, capturing their real-time interaction to build the PKG and extracting up to 103 relationships of personal knowledge. Step further to SVM, LSTM, and other neural networks, we showed the effects of prompting or finetuning of LLMs and provide an in-depth analysis on the results.

	Train	Valid	Test
# Sample	1,631	141	110
# Triple	3,473	362	237

Table 4: Data statistics.

B Details about Life Event Dialog Dataset

The Life Event Dialog (LED) dataset is built on DailyDialogue, both datasets are licensed under CC BY-NC-SA 4.0. LED covers five topics (Relationship, Ordinary Life, Work, Tourism, and Attitde & Emotion) on English conversations. LED annotates personal life events, each consisting of a subject, an object, three granularities of event types, and event statuses (polarity, modality, and time). We considered only events with positive and actual statuses, in which the subject is one of the two speakers (S1 or S2). In our experiment, we utilized the *Class* event type as the relation and limited the triples to those with relations appearing more than 5 times in the training set, resulting in 103 relation types. Additionally, we converted the mentions of S1 and S2 to either "S1" or "S2" in the triples. The data statistics are presented in Table 4.

This research aligns with the intended usage of LED. We build upon the task of conversational life event extraction proposed in LED, focusing on the events involving both speakers and further populating a PKG. We authors have manually checked for offensive content and identifiers by sampling 10% of dialogues in the dataset.

C Details about Experiments

Given a dialog, extract a personal knowledge graph in the form of triples: (SUBJECT, RELATION, OBJECT), where the RELATION is from the following list: {relation_list}.

Example:
dialog: "S1 : May I help you ?\nS2 : Yes . I have to stay in your city for just one day , can you suggest a short tour ?\nS1 : Are you interested in the natural landscape or the human landscape ?"
triples: [["S1", "suggest", "S2"], ["S1", "suggest", "a short tour"],["S1", "help", "S2"],["S2", "stay", "in your city"]]

Example:
dialog: "S1 : Do you really have to work today ?\nS2 : Yes . I 'm afraid so .\nS1 : But you 'll miss out on the football game .\nS2 : Oh . Well , it ca n't be helped ."
triples:[["S2", "work", "NO_OBJ"],["S2", "miss", "football game"]]

dialog: {input}
triples:

Table 5: The prompt design for GPT-3.5.

C.1 GPT

Table 5 is the prompt design for GPT-3.5. We replaced the injectable slots relation_list and input

with the relation list and the input dialogue respectively, and provided two demonstrative examples to guide the generation of desired triples output. We set the temperature to 0.1 to ensure the deterministic generation and kept other parameters the same as the default setting.

C.2 T5 Templates

We introduced special tokens <SOE>, <EOE>, <SBJ>, <VERB>, <OBJ>, and tried different combinations with and without these special tokens in the five templates in Table 6.

We conducted experiments with each template 10 times and averaged the results, as presented in Table 7. The best template scores for each evaluation metric are reported in Table 1.

D Human Evaluation on Valid Ratio

The author assessed the *Valid Ratio* on 110 samples from the test set, determining whether the output triple is a fact in the dialogue. As this judgement is **not subjective**, we did not recruit annotators for this evaluation.

ID	Templates
1	["{sbject}", "{relation}", "{object}"]
2	<S><SBJ>{sbject}<REL>{relation}<OBJ>{object}<E>
3	<S><REL>{relation}<SBJ>{sbject}<OBJ>{object}<E>
4	<S><SBJ>{sbject}</SBJ><REL>{relation}</REL><OBJ>{object}</OBJ><E>
5	<S><REL>{relation}</REL><SBJ>{sbject}</SBJ><OBJ>{object}</OBJ><E>

Table 6: Templates used for finetuning T5.

Strict	**P**	**R**	**F1**	**BS**
1	0.199 (± 0.02)	0.178 (± 0.01)	0.188 (± 0.01)	0.934 (± 0.00)
2	0.265 (± 0.03)	0.221 (± 0.03)	**0.241** (± 0.03)	**0.944** (± 0.01)
3	0.233 (± 0.03)	0.233 (± 0.04)	0.231 (± 0.03)	0.937 (± 0.00)
4	0.246 (± 0.03)	0.221 (± 0.01)	0.233 (± 0.02)	0.938 (± 0.01)
5	0.268 (± 0.04)	0.214 (± 0.03)	0.237 (± 0.03)	0.943 (± 0.01)

Sbj-Rel	**P**	**R**	**F1**	**Obj-BS**
1	0.329 (± 0.03)	0.355 (± 0.03)	0.341 (± 0.02)	0.918 (± 0.00)
2	0.385 (± 0.03)	0.388 (± 0.03)	0.386 (± 0.03)	0.924 (± 0.01)
3	0.362 (± 0.04)	0.428 (± 0.04)	**0.391** (± 0.03)	0.919 (± 0.01)
4	0.367 (± 0.03)	0.403 (± 0.03)	0.384 (± 0.02)	0.922 (± 0.01)
5	0.375 (± 0.05)	0.368 (± 0.06)	0.370 (± 0.05)	**0.926** (± 0.01)

Sbj-first	**Sbj-P**	**Sbj-R**	**Sbj-F1**	**Rel-P**	**Rel-R**	**Rel-F1**	**Obj-BS**
1	0.661 (± 0.02)	0.807 (± 0.04)	**0.726** (± 0.02)	0.500 (± 0.04)	0.355 (± 0.03)	0.415 (± 0.03)	0.961 (± 0.01)
2	0.660 (± 0.02)	0.768 (± 0.05)	0.709 (± 0.02)	0.581 (± 0.05)	0.388 (± 0.03)	0.464 (± 0.03)	0.967 (± 0.01)
3	0.644 (± 0.02)	0.827 (± 0.06)	0.723 (± 0.03)	0.558 (± 0.05)	0.428 (± 0.04)	**0.483** (± 0.03)	0.964 (± 0.01)
4	0.652 (± 0.03)	0.802 (± 0.03)	0.719 (± 0.02)	0.560 (± 0.04)	0.403 (± 0.03)	0.468 (± 0.03)	0.966 (± 0.01)
5	0.653 (± 0.02)	0.738 (± 0.06)	0.692 (± 0.02)	0.567 (± 0.06)	0.368 (± 0.06)	0.445 (± 0.05)	**0.974** (± 0.01)

Rel-first	**Sbj-P**	**Sbj-R**	**Sbj-F1**	**Rel-P**	**Rel-R**	**Rel-F1**	**Obj-BS**
1	0.580 (± 0.02)	0.355 (± 0.03)	0.440 (± 0.03)	0.577 (± 0.04)	0.534 (± 0.04)	0.554 (± 0.03)	0.961 (± 0.01)
2	0.608 (± 0.04)	0.388 (± 0.03)	0.474 (± 0.03)	0.644 (± 0.05)	0.567 (± 0.02)	**0.603** (± 0.03)	0.967 (± 0.01)
3	0.607 (± 0.03)	0.428 (± 0.04)	**0.501** (± 0.03)	0.605 (± 0.05)	0.593 (± 0.03)	0.598 (± 0.03)	0.964 (± 0.01)
4	0.605 (± 0.03)	0.403 (± 0.03)	0.483 (± 0.03)	0.600 (± 0.04)	0.581 (± 0.03)	0.590 (± 0.03)	0.966 (± 0.01)
5	0.595 (± 0.05)	0.368 (± 0.06)	0.454 (± 0.06)	0.637 (± 0.07)	0.558 (± 0.03)	0.593 (± 0.04)	**0.974** (± 0.01)

Table 7: Result of finetuning T5 using different templates.