PHALM: Building a Knowledge Graph from Scratch by Prompting Humans and a Language Model

Anonymous ACL submission

Abstract

Despite the remarkable progress in natural language understanding with pretrained Transformers, neural language models often do not have commonsense knowledge. Toward commonsense-aware models, there have been attempts to obtain knowledge, ranging from automatic acquisition to crowdsourcing. However, it is difficult to obtain a high-quality knowledge base at a low cost, especially from scratch. In this paper, we propose PHALM, a method of building a knowledge graph from scratch, by prompting both crowdworkers and a large language model. We used this method to build a Japanese event knowledge graph and trained Japanese neural commonsense models. Experimental results revealed the acceptability of the built graph and inferences generated by the trained models. We also report the difference in prompting humans and a language model.

1 Introduction

Since pretrained models (Radford and Narasimhan, 2018; Devlin et al., 2019; Yang et al., 2019) based on Transformer (Vaswani et al., 2017) appeared, natural language understanding has made remarkable progress. In some benchmarks, the performance of natural language understanding models has already exceeded that of humans. These models are applied to various downstream tasks ranging from translation and question answering to narrative understanding and dialogue response generation. In recent years, the number of parameters in such models has continued to increase (Radford et al., 2019; Brown et al., 2020), and so has their performance.

When we understand or reason, we usually rely on commonsense knowledge. Computers also need such knowledge to answer open-domain questions and to understand narratives and dialogues, for example. However, pretrained models often do not have commonsense knowledge.

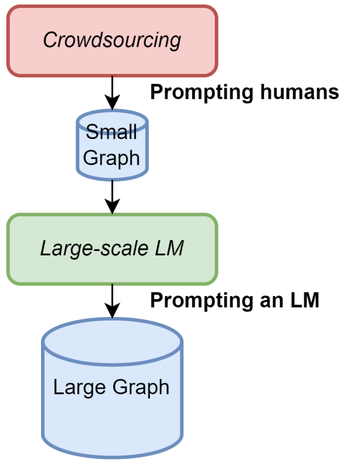


Figure 1: An overview of our method. We build a knowledge graph step by step from scratch, by prompting both humans and a language model.

There are many knowledge bases for commonsense inference. Some are built by crowdsourcing (Speer et al., 2017; Sap et al., 2019; Hwang et al., 2021), but acquiring a large-scale knowledge base is high-cost. Others are built by automatic acquisition (Zhang et al., 2019, 2020), but it is difficult to acquire high-quality commonsense knowledge. Recently, there have been some methods using large language models (LLMs) for building knowledge bases (Yuan et al., 2021; West et al., 2022; Liu et al., 2022). They often extend existing datasets, but do not build new datasets from scratch.

In this paper, we propose PHALM[1], a method to build a knowledge graph from scratch with both crowdsourcing and an LLM. Asking humans to describe knowledge using crowdsourcing and generating knowledge using a language model are essentially the same (as it were, the latter is an analogy of the former), and both can be considered to be *prompting*. Therefore, we consider prompting for

[1]PHALM stands for **P**rompting **H**umans **A**nd a **L**anguage **M**odel.

both humans and a language model and gradually acquire a knowledge graph from a small scale to a large scale. Specifically, we acquire a small-scale knowledge graph by asking crowdworkers to describe knowledge and use them as a few shots for an LLM to generate a large-scale knowledge graph. At each phase, we guarantee the quality of graphs by applying appropriate filtering.

We built a Japanese knowledge graph on events, considering prompts for both humans and a language model. With Yahoo! Crowdsourcing[2] and HyperCLOVA JP, a Japanese variant of the LLMs built by Kim et al. (2021), we obtained a knowledge graph that is not a simple translation, but unique to the culture. Then, we compared inferences collected by crowdsourcing and generated by the LLM. In addition to acquisition, we trained a Japanese neural commonsense model based on the built knowledge graph. With the model, we verified the acceptability of output inferences for unseen events. The resulting knowledge graph and the commonsense model created in this paper will be released to the public.[3]

2 Related Work

2.1 Commonsense Knowledge Datasets

There are several knowledge bases about commonsense, from what appears in the text to what is tacit but not written in the text. ConceptNet (Speer et al., 2017), for example, is a knowledge graph that connects words and phrases by relations. GenericsKB (Bhakthavatsalam et al., 2020) is a corpus describing knowledge of entities in natural language rather than in graph.

In some datasets, commonsense knowledge is collected in the form of question answering. Roemmele et al. (2011) acquire plausible causes and effects for premises as two-choice questions. Zellers et al. (2018) provide SWAG, acquiring inferences about a situation from video captions as four-choice questions. KUCI (Omura et al., 2020) is a dataset for commonsense inference in Japanese, which is obtained by combining automatic extraction and crowdsourcing. Talmor et al. (2019) build CommonsenseQA, which treats commonsense on ConceptNet's entities as question answering.

[2] `https://crowdsourcing.yahoo.co.jp/`

[3] if accepted

2.2 Knowledge Graphs on Events

Regarding commonsense knowledge bases, there are several graphs that focus on events. ATOMIC (Sap et al., 2019) describes the relationship between events, mental states (Rashkin et al., 2018), and personas. Hwang et al. (2021) merge ATOMIC and ConceptNet, proposing ATOMIC-2020.

There are also studies for leveraging context. GLUCOSE (Mostafazadeh et al., 2020) is a commonsense inference knowledge graph for short stories, built by annotating ROCStories (Mostafazadeh et al., 2016). CIDER (Ghosal et al., 2021) and CICERO (Ghosal et al., 2022) are the graphs for dialogues, where DailyDialog (Li et al., 2017) and other dialogue corpora are annotated with inferences.

ASER (Zhang et al., 2019) is an event knowledge graph, automatically extracted from text corpora by focusing on discourse. With ASER, TransOMCS (Zhang et al., 2020) aims at bootstrapped knowledge graph acquisition by pattern matching and ranking.

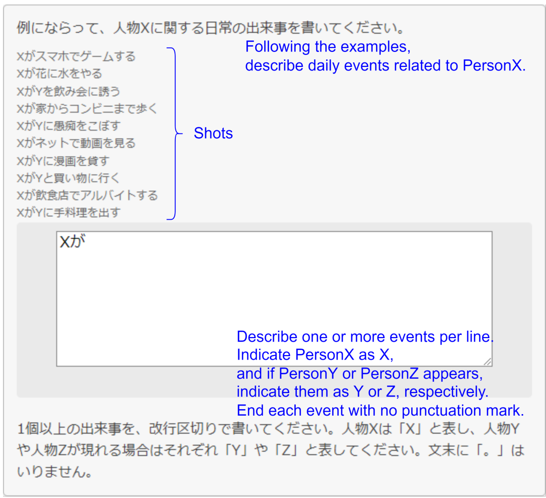
While ConceptNet and ATOMIC are acquired by crowdsourcing, ASER and TransOMCS are automatically built. On one hand, a large-scale graph can be built easily in an automatic way, but it is difficult to obtain knowledge not appearing in the text. On the other hand, crowdsourcing can gather high-quality data, but it is expensive in terms of both money and time.

There is a method that uses crowdsourcing and neural language models together to build an event knowledge graph (West et al., 2022). Although it is possible to acquire a large-scale and high-quality graph, they assume that an initial graph, ATOMIC in this case, has already been available.

2.3 Neural Commonsense Models

There have been studies on storing knowledge in a neural form rather than a symbolic form. In particular, methods of considering neural language models as knowledge bases (Petroni et al., 2019; AlKhamissi et al., 2022) have been developed. Bosselut et al. (2019) train COMET by finetuning pretrained Transformers on ATOMIC and ConceptNet, aiming at inference on unseen events and concepts. Gabriel et al. (2021) point out that COMET ignores discourse, introducing recurrent memory for paragraph-level information.

West et al. (2022) propose symbolic knowledge distillation where specific knowledge in a general



(a) For events

(b) For inferences (xEffect)

Figure 2: Examples of crowdsourcing interfaces. Crowdworkers are asked to describe events and inferences.

language model is distilled into a specific language model via a symbolic form. They expand ATOMIC using GPT-3 (Brown et al., 2020), filter the outputs using RoBERTa (Liu et al., 2019), and finetune GPT-2 (Radford et al., 2019) on the filtered ones.

3 Prompting Humans and a Language Model

We propose a method to build a knowledge graph for commonsense inference from scratch, with both crowdsourcing and a language model. In our method, we first construct a small-scale knowledge graph by crowdsourcing. Using the small-scale graph for prompts, we then extract commonsense knowledge from a language model. The flow of our method is shown in Figure 1. Building a knowledge graph from scratch only by crowdsourcing is expensive in terms of both money and time. Hence, the combination of crowdsourcing and a language model is expected to reduce the cost, especially in terms of time.

In other words, our method consists of the following two phases: (1) collecting a small-scale graph by crowdsourcing and (2) generating a large-scale graph by a language model. While crowdsourcing elicits commonsense from people, shots are used to extract knowledge from a language model. At this point, these phases are intrinsically the same, being considered as *prompting*. In the two phases, namely, we prompt people and a language model, respectively.

We build a commonsense inference knowledge graph in Japanese, with the concept of Section 3. We focus on an event knowledge graph such as ATOMIC (Sap et al., 2019) and ASER (Zhang et al., 2019). Handling commonsense on events and mental states would facilitate understanding of narratives and dialogues. We use Yahoo! Crowdsourcing in the first phase and HyperCLOVA JP (Kim et al., 2021), an LLM in Japanese, in the second phase.

3.1 Acquisition by Crowdsourcing

We first acquire a small-scale high-quality knowledge graph by crowdsourcing. With Yahoo! Crowdsourcing, specifically, we ask crowdworkers to write events and inferences. In a task, we provide them with 10 shots as a prompt for each event and inference. Note that for inferences, the prompts differ for each relation as mentioned later. We obtain a graph by filtering the collected inferences syntactically and semantically.

Events We ask crowdworkers to write daily events related to at least one person (PersonX). An example of the crowdsourcing task interface is shown in Figure 2a. The task provides instructions and 10 examples, and each crowdworker is asked to write at least one event. After all tasks are completed, we remove duplicate events. As a result, 257 events were acquired from 200 crowdworkers. We manually verified that all of the acquired events have a sufficient quality.

Inferences For the events collected above, we ask crowdworkers to write inferences about what happens and how a person feels before and after the events. In this paper, the relations for inference

	Inst #	Val #	Val %	IAA
Event	257	-	-	-
xNeed	504	402	79.76	39.85
xEffect	621	554	89.21	25.00
xIntent	603	519	86.07	36.11
xReact	639	550	86.07	31.82

Table 1: The statistics on events and inferences acquired by crowdsourcing.

1. Xがスマホでゲームする (X plays a game on X' s phone)
2. Xが花に水をやる (X waters flowers)
3. XがYを飲み会に誘う (X invites Y to a drinking party)

...

11. XがYに謝る (X apologizes to Y)

(a) For events

1. Xがにわか雨にあう。結果として、Xが軒先で雨宿りする。
(X gets caught in a shower. As a result, X takes shelter from the rain under the eaves.)
2. Xがネットで服を買う。結果として、Xが荷物を受け取る。
(X buys clothes on the Internet. As a result, X receives a package.)
3. Xが小腹を空かせる。結果として、Xが菓子を食べる。
(X gets hungry. As a result, X eats a snack.)

...

11. Xが筆箱を忘れる。結果として、Xが鉛筆を借りる。
(X forgets to bring X' s pencil case. As a result, X borrows a pencil.)

(b) For inferences (xEffect)

Figure 3: Prompts for generating events and inferences from an LLM. The underlined parts are generated.

are based on ATOMIC.[4] The following four are adopted as our target relations.

- What would have happened before (xNeed)
- What would happen after (xEffect)
- What PersonX would have felt before (xIntent)
- What PersonX would feel after (xReact)

While xNeed and xEffect are inferences about events, xIntent and xReact are inferences about mental states.

Three crowdworkers are hired per event. Given an instruction and 10 examples, each crowdworker is asked to write one inference. An example of the crowdsourcing task interface is shown in Figure 2b. We remove duplicate inferences as in the case of events, and then apply syntactic filtering[5] using the Japanese syntactic parser KNP[6].

The statistics of the acquired events and inferences are shown in the Inst # column of Table 1. The whole process costed 16,844 JPY (approximately 123 USD) by hiring 547 crowdworkers. Examples of acquired inferences are shown in Table 2.

3.2 Evaluation and Filtering

To examine the qualities of the inferences acquired by crowdsourcing, we crowdsource their evaluation. We ask three crowdworkers whether the inferences are acceptable or not and judge their acceptability by majority voting. The evaluation is crowdsourced independently for each relation. The inferences judged to be unacceptable by majority voting are filtered out.

The inferences collected in Section 3.1 are evaluated and filtered as above. The statistics are listed in the middle two columns of Table 1. As a result, we employed 465 crowdworkers and spent 8,679 JPY (approximately 63 USD). We also calculated Fleiss's κ as an inner-annotator agreement in the evaluation, which is shown in the rightmost column of Table 1.

There are several tendencies in the inferences filtered out, i.e., judged to be unacceptable. In some inferences, the order is reversed, as in the triple ⟨PersonX sleeps twice, xEffect, PersonX thinks that they are off work today⟩. Others are not plausible, as in ⟨PersonX surfs the Internet, xNeed, PersonX gets to the ocean⟩.

3.3 Generation from an LLM

From a small-scale high-quality knowledge graph acquired in Sections 3.1 and 3.2, we generate a large-scale knowledge graph with an LLM. We use the Koya 39B model of HyperCLOVA JP as a language model. Both events and inferences are generated by providing 10 shots. The shots are randomly chosen from the small-scale graph for each generation.

Events New events are generated by HyperCLOVA JP, using the events acquired in Section 3.2 as shots. An example prompt for event generation

[4]The relations are not exactly the same as those of ATOMIC. xIntent in this paper covers xIntent and xWant in ATOMIC, and tails for our xIntent and xReact may contain not mental states but events. The reason for the difference is that English and Japanese have different linguistic characteristics, i.e., it is difficult to collect knowledge in the same structure as the original.

[5]KNP determines if the subject is PersonX, if the tense is present, and if the event is a single sentence.

[6]https://nlp.ist.i.kyoto-u.ac.jp/?KNP

Head	Rel	Tail	Eval
Xが顔を洗う (X washes X's face)	xNeed	Xが水道で水を出す (X runs water from the tap)	✓
		Xが歯を磨く (X brushes X's teeth)	
	xEffect	Xがタオルを準備する (X prepares a towel)	✓
		Xが鏡に映った自分の顔に覚えのない傷を見つける (X finds an unrecognizable scar on X's face in the mirror)	✓
		Xが歯磨きをする (X brushes his teeth)	✓
	xIntent	スッキリしたい (Want to feel refreshed)	✓
		眠いのでしゃきっとしたい (Sleepy and Want to feel refreshed)	✓
	xReact	さっぱりして眠気覚ましになる (Feel refreshed and shake off X's sleepiness)	✓
		きれいになる (Be clean)	✓
		さっぱりした (Felt refreshed)	✓

Table 2: Examples of inferences acquired through crowdsourcing. Triples with ✓ in the eval column were judged to be acceptable by the evaluation in Section 3.2.

Rel	Template
xNeed	hためには、t必要がある。 (To h, need to t.)
xEffect	h。結果として、t。 (h. As a result, t.)
xIntent	hのは、tと思ったから。 (h because felt t.)
xReact	hと、tと思う。 (h then feel t.)

Table 3: The templates of shots for an LLM. h and t stand for head and tail, respectively. When generating, t is extracted.

	Inst #	Val %	IAA
Event	1,471	-	-
xNeed	9,403	80.81	36.07
xEffect	8,792	85.45	34.03
xIntent	10,155	86.06	43.42
xReact	10,941	90.30	21.51

Table 4: The statistics of events and inferences generated from an LLM. % Val and IAA are the evaluation results of 500 randomly selected inferences.

is shown in Figure 3a. We generate 10,000 events, remove duplicates, and apply the same syntactic filtering as in Section 3.1.

Inferences As in event generation, the inferences acquired in Sections 3.1 and 3.2 are used as shots. We generate 10 inferences for each event and remove duplicate triples. While we simply list the shots as a prompt in event generation, different prompts are used for each relation in inference generation. An example prompt for xEffect generation is shown in Figure 3b. Shots are given in natural language, and tails are extracted by pattern matching. Shot templates for each relation are shown in Table 3. Finally, the syntactic filtering is applied to obtain the graph.

The statistics of events and inferences generated by HyperCLOVA JP are shown in Table 4, and the results of the evaluation and the inter-annotator agreement are also shown in Table 4. For this evaluation, we sampled 500 inferences per relation. We hired 409 crowdworkers for a fee of 7,260 JPY (approximately 53 USD) in total. A comparison with Table 1 indicates that the quality is as good as those written by crowdworkers. Examples of generated inferences are shown in Table 5.

The generated knowledge graph in Japanese reflects the culture of Japan, such as ⟨PersonX goes to the office, xNeed, *PersonX takes a train*⟩. This fact indicates the importance of building from scratch for a specific language, rather than translating a similar dataset in a different language, which emphasizes the value of our method proposed in this paper.

4 Analysis on the Built Knowledge Graph

4.1 Effect of Filtering

In this paper, a small-scale knowledge graph is collected as in Sections 3.1 and 3.2, and a large-scale knowledge graph is generated as in Section 3.3. Here, we examine how effective the filtering in Section 3.2 is. As an experiment, we use filtered and unfiltered small-scale graphs as prompts to generate a large-scale graph. Then, we randomly select 500 generated triples for each relation and evaluate them by crowdsourcing as in Section 3.2. Note that the results for the filtered triples are the same as Section 3.3. For the triples without filtering, we crowdsourced again, paying 393 croweworkers 7,260 JPY (approximately 53 USD).

The ratios of appropriate inferences with and without filtering are shown in Table 6. For all rela-

Head	Rel	Tail
Xがコンビニへ行く (X goes to a convenience store)	xNeed	Xが財布を持っている (X has X's wallet), Xが外出する (X goes out), Xが外出着に着替える (X changes into going-out clothes), Xが財布を持って出かける (X goes out with X's wallet), Xが外へ出る (X goes outside)
	xEffect	Xが買い物をする (X goes shopping), Xが雑誌を立ち読みする (X browses through magazines), XがATMでお金をおろす (X withdraws money from ATM), Xが弁当を買う (X buys lunch), Xがアイスを買う (X buys ice cream)
	xIntent	何か買いたいものがある (Want to buy something), 雑誌を買う (Buy a magazine), 飲み物を買おう (Going to buy a drink), 飲み物や食べ物を買いたい (Want to buy a drink or food), なんでもある (There is everything X wants)
	xReact	何か買いたいものがある (Want to buy something), 何か買う (Buy something), 何か買おう (Going to buy something), 何か買いたくなる (Come to buy something), ついでに何か買ってしまう (Buy something incidentally)

Table 5: Examples of inferences generated from an LLM. For each relation, five examples are displayed.

	xNeed	xEffect	xIntent	xReact
w/o Fltr	**81.62**	82.42	83.84	89.29
w/ Fltr	80.81	**85.45**	**86.06**	**90.30**

Table 6: The ratios of appropriate inferences with respect to filtering. Note that the w/ Fltr row is the same as the Val % column in Table 4.

tions except xNeed, filtering improves the quality of triples.

4.2 Comparison between humans and a Language Model

In Section 3.1, on one hand, we asked crowdworkers to describe events and inferences. In Section 3.3, on the other hand, we had an LLM generate them. Here, we compare a small-scale knowledge graph by crowdsourcing and a large-scale one from a language model, i.e., inferences generated by humans and a computer. Because the relationships between events can be largely divided into contingent and temporal relationships (Bethard et al., 2008), we adopt contingency and time interval as metrics for comparison.

Of the four relations, we focus on xEffect as a representative, which is a typical causal relation. For each head of the triples acquired by crowdsourcing in Sections 3.1 and 3.2, we generate three tails using the language model in Section 3.3 and compare them with the original tails. From the 554 heads for xEffect in the small-scale graph, we obtained 586 unique inferences.

Contingency One measure is how likely a given event is to be followed by a subsequent event. Crowdworkers are given a pair of events in an xEffect relation and asked to judge how likely the following event is to happen on a three-point scale: "must happen," "likely to happen," and "does not happen." We ask three crowdworkers per inference and calculate the median of them.

Time Interval The other measure is the time interval between the occurrence of an event and that of a subsequent event. As in the evaluation of contingency, crowdworkers are given a triple on xEffect. We ask them to judge the time interval between the two events in five levels: almost simultaneous, seconds to minutes, hours, days to months, and longer. Finally, the median is calculated from the results of three crowdworkers.

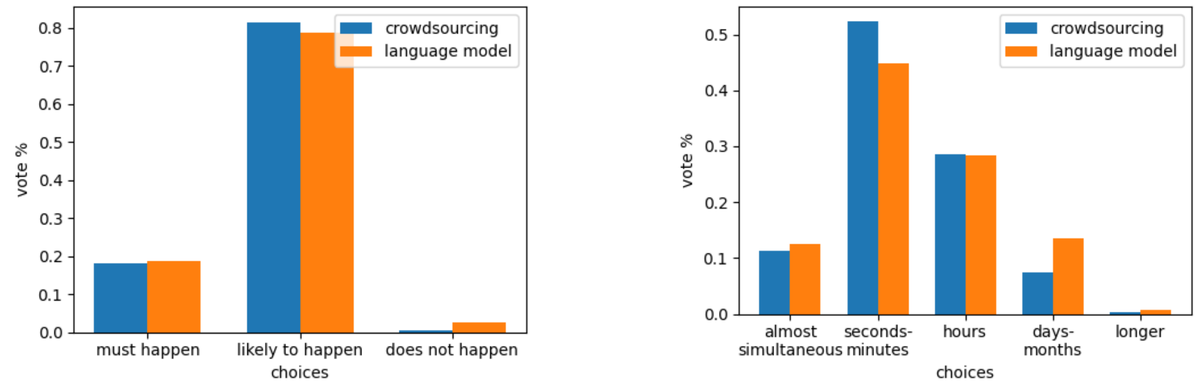
The comparison between humans and a language model for each measure is shown in Figure 4. Figure 4a shows that the subsequent events by crowdsourcing, or humans, are slightly more probable. In Figure 4b, the inferences generated by an LLM have a longer time interval. This result indicates a difference in the results of prompting humans and a language model; for xEffect, humans infer events that happen relatively soon, while a language model infers events that happen a bit later.

5 Japanese Neural Commonsense Models

We train Japanese neural commonsense models using the knowledge graph constructed in Section 4. Japanese versions of GPT-2 (Radford et al., 2019) and T5 (Raffel et al., 2020) are finetuned to generate inferences on unseen events. We conduct automatic and manual evaluations and compare their performances.

5.1 Training

Base models and data Using the constructed knowledge graph, we finetune pretrained models to construct Japanese neural commonsense models. To evaluate inferences on unseen events, triples in



(a) Contingency

(b) Time interval

Figure 4: A comparison between crowdsourcing and language model generation.

the knowledge graph are randomly partitioned into training and test sets at a ratio of 9:1. For pretrained models, we adopt Japanese T5[7] and GPT-2[8] of the Hugging Face implementation (Wolf et al., 2020).

Input format to models The input for each model differs. See Appendix C for the full input formats for each model. Since T5 is a seq2seq model, the head and the relation are given in the form of "$r : h$" as an input, and the tail is given as the correct output. The relation for T5 is changed to a natural language sentence. For example, "xNeed" is rewritten to "What event occurs before this statement?" The inputs for all relations are shown in Appendix C. For GPT-2, since it predicts the next word, the head and the relation are given as an input, and the model is trained to output the tail. Since the relations are not included in the vocabulary of the pretrained models, they are added as special tokens. In the constructed knowledge graph, the subject of an event is generalized as "X," but it would be better to change it into a natural expression as the input to the pretrained models. We randomly replace the subject with a personal pronoun during training and inference. To confirm this effect, in section 5.2, we also train GPT-2 with the subject represented as "X." We denote this as GPT-2_{X}.

5.2 Evaluation

We generate inferences for the head events in the test set using the trained Japanese neural commonsense models and evaluate the inferences automatically and manually. We also show correlation between the automatic and manual evaluations. Examples of the inference results are shown in Appendix C. The average output length and the number of unique words are also reported in Appendix C. In summary, the number of unique words in GPT-2 is larger than that in T5 (392 unique words), with a difference of 35 to 59 words.

Model	AR	MP	BS	BLEU
T5	87.5	1.64	90.26	18.57
GPT-2	**91.0**	**1.73**	**92.31**	18.26
GPT-2_{X}	**91.0**	1.68	92.03	**18.99**

Table 7: Total evaluation scores. AR, MP, and BS indicate the accept rate, the mean point, and BERTScore, respectively.

Rel	AR	MP	BS	BLEU
xNeed	88.9	1.58	92.73	22.22
xEffect	92.4	1.72	93.98	22.24
xIntent	88.9	1.66	90.12	9.91
xReact	93.8	1.98	93.00	11.83

Table 8: Evaluation scores of GPT-2 for each relation.

Automatic evaluation We calculate BLEU (Papineni et al., 2002) and BERTScore (Zhang* et al., 2020) as automatic metrics. Table 7 shows these results. GPT-2_{X} and GPT-2 performed the best in BLEU and BERTScore, respectively.

Manual evaluation Using crowdsourcing, we evaluate how likely the generated inferences are. Following the previous study (West et al., 2022), we show crowdworkers two events (a head and a tail) and a relation. Then, we ask them to evaluate the appropriateness of the inference by choos-

[7]https://huggingface.co/megagonlabs/t5-base-japanese-web
[8]https://huggingface.co/nlp-waseda/gpt2-small-japanese

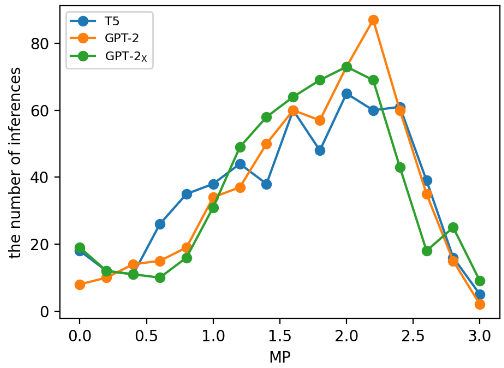


Figure 5: The number of inferences for each MP.

ing from the following options: "always," "often," "sometimes," and "never." The choices are displayed with an appropriate verb for each relation (e.g., "always happens" for xEffect). Five crowdworkers are asked to judge per inference. For each inference, the numbers of crowdworkers who choose "never" and other than "never" (i.e., at least "sometimes") are used to determine the majority vote. The acceptance rate (AR), the proportion of inferences in which more crowdworkers choose other than "never." By assigning 0 to 3 points each to "never," "sometimes," "often," and "always," we also calculate the mean point (MP) as the average score of all the inferences. Table 7 shows these results. AR is higher than 85% for all models, indicating that the inferences for unseen events are almost correct. GPT-2 obtained the highest scores for both AR and MP. Furthermore, as shown in Table 8, ARs of xNeed and xIntent are lower than xEffect and xReact, respectively, for all models. This can be attributed to the fact that we used an autoregressive model, which makes it difficult to infer in reverse order of time.

Although the replacement of subjects did not make a difference in AR, there is a difference in the distributions of MP as shown in Figure 5. The number of crowdworkers who chose "never" for the inference of GPT-2 is less than half of that for GPT-2_X. This result indicates that it is better for the model to replace subjects "X" with personal pronouns.

Correlation between the evaluation metrics Table 9 shows the correlation coefficients between the manual and automatic evaluation metrics. The correlation coefficients between the manual metrics (AR and MP) and BERTScore are positive, while those between the manual metrics and BLEU are negative or no correlation. It seems that BERTScore, which uses vector representations, can evaluate equivalent sentences with different expressions, but BLEU, which is based on n-gram agreement, cannot correctly judge the equivalence. One of the reasons for the negative correlation in BLEU is that many inferences of the mental state consist of a single word in Japanese, such as "tired" and "bored," for both the gold answer and the generated result. In this case, BLEU tends to be low because the words are rarely matched, but the shorter the sentences are, the easier it is for the model to generate appropriate results.

	AR	MP	BS	BLEU
AR	1.00	0.75	0.59	-0.11
MP	-	1.00	0.43	-0.46
BS	-	-	1.00	0.30
BLEU	-	-	-	1.00

Table 9: Correlation coefficients between automatic and manual evaluation metrics.

6 Conclusion

We proposed a method for building a knowledge graph from scratch with both crowdsourcing and a language model. Based on our method, we built a knowledge graph on events and mental states in Japanese using Yahoo! Crowdsourcing and HyperCLOVA JP. Since designing tasks for having humans describe commonsense and engineering prompts for having a language model generate knowledge are similar to each other, we compared the characteristics of them. We evaluated the graph generated by HyperCLOVA JP and found that it was similar in quality to the graph written by humans.

Furthermore, we trained a neural commonsense model for event inference based on the built knowledge graph. We attempted inference generation for unseen events by finetuning GPT-2 and T5 in Japanese on the built graph. The experimental results showed that these models are able to generate acceptable inferences for events and mental states.

We hope that our method for building a knowledge graph from scratch and the acquired knowledge graph lead to further studies on commonsense inference, especially in low-resource languages.

Ethical Considerations

For acquiring a small-scale event knowledge graph and analyzing the built graph, we crowdsource commonsense knowledge, using Yahoo! Crowdsourcing. Specifically, we collect the descriptions of commonsense, filter them, and explore the characteristics of the graph by crowdsourcing. Fees and the numbers of crowdworkers per process are in the text. In total, we employed 1,814 crowdwoekers paying 40,043 JPY (apploximately 288 USD). We obtained a consent from crowdworkers on the platform of Yahoo! Crowdsourcing.

The event knowledge graph and the neural commonsense models built in this paper help computers understand commonsense. A commonsense-aware computer, for example, can answer open-domain questions by humans, interpret human statements in detail, and converse with humans naturally. However, such graphs and models may contain incorrect knowledge even with filtering, which leads the applications to harmful behavior.

References

Badr AlKhamissi, Millicent Li, Asli Celikyilmaz, Mona Diab, and Marjan Ghazvininejad. 2022. A review on language models as knowledge bases.

Steven Bethard, William Corvey, Sara Klingenstein, and James H. Martin. 2008. Building a corpus of temporal-causal structure. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Sumithra Bhakthavatsalam, Chloe Anastasiades, and Peter Clark. 2020. Genericskb: A knowledge base of generic statements.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. COMET: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4762–4779, Florence, Italy. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Saadia Gabriel, Chandra Bhagavatula, Vered Shwartz, Ronan Le Bras, Maxwell Forbes, and Yejin Choi. 2021. Paragraph-level commonsense transformers with recurrent memory. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(14):12857–12865.

Deepanway Ghosal, Pengfei Hong, Siqi Shen, Navonil Majumder, Rada Mihalcea, and Soujanya Poria. 2021. CIDER: Commonsense inference for dialogue explanation and reasoning. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 301–313, Singapore and Online. Association for Computational Linguistics.

Deepanway Ghosal, Siqi Shen, Navonil Majumder, Rada Mihalcea, and Soujanya Poria. 2022. CICERO: A dataset for contextualized commonsense inference in dialogues. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5010–5028, Dublin, Ireland. Association for Computational Linguistics.

Jena D. Hwang, Chandra Bhagavatula, Ronan Le Bras, Jeff Da, Keisuke Sakaguchi, Antoine Bosselut, and Yejin Choi. 2021. (comet-) atomic 2020: On symbolic and neural commonsense knowledge graphs. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(7):6384–6392.

Boseop Kim, HyoungSeok Kim, Sang-Woo Lee, Gichang Lee, Donghyun Kwak, Jeon Dong Hyeon, Sunghyun Park, Sungju Kim, Seonhoon Kim, Dongpil Seo, Heungsub Lee, Minyoung Jeong, Sungjae Lee, Minsub Kim, Suk Hyun Ko, Seokhun Kim, Taeyong Park, Jinuk Kim, Soyoung Kang, Na-Hyeon Ryu, Kang Min Yoo, Minsuk Chang, Soobin Suh, Sookyo In, Jinseong Park, Kyungduk Kim, Hiun Kim, Jisu Jeong, Yong Goo Yeo, Donghoon Ham, Dongju Park, Min Young Lee, Jaewook Kang, Inho Kang, Jung-Woo Ha, Woomyoung Park, and Nako Sung. 2021. What changes can large-scale language models bring? intensive study on HyperCLOVA: Billions-scale Korean generative pretrained transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3405–3424, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Alisa Liu, Swabha Swayamdipta, Noah A. Smith, and Yejin Choi. 2022. Wanli: Worker and ai collaboration for natural language inference dataset creation.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849, San Diego, California. Association for Computational Linguistics.

Nasrin Mostafazadeh, Aditya Kalyanpur, Lori Moon, David Buchanan, Lauren Berkowitz, Or Biran, and Jennifer Chu-Carroll. 2020. GLUCOSE: GeneraLized and COntextualized story explanations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4569–4586, Online. Association for Computational Linguistics.

Kazumasa Omura, Daisuke Kawahara, and Sadao Kurohashi. 2020. A method for building a commonsense inference dataset based on basic events. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2450–2460, Online. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Alec Radford and Karthik Narasimhan. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

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. *Journal of Machine Learning Research*, 21(140):1–67.

Hannah Rashkin, Maarten Sap, Emily Allaway, Noah A. Smith, and Yejin Choi. 2018. Event2Mind: Commonsense inference on events, intents, and reactions. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 463–473, Melbourne, Australia. Association for Computational Linguistics.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S. Gordon. 2011. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *AAAI Spring Symposium on Logical Formalizations of Commonsense Reasoning*, Stanford University.

Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A. Smith, and Yejin Choi. 2019. Atomic: An atlas of machine commonsense for if-then reasoning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):3027–3035.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1).

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Peter West, Chandra Bhagavatula, Jack Hessel, Jena Hwang, Liwei Jiang, Ronan Le Bras, Ximing Lu, Sean Welleck, and Yejin Choi. 2022. Symbolic knowledge distillation: from general language models to commonsense models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4602–4625, Seattle, United States. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Ann Yuan, Daphne Ippolito, Vitaly Nikolaev, Chris Callison-Burch, Andy Coenen, and Sebastian Gehrmann. 2021. Synthbio: A case study in faster curation of text datasets. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. SWAG: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 93–104, Brussels, Belgium. Association for Computational Linguistics.

Hongming Zhang, Daniel Khashabi, Yangqiu Song, and Dan Roth. 2020. Transomcs: From linguistic graphs to commonsense knowledge. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 4004–4010. International Joint Conferences on Artificial Intelligence Organization. Main track.

Hongming Zhang, Xin Liu, Haojie Pan, Yangqiu Song, and Cane Wing-Ki Leung. 2019. Aser: A large-scale eventuality knowledge graph.

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 An Example of Crowdsourced Evaluation

We evaluate and filter the inferences obtained in Sections 3.1 and 3.3 by crowdsourcing. An example of the interface for evaluating an xEffect inference is shown in Figure 6.

B Hyperparameter Details

We generate a large-scale knowledge graph using HyperCLOVA JP in Section 3.3. The hyperparameters for the generation is shown in Table 10.

With the built knowledge graph, we finetune Japanese T5 and GPT-2 on the task of commonsense inference in Section 5. The hyperparameters for T5 and GPT-2 are shown in Table 11.

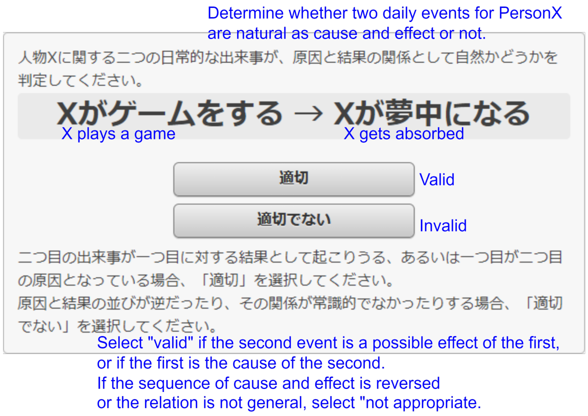


Figure 6: An example of evaluation regarding xEffect relations. We ask three crowdworkers whether a given inference is acceptable or not.

Max tokens	32
Temperature	0.5
Top-P	0.8
Top-K	0
Repeat penalty	5.0

Table 10: Hyperparameters for event and inference generation with HyperCLOVA JP.

C Details of Neural Commonsense Models

Table 12 shows the average output length and the number of unique words for each model. The average output length of T5 is longer than those of GPT-2s, but GPT-2s have the greater numbers of unique words than T5.

Table 13 shows the input formats to the models. The prompts to T5 may not be the best; prompt-engineering could improve the results.

Examples of outputs are shown in Table 14. We can see that the obtained outputs are acceptable to humans. The outputs vary for each model.

	T5	GPT-2
Batch size	64	64
Learning rate	5e-5	5e-5
Weight decay	0.0	0.0
Adam betas	(0.9, 0.999)	(0.9, 0.999)
Adam epsilon	1e-8	1e-8
Max grad norm	1.0	1.0
Num epochs	30	3
LR scheduler type	Linear	Linear
Warmup steps	0	0

Table 11: Hyperparameters for finetuning T5 and GPT-2 on the knowledge graph.

Model	Avg Out Len	Uniq Word #
T5	5.29	392
GPT-2	5.03	451
GPT-2_X	5.03	436

Table 12: Average output length and the number of unique words.

Model	Rel	Encoder Input	Decoder Input
T5	xNeed	この文の前に起こるイベントは何ですか？: h (What event occurs before this statement?: h)	t
	xEffect	このイベントの次に発生する事象は何ですか？: h (What is the next event to occur after this event?: h)	t
	xIntent	次の文の発生した理由は何ですか？: h (What is the reason for the occurrence of the following statement?: h)	t
	xReact	次の文の後に感じることは何ですか？: h (What will be felt after the following statement?: h)	t
GPT-2	xNeed	-	h xNeed t
	xEffect	-	h xEffect t
	xIntent	-	h xIntent t
	xReact	-	h xReact t

Table 13: The input formats for training. Note that h and t denote a head and a tail.

Model	Input	Output
T5	この文の前に起こるイベントは何ですか？:あなたが友人たちと旅行に出かける (What event occurs before this statement?: You go on a trip with your friends)	あなたが車を運転する (You drive a car)
	このイベントの次に発生する事象は何ですか？:あなたが友人たちと旅行に出かける (What is the next event to occur after this event?: You go on a trip with your friends)	あなたが楽しい時間を過ごす (You have a good time)
	次の文の発生した理由は何ですか？:あなたが友人たちと旅行に出かける (What is the reason for the occurrence of the following statement?: You go on a trip with your friends)	楽しい (Have fun)
	次の文の後に感じることは何ですか？:あなたが友人たちと旅行に出かける (What will be felt after the following statement?: You go on a trip with your friends)	楽しい (Have fun)
GPT-2	僕が友人たちと旅行に出かけるxNeed (I go on a trip with your friends xNeed)	僕がパスポートを取得する (I get my passport)
	僕が友人たちと旅行に出かけるxEffect (I go on a trip with your friends xEffect)	僕が楽しい時間を過ごす (I have a good time)
	僕が友人たちと旅行に出かけるxIntent (I go on a trip with your friends xIntent)	楽しいことがしたい (Want to have fun)
	僕が友人たちと旅行に出かけるxReact (I go on a trip with your friends xReact)	楽しい (Feel fun)

Table 14: Examples of the inferences generated by T5 and GPT-2.