SituatedGen: Incorporating Geographical and Temporal Contexts into Generative Commonsense Reasoning

Anonymous EMNLP submission

Abstract

Recently, commonsense reasoning in text generation has attracted much attention. Generative commonsense reasoning is the task that requires machines, given a group of keywords, to compose a single coherent sentence with commonsense plausibility. While existing datasets targeting generative commonsense reasoning focus on everyday scenarios, it is unclear how well machines reason under specific geographical and temporal contexts. We formalize this challenging task as SituatedGen, where machines with commonsense should generate a pair of contrastive sentences given a group of keywords including geographical or temporal entities. We introduce a corresponding English dataset consisting of 9,060 contrastive sentence pairs, which are built upon several existing commonsense reasoning benchmarks with minimal manual labor. Experiments show that state-of-the-art text generation models struggle to generate sentences with commonsense plausibility and still lag far behind human performance.

1 Introduction

In recent years, there has been a substantial growth in new benchmarks evaluating commonsense reasoning for natural language processing (NLP) models, especially large-scale Pretrained Language Models (PLMs). Most existing commonsense reasoning benchmarks adopt natural language *understanding* formats due to easy evaluation (e.g., accuracy), including multiple-choice question answering (Talmor et al., 2019; Sap et al., 2019; Huang et al., 2019; Lin et al., 2021), natural language inference (Bhagavatula et al., 2020), and detecting true/false statements (Onoe et al., 2021; Singh et al., 2021). However, datasets measuring commonsense knowledge in natural language *generation* are still relatively scarce. We aim to fill this research gap since advancing commonsense reasoning skills of text generation models benefits many downstream applications such as document summarization (Sha, 2020), story writing (Yao et al., 2019) and dialogue response generation (Mou et al., 2016).

CommonGen (Lin et al., 2020), a generative commonsense reasoning challenge, has attracted wide attention recently. Given a set of keywords (e.g., `{dog, frisbee, catch, throw}`), the task requires models to compose a plausible sentence describing everyday scenario using all the provided keywords (e.g., "*The dog catches the frisbee when the boy throws it.*"). While CommonGen focuses on social and physical commonsense in everyday life, it is unclear how well current commonsense generation models reason with factual knowledge about specific entities, which is referred to as *entity commonsense* (Onoe et al., 2021). In this work, we mainly consider geographical and temporal entities, as they provide extra-linguistic contexts (Zhang and Choi, 2021) for commonsense reasoning and appear in a significant proportion of existing commonsense benchmarks (Section 4.2). Although Zhang and Choi (2021) have studied the effect of geographical and temporal contexts on Question Answering (QA), to the best of our knowledge, we are the first to incorporate these situations into generative commonsense reasoning.

Furthermore, we argue that geographical and temporal contexts are important for commonsense reasoning. On the one hand, basic knowledge about geography and time is part of human commonsense (Allen, 1983; Bhatt and Wallgrün, 2014), such as "*Earth rotates on its axis once in **24 hours**.*" On the other hand, certain types of commonsense knowledge are correlated with specific situations (Yin et al., 2021). For example, "*July is summer*" is true for people living in the northern hemisphere, while those living in the southern hemisphere would agree that "*July is winter*".

Our proposed task SituatedGen (**Situated Gen**erative Commonsense Reasoning) requires the machines to generate a pair of contrastive sentences (formally speaking, *antithesis*) with commonsense plausibility, given a group of keywords including geographical or temporal entities. For example, when provided with `[July, United States, winter, Australia, summer, July]`, a reasonable output could be "*July is summer in the United States. July is winter in Australia.*", while a slightly different version "*July is summer in Australia. July is winter in the United States.*" does not adhere to commonsense.

There are two key challenges for machines to solve the SITUATEDGEN task. The first challenge is *situated semantic matching*. In order to generate a pair of contrastive sentences, machines need to split the keywords into two groups (either explicitly or implicitly) based on geographical/temporal relevance and perform relational reasoning (Nickel et al., 2016) within/between the keyword groups. Second, models should master *compositional generalization* (Keysers et al., 2020), reasoning over new combinations of keywords during the inference stage instead of memorizing existing keywords matching results.

To study the challenging SITUATEDGEN task, we construct a corresponding large-scale English dataset containing 9,060 pairs of situated commonsense statements. We design an automatic pipeline to collect data at scale with quality assurance and minimal human annotation efforts. Concretely, we derive commonsense statements with geographical or temporal contexts from existing commonsense benchmarks and mine contrastive sentence pairs based on entity-masked sentence similarity. We further manually filter out invalid examples in the test set to ensure the evaluation soundness. To assess the difficulty of our dataset, we conduct baseline experiments on pretrained text generation models with automatic evaluation metrics. Results show these models lag far behind human performance, indicating that current models struggle to generate sentences adhering to commonsense under the SITUATEDGEN setting. We believe that SITUATEDGEN could serve as a complement to COMMONGEN and enrich the resource for evaluating constrained commonsense text generation in a more realistic setting.

The main contributions of this work are threefold:

- **Task.** We incorporate geographical and temporal contexts into generative commonsense reasoning and propose a novel task SITUATEDGEN.
- **Resource.** We construct a large-scale dataset in a non-trivial way to facilitate the studies of situated generative commonsense reasoning. The dataset will be released and contribute to the commonsense reasoning community.
- **Evaluation.** We benchmark the performance of state-of-the-art pretrained text generation models on our dataset and demonstrate the difficulty of the task with a significant gap between machine and human performance.

2 Related Work

Constrained Commonsense Text Generation. Constrained Commonsense Text Generation (Bhargava and Ng, 2022) requires PLMs to generate commonsense text subject to a set of constraints. Commonsense generation models are currently evaluated by three tasks. COMMONSENSE EXPLANATION aims to generate an explanation for why a model selects a candidate answer to a given question. α NLG (Bhagavatula et al., 2020) is another commonsense generation task. The artificial intelligence models are provided with two observations in chronological order and need to generate a plausible hypothesis/explanation describing what happened between the observations. Obviously, SITUATEDGEN is different from these two tasks. In COMMONGEN (Lin et al., 2020), models should compose a plausible sentence describing everyday scenario using all the provided concepts. This task has attracted much attention recently, and researchers advance machine performance on the dataset with contrastive learning (Li et al., 2021), prototype editing (Liu et al., 2021b), scene knowledge graph (Wang et al., 2021), etc. Our proposed task differs to COMMONGEN in the focus on composing a *pair* of contrastive sentences instead of a *single* sentence and incorporating extra-linguistic contexts.

NLP Benchmarks with Geographical and Temporal Contexts. There are many emerging benchmarks in NLP that incorporate extra-linguistic contexts such as geographical and temporal contexts. TEMPLAMA (Dhingra et al., 2021) and GEOMLAMA (Yin et al., 2022) probe language models with masked text prompts to query geographical and temporal knowledge. In question answering, MCTACO (Zhou et al., 2019), TORQUE (Ning et al., 2020) and TIMEQA (Chen et al., 2021) contains challenging questions involving temporal commonsense reasoning over duration, frequency, temporal order, and other various aspects of events. SITUATEDQA (Zhang and Choi, 2021) is made up of open-domain questions whose answers vary across different geographical and temporal contexts. TIMEDIAL (Qin et al., 2021) studies temporal reasoning in dialogues with a multiple-choice cloze task. In vision-and-language tasks, GD-VCR (Yin et al., 2021) and MaRVL (Liu et al., 2021a) aim to collect commonsense questions and statements that are visually grounded and geographically diverse. Our dataset SITUATEDGEN also considers such geographical and temporal contexts/reasoning in language. However, our work is different from the previous ones in that we choose the task of generative commonsense reasoning, pioneered by Lin et al. (2020), as it focuses on the commonsense reasoning capabilities of NLG models rather than NLU. We note that benchmarks targeting at machines commonsense in NLG are far less than those for NLU and thus require more empirical attention.

3 Task Definitions and Challenges

We use antithesis generation for evaluating generative commonsense reasoning under extra-linguistic contexts. In this section, we first introduce the definition of antithesis as a literary device, followed by a mathematical formulation of situated generative commonsense reasoning. We then analyze the two key challenges of our proposed task.

3.1 Definitions

Antithesis. Antithesis refers to a figure of speech that expresses an opposition of ideas with a parallel grammatical structure of words, clauses, or sentences (Lloyd, 1911; Bridgwater, 1963). An example of antithesis could be Neil Armstrong's famous quote "*That's one small step for a man, one giant leap for mankind*". In this work, we adopt the definition of sentence-level antithesis, which means two simple sentences with similar syntactic structure creating a contradiction in semantics. We emphasize on the commonsense plausibility rather than the rhetoric effect of antithesis within the scope of this paper.

Extra-Linguistic Contexts. Following Zhang and Choi (2021), we focus on two context types: geographical (GEO) and temporal (TEMP). GEO defines each context value as a geopolitical entity ("GPE"). TEMP defines each context value as timestamp ("DATE", "TIME", "EVENT").

Contextual Dependence. We define that a contrastive sentence pair is *context-dependent* if swapping any of the GEO or TEMP entities between the two sentences could lead to contradiction with commonsense yet grammatical correctness. For example, for the sentence pair "*July is summer in China. July is winter in Australia.*", if the two GEO entities "China" and "Australia" are swapped, the resulting sentences do not adhere to commonsense anymore: "*July is summer in Australia. July is winter in China.*" This indicates that they are context-dependent.

Contextual dependence is crucial for a proper evaluation of the generation results. Because sentence pairs that do not satisfy context dependence may have multiple valid answers (swapping the entity words leads to an extra correct answer), the metrics introduced in Section 6 cannot make a sound evaluation with only a single reference.

Situated Generative Commonsense Reasoning. We modify the mathematical formulation of the task COMMONGEN to define SITUATEDGEN. The input of the task is a multiset[1] consisting of k keywords $x = [c_1, c_2, ..., c_k] \in \mathcal{X}$, where each keyword $c_i \in \mathcal{C}$ is a noun or entity, a single word or phrase. We denote $\mathcal{X}$ as all possible combinations of keywords and $\mathcal{C}$ as the vocabulary of keywords. Keywords in x should contain at least two GEO or TEMP entities and two other keywords[2].

The output of the task is an unordered pair of coherent and plausible sentences $y = \{s_1, s_2\} \in \mathcal{Y}$ that satisfies the following conditions: 1) the sentence pair includes all keywords in x; 2) each sentence has at least one GEO or TEMP keyword; 3) each sentence is geographical-temporal-semantically correct; 4) s_1 and s_2 form a pair of contrastive sentences, or antithesis; 5) s_1 and s_2 are context-dependent. The goal of the task is to learn a function $f : \mathcal{X} \rightarrow \mathcal{Y}$ that maps a group of pf keywords x to a pair of sentences y.

3.2 Challenges

Situated Semantic Matching. As the goal of our task is to generate a pair of sentences instead of a single sentence, machines need to explicitly or implicitly classify the keywords into two subgroups

[1]Multiset is a set that allows multiple instances for each of its elements.

[2]We do not explicitly provide the types of keywords in our dataset. The models are expected to infer which keyword is GEO or TEMP if needed.

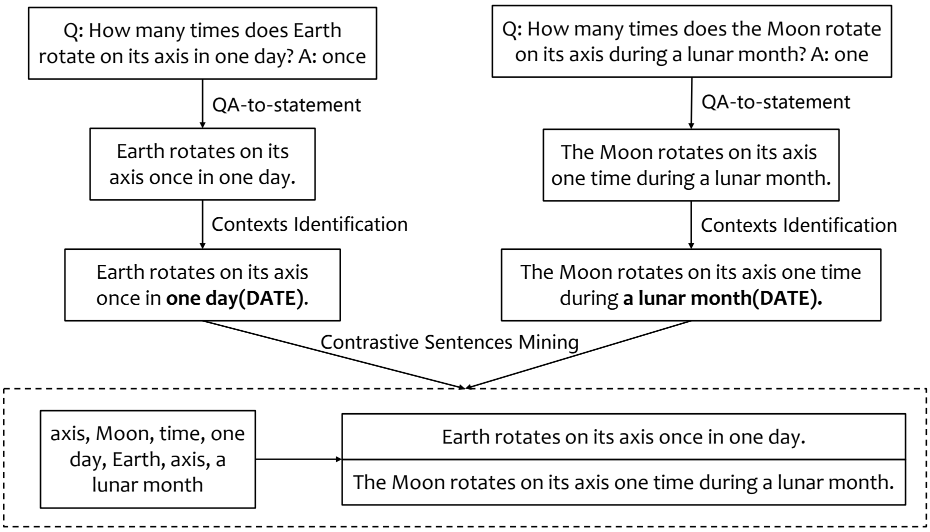

Figure 1: An overview of data collection pipeline. Inside the dotted box is a final example in the dataset.

based on their geographical and temporal semantic relevance, so as to generate one commonsense sentence with each subgroup. For example, given `[July, China, winter, Australia, summer, July]`, the resulting keyword subgroups should be `{July, China, summer}` and `{July, winter, Australia}`.

During the process of keyword grouping and matching, machines determine which keywords are more relevant to each other with relational reasoning (Nickel et al., 2016) over factual knowledge about these nouns and entities, a.k.a. *entity knowledge* (Zhang and Choi, 2021), such as geographical location, temporal order, physical rules, social customs, etc. The matching process is important since wrong grouping results will lead to generated sentences without commonsense plausibility[3].

In order to prevent the model from exploiting "shortcuts" (Gururangan et al., 2018; Tu et al., 2020) to group keywords based on syntactic forms instead of semantic meanings, we ask the model to generate contrastive sentences that are syntactically similar, rather than two coherent (yet possibly irrelevant) sentences.

Compositional Generalization. In machine learning practice, compositional generalization (Keysers et al., 2020) means that models can generalize to test examples of novel combinations after being exposed to the necessary components during training. Specifically, in the SITUATEDGEN task, the components refer to keywords. We ensure that there is no overlap among the keyword combinations in the training, validation and test set during the dataset collection. While humans can easily compose sentences with unfamiliar combinations of keywords, it is very challenging for the machines to make analogy and inference with unseen keyword combinations, instead of simply memorizing existing keyword combinations.

4 Dataset Collection

To study the SITUATEDGEN challenge, we construct a large-scale English dataset. We design a pipeline to collect high-quality data at scale with minimal manual annotation efforts. Figure 1 illustrates the overall pipeline for dataset collection, which consists of three steps:

1. **QA-to-statement.** Converting question-answer pairs of existing commonsense question answering benchmarks into corresponding statements.
2. **Contexts Identification.** Identifying all entities in a statement with an NER tagger and removing those statements without GEO and TEMP entities.
3. **Contrastive Sentences Mining.** Automatically mining contrastive sentence pairs (antithesis) from the remaining commonsense statements based on entity-masked sentence similarity.

[3] We note that under certain circumstances, wrong grouping results might produce correct answer via negative sentences. For example, the machine could generate "*July is **not** summer in Australia*" with `{July, Australia, summer}`. However, we observe that these are rare scenarios in our datasets and also uncommon expressions in everyday life, so we do not consider their confusing effects in our study.

Dataset	# Sent	# GEO	# TEMP	# GEO & TEMP	# Valid Sent
CREAK	5,779	868	552	153	1,573
StrategyQA	4,976	501	366	86	953
CommonsenseQA	10,962	487	215	12	714
ARC	7,787	165	426	52	643
OpenbookQA	6,493	31	119	5	155
Total	35,997	2,052	1,678	308	4,038

Table 1: Statistics of contexts identification results. "Sent" means the commonsense statements collected in Section 4.1. "GEO"/"TEMP" refer to statements with *only* geographical/temporal entities. "GEO & TEMP" refers to statements with *both* geographical and temporal entities. "Valid Sent" means the commonsense statements with GEO or TEMP contexts.

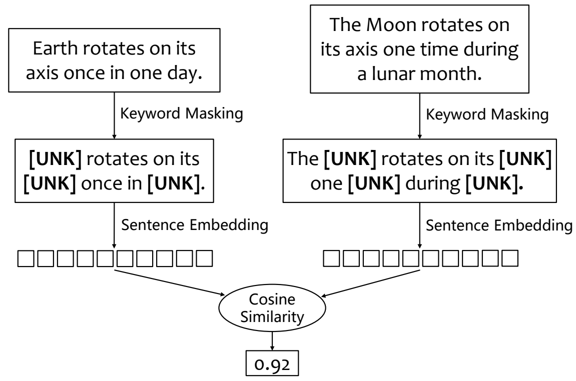

Figure 2: An illustration of the contrastive sentence mining algorithm.

4.1 QA-to-Statement

Our dataset is composed of commonsense statements, which are simple sentences describing commonsense knowledge, e.g., "*You would find many canals in Venice.*" In recent years, numerous commonsense reasoning benchmarks have been proposed and they form a potentially available commonsense knowledge base with high quality and diverse content. Inspired by recent benchmarks that are sourced from existing datasets (Zhang and Choi, 2021; Park et al., 2022), we aim to extract commonsense statements from these commonsense benchmarks[4].

We conduct a holistic study of commonsense reasoning datasets to date and select five different data sources after considering their size, annotation quality and reasoning difficulty. They are CREAK (Onoe et al., 2021), StrategyQA (Geva et al., 2021), CommonsenseQA (Talmor et al., 2019), ARC (Clark et al., 2018) and OpenbookQA (Mihaylov et al., 2018), respectively. We briefly introduce the nature of each dataset in Appendix A.1. Since the raw data come in different formats such as multiple-choice questions and Yes/No questions, we apply a specific preprocessing method for each dataset to transform them (i.e., question-answer pairs) into statements. The transformation details are also included in Appendix A.1. In general, we collected 35,997 commonsense statements from the five source datasets (statistics in Table 1).

[4]We assume that the knowledge in these commonsense benchmarks is *actually* commonsense, though they might not be shared locally in certain groups of people due to a lack of geographical diversity.

4.2 Contexts Identification

We now filter out commonsense statements without geographical or temporal contexts. Following (Zhang and Choi, 2021), we identify sentences with extra-linguistic contexts by GEO and TEMP entities. We use FLERT[5] (Schweter and Akbik, 2020), a named entity recognition (NER) model, to extract all entities from a sentence and remove those statements without any GEO ("GPE") or TEMP ("DATE", "TIME", "EVENT") entities.

Table 1 shows that of all the commonsense statements extracted from the five source datasets, 6.6% sentences have GEO contexts and 5.5% have TEMP contexts, which we count as a significant proportion. Finally, we obtain 4,038 (11.2%) commonsense statements with extra-linguistic contexts.

4.3 Contrastive Sentences Mining

We aim to automatically mine contrastive sentence pairs from the commonsense statement corpus. Antithesis mining has not been studied in the existing literature, so we propose a pilot algorithm. We observe that after removing keywords from contrastive sentences, the remaining parts are very similar, since antithesis sentences have parallel syntactic structures (Bridgwater, 1963). Based on this observation, we design the antithesis mining algorithm illustrated in Figure 2 consisting of three steps:

1. **Keyword Masking.** We extract all entities and other nouns as keywords in the sentence and replace each keyword with a `[UNK]` token, telling the pretrained language models to neglect the meaning of these keywords.

[5]https://huggingface.co/flair/ner-english-ontonotes-large

2. **Masked Sentence Similarity Matching.** We obtain the embedding of the keyword-masked sentence from a pretrained language model and calculate the cosine similarity between all possible sentence pairs.
3. **Rule-based Filtering.** We filter out invalid sentence pairs base on a fixed threshold of masked sentence similarity, number of keywords and entity types.

We introduce the implementation of our antithesis mining algorithm in Appendix A.2. In this way, we efficiently extracted 9,378 contrastive sentence pairs from all possible pairwise combinations of the previous 4,038 commonsense statements with extra-linguistic contexts[6] (Section 4.2). For each contrastive sentence pair, we merge the keywords from each statement and randomly shuffle them to get the input data. The output is the concatenation of two statements. When splitting the data into training, validation and test set, we explicitly require that one statement cannot appear simultaneously in any two sets. This ensures the compositional generalization challenge (Section 3.2) since there is no overlap among the sentence-level keyword combinations in the training, validation and test data. Statements with similar syntactic structures will also be divided into the same set to reduce overlap of syntactic templates across different sets[7]. To ensure the evaluation soundness, we manually filter out invalid examples in the *test* set that are not fluent antitheses or context dependent. 13.6% of test data are removed and the final dataset has 9,060 examples in total.

5 Dataset Analysis

5.1 Quality Analysis

To measure the quality of our automatically collected data, we randomly select 100 examples (i.e. sentence pairs) from the validation set (which is not manually filtered) and annotate each example for whether it is actually 1) (fluent) antithesis and 2) context dependent. We find that 87% of the data are real antitheses with fluency and 80% of the data satisfy both of the two requirements. Considering that our dataset is constructed through a fully automatic pipeline, this quality is pretty satisfying and can meet the needs of training and evaluation. As we have discussed in Section 3.1, test examples not satisfying contextual dependence can fool the evaluation metrics, since there are multiple valid references despite the single one provided in the test set. Thanks to the additional manual filtering, the test set is now qualified for evaluation. As for the unfiltered training set, even if a contrastive sentence pair is not context-dependent, it is still valuable training data, satisfying the other requirements for the target side (Section 3.1). A reduced size of training data after potential manual filtering is also unfavourable to the learning of models. As a result, we retain all the examples in the training set.

[6]One statement might be paired with multiple statements, formulating multiple contrastive sentence pairs.

[7]Please refer to Appendix A.3 for details of our dataset splitting algorithm.

Statistics	Train	Dev	Test
Size (# Sent Pairs)	5,641	1,407	2,012
# Unique Sents	788	309	449
per Sent Pair	0.14	**0.22**	**0.22**
# Unique Keywords	1,847	725	1,075
# Avg. Input Keywords	7.34	6.96	6.91
# Avg. Output Tokens	20.89	24.08	20.73

Table 2: The basic statistics of the SITUATEDGEN dataset. "Sent" means commonsense statement.

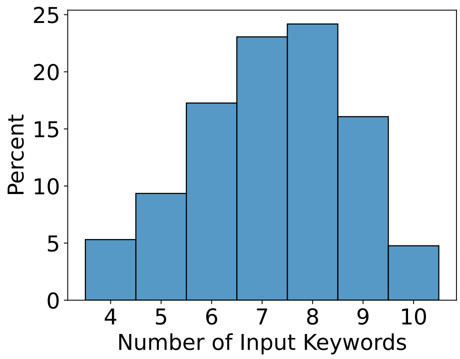

Figure 3: Distribution of numbers of input keywords.

Below, we analyze the bad cases in detail, including non-contrastive and non-context-dependent sentence pairs. The main reason for producing non-contrastive sentence pair is that the remaining verbs after keyword masking may have lexical ambiguity, e.g. "play" in "*Slaves* ***play*** *a role in the history of the united states.*" and "*A team sport* ***played*** *mostly in Canada is Lacrosse.*" Although the pretrained language models could infer the meaning of a word according to its context (Devlin et al., 2019), the contexts are lost after keyword masking. As a result, two sentences with different syntactic structures are matched together, thus violating the antithesis rule. This poses a limitation of our antithesis mining algorithm.

In addition, 7% of the sentence pairs are antitheses yet not context-dependent. Take the following sentence pair as an example: "*You could find millions of brownstone in New York City.*[8] *One can find a Holiday Inn inside the United States.*". After swapping the GEO entity "New York City" and "United States" in these two sentences, they still conform to commonsense. The reason for this phenomenon is that New York City is part of the United States, and thus the "brownstone" related to New York will also be related to the United States.

5.2 Dataset Statistics

Table 2 includes the basic statistics of the SITUATEDGEN dataset. If we use the ratio of unique statement count to sentence pair count ("# Unique Sents per Sent Pair") to represent the content/keyword diversity of the dataset, the validation set and the test set are relatively high (0.22), compared to the training set (0.14). This also shows that the test set is more challenging than the training set, which further increases the difficulty of the dataset.

Distribution of Numbers of Input Keywords. Figure 3 shows the distribution of numbers of input keywords for all examples in the dataset. More input keywords are more difficult for the models to handle. The average number of input keywords is 7.19 and the distribution is fairly symmetrical (skewness=-0.24), suggesting that the SITUATEDGEN has a reasonable difficulty.

Distribution of Context Types. We define three context types of pairs of contrastive sentences: a GEO pair of sentences contains only GEO entities; a TEMP pair of sentences contains only TEMP entities; If both sentences contain GEO and TEMP entities, the pair of sentences belongs to the type of GEO & TEMP . We find that 78% of all sentence pairs are GEO , 21% are TEMP and the rest 1% are GEO & TEMP .

6 Methods

Baseline Models. We benchmark the performance of two prominent pretrained language generation models: BART (Lewis et al., 2020) and T5 (Raffel et al., 2020). We fine-tuned all models on our training data with the seq2seq format and expect that the models can learn to group keywords

[8]As background knowledge, there are many historical buildings in New York City whose facades are made of brown sandstone, see https://bungalow.com/articles/what-exactly-is-a-brownstone.

Model	MATCH	BLEU-4	ROUGE-2	METEOR	CIDEr	SPICE
BART-base	61.2	22.7	29.6	29.9	18.2	53.9
BART-large	62.6	23.0	30.8	29.1	17.9	55.3
T5-base	56.5	22.1	28.8	30.1	17.4	54.1
T5-large	**67.7**	**26.3**	**33.1**	**31.9**	**20.9**	**57.9**
Human	92.1	41.5	48.2	40.5	40.1	72.0

Table 3: Experimental results on the test set of SITUATEDGEN . The best model performance is in **bold**. Human performance is tested on a subset of 100 random samples.

Context	MATCH	BLEU-4	ROUGE-2	METEOR	CIDEr	SPICE
GEO	**68.2**	25.5	31.9	31.7	20.0	57.3
TEMP	64.5	**31.1**	**42.2**	**33.8**	**24.0**	**62.5**
ALL	67.7	26.3	33.1	31.9	20.9	57.9

Table 4: The performance of **T5-large** across different context types on the test set of SITUATEDGEN . The best type performance is in **bold**.

implicitly. Specifically, for the input of BART, we concatenate all shuffled keywords with a comma as the separation token "$c_1, c_2, ..., c_k$". Regarding the input format of T5, we prepend the keyword sequence with a simple task description to align with its pretraining objective: "*generate two sentences with:* $c_1, c_2, ..., c_k$". The outputs of all models are simple concatenation of the two target sentences s_1 and s_2. Since the output is an unordered pair, we feed two examples "$x \rightarrow s_1\ s_2$" and "$x \rightarrow s_2\ s_1$" to the model for each original training example. We report the model hyper-parameters in Appendix B.1.

Evaluation Metrics. Lin et al. (2020) have well established the automatic evaluation protocol of the generative commonsense reasoning task. They demonstrated a strong correlation between the automatic metrics and human evaluation results. Since SITUATEDGEN adopts a similar format of keyword-to-text generation to COMMONGEN , we follow the evaluation protocol of COMMONGEN and do not include an extra manual evaluation in our study.

Concretely, we employ several widely-used automatic NLG metrics based on n-gram overlap — BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), METEOR (Banerjee and Lavie, 2005) — and image caption metrics that focus on the consistency of keywords and their relationships — CIDEr (Vedantam et al., 2015) and SPICE (Anderson et al., 2016). Additionally, we report the accuracy of keyword

Input Keywords	24 hours, axis, one month, Earth, axis, Moon
Reference	It takes for the Moon to rotate on its axis one month. Earth rotating on its axis takes 24 hours.
BART-base	The axis of the Moon is 24 hours. One month is one month.
BART-large	There are 24 hours in one month.
T5-base	Earth has a 24 hour axis. One month is one month.
T5-large	One month is one month on Earth. The Moon is 24 hours away from the axis of the Earth.
Input Keywords	Paul, Emperor, China, Qin, Russia, dynasty
Reference	The Qin dynasty reigned in China. Paul I of Russia reigned as the Emperor of Russia.
BART-base	The Emperor of China worked in China. Paul served as the first emperor of the dynasty Qin.
BART-large	Emperor of the Qin dynasty. Paul existed in Russia.
T5-base	China is a dynasty of China. Paul Qin is the Emperor of China.
T5-large	Paul was the Emperor of Russia. The Qin dynasty ruled China.

Table 5: Case studies of machine generations. Keywords appearing in the generation results are underlined.

grouping results[9] as `MATCH`, which serves as a good indicator of the commonsense plausibility of the generated texts. In particular, if a keyword does not appear in the output, we treat it as unmatched. In this way, `MATCH` also reflects the coverage of keywords in the output. See Appendix B.2 for the implementation details of these evaluation metrics.

7 Results

In Table 3, we report the experimental results of different baseline models on the test set of SituatedGen. We approximate human performance with 100 randomly sampled examples from the test set which are annotated by the authors of this paper. We observe that larger models tend to have better performance than smaller ones. The biggest tested model, T5-large, surpasses other models in every metric, but it still lags far behind human performance. For example, there is a difference of about 24 points in `MATCH`, indicating the lack of commonsense in machine generations. The large gap of keyword-based metrics (`CIDEr` and `SPICE`) also suggests that models find it difficult to infer the relationship between keywords. Furthermore, machine-generated outputs are considered less fluent by n-gram-based metrics (`BLEU`, `ROUGE` and `METEOR`). The significant gap between models and humans demonstrates the difficulty of SituatedGen and leaves much room for improvement in future research.

Performance across Different Context Types. Table 4 reports the performance of the T5-large model across different context types. The results show that the matching accuracy of TEMP type is lower than GEO, indicating that temporal-dependent test examples are more challenging. However, the amount of TEMP data is less than GEO in the training set, which may also give rise to the performance difference. Interestingly, the generation fluency of GEO type is worse than TEMP, suggesting that it is more difficult to use GEO entities to compose sentences smoothly.

[9]Keywords appearing in the same lemmatized output sentence are considered to be grouped together by models.

Case Study. Table 5 shows two groups of generation examples by different models. The first example belongs to TEMP type (“24 hours” and “one month”) and the second one is GEO (“Russia” and “China”). We find that models are prone to omit keywords in their outputs. For example, BART-large only covers 2 out of 6 keywords in the first example. Besides, most of the observed generated outputs are not commonsensical due to wrong keyword grouping results, e.g., “*There are 24 hours in one month*” and “*Paul served as the first emperor of the dynasty Qin*”. Surprisingly, the generation result of T5-large in the second example is quite close to the gold reference.

8 Conclusion

In this paper, we introduce the challenging task SituatedGen to incorporate geographical and temporal contexts into generative commonsense reasoning. We build a corresponding testbed to evaluate the situated reasoning capabilities of state-of-the-art text generation models. The benchmark performance shows that models struggle to generate commonsensical sentences and lag far behind human on our proposed task. Altogether, our data will serve as a challenging benchmark for measuring commonsense knowledge in language generation models and support future progress of constrained commonsense text generation in a more realistic situation.

Ethics Statement

Our data is built upon publicly available datasets and we will follow their licenses when releasing our data. There is no explicit detail that leaks an annotator's personal information. The dataset has very low risks of containing sentences with toxicity and offensiveness. Since our data is sourced from existing datasets, we may inherit geographical biases (Faisal et al., 2022) that result in an uneven distribution of commonsense knowledge about western and non-western regions. The commonsense statements may not sound familiar to people who live in locations that are poorly represented in the source datasets. Therefore, models developed on our dataset may preserve biases learned from the annotators of the source datasets. We note that pretrained language models may also inherit the bias in the massive pretraining data. It is important that interested parties carefully address those biases before deploying the model to real-world settings.

References

James F. Allen. 1983. Maintaining knowledge about temporal intervals. *Commun. ACM*, 26(11):832–843.

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. SPICE: semantic propositional image caption evaluation. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V*, volume 9909 of *Lecture Notes in Computer Science*, pages 382–398. Springer.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: an automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization@ACL 2005, Ann Arbor, Michigan, USA, June 29, 2005*, pages 65–72. Association for Computational Linguistics.

Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Wen-tau Yih, and Yejin Choi. 2020. Abductive commonsense reasoning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Prajjwal Bhargava and Vincent Ng. 2022. Commonsense knowledge reasoning and generation with pre-trained language models: A survey. *CoRR*, abs/2201.12438.

Mehul Bhatt and Jan Oliver Wallgrün. 2014. Geospatial narratives and their spatio-temporal dynamics: Commonsense reasoning for high-level analyses in geographic information systems. *ISPRS Int. J. Geo Inf.*, 3(1):166–205.

Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008.

William Bridgwater. 1963. The columbia encyclopedia. Technical report.

Wenhu Chen, Xinyi Wang, and William Yang Wang. 2021. A dataset for answering time-sensitive questions. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the AI2 reasoning challenge. *CoRR*, abs/1803.05457.

Dorottya Demszky, Kelvin Guu, and Percy Liang. 2018. Transforming question answering datasets into natural language inference datasets. *CoRR*, abs/1809.02922.

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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Bhuwan Dhingra, Jeremy R. Cole, Julian Martin Eisenschlos, Daniel Gillick, Jacob Eisenstein, and William W. Cohen. 2021. Time-aware language models as temporal knowledge bases. *CoRR*, abs/2106.15110.

Fahim Faisal, Yinkai Wang, and Antonios Anastasopoulos. 2022. Dataset geography: Mapping language data to language users. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 3381–3411. Association for Computational Linguistics.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? A question answering benchmark with implicit reasoning strategies. *Trans. Assoc. Comput. Linguistics*, 9:346–361.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R. Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018*

Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers), pages 107–112. Association for Computational Linguistics.

Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Cosmos QA: machine reading comprehension with contextual commonsense reasoning. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 2391–2401. Association for Computational Linguistics.

Daniel Keysers, Nathanael Schärli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, Dmitry Tsarkov, Xiao Wang, Marc van Zee, and Olivier Bousquet. 2020. Measuring compositional generalization: A comprehensive method on realistic data. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. Association for Computational Linguistics.

Haonan Li, Yeyun Gong, Jian Jiao, Ruofei Zhang, Timothy Baldwin, and Nan Duan. 2021. KFCNet: Knowledge filtering and contrastive learning for generative commonsense reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2918–2928, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Bill Yuchen Lin, Ziyi Wu, Yichi Yang, Dong-Ho Lee, and Xiang Ren. 2021. Riddlesense: Reasoning about riddle questions featuring linguistic creativity and commonsense knowledge. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 1504–1515. Association for Computational Linguistics.

Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. 2020. Commongen: A constrained text generation challenge for generative commonsense reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 1823–1840. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Fangyu Liu, Emanuele Bugliarello, Edoardo Maria Ponti, Siva Reddy, Nigel Collier, and Desmond Elliott. 2021a. Visually grounded reasoning across languages and cultures. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 10467–10485. Association for Computational Linguistics.

Xin Liu, Dayiheng Liu, Baosong Yang, Haibo Zhang, Junwei Ding, Wenqing Yao, Weihua Luo, Haiying Zhang, and Jinsong Su. 2021b. Kgr^4: Retrieval, retrospect, refine and rethink for commonsense generation. *CoRR*, abs/2112.08266.

Alfred H Lloyd. 1911. The logic of antithesis. *The Journal of Philosophy, Psychology and Scientific Methods*, 8(11):281–289.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? A new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2381–2391. Association for Computational Linguistics.

Lili Mou, Yiping Song, Rui Yan, Ge Li, Lu Zhang, and Zhi Jin. 2016. Sequence to backward and forward sequences: A content-introducing approach to generative short-text conversation. In *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, pages 3349–3358. ACL.

Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. 2016. A review of relational machine learning for knowledge graphs. *Proc. IEEE*, 104(1):11–33.

Qiang Ning, Hao Wu, Rujun Han, Nanyun Peng, Matt Gardner, and Dan Roth. 2020. TORQUE: A reading comprehension dataset of temporal ordering questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 1158–1172. Association for Computational Linguistics.

Yasumasa Onoe, Michael J. Q. Zhang, Eunsol Choi, and Greg Durrett. 2021. CREAK: A dataset for commonsense reasoning over entity knowledge. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.

Liangming Pan, Wenhu Chen, Wenhan Xiong, Min-Yen Kan, and William Yang Wang. 2021. Zero-shot fact verification by claim generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 2: Short Papers), Virtual Event, August 1-6, 2021*, pages 476–483. 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, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318. ACL.

Jungsoo Park, Sewon Min, Jaewoo Kang, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2022. Faviq: Fact verification from information-seeking questions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 5154–5166. Association for Computational Linguistics.

Lianhui Qin, Aditya Gupta, Shyam Upadhyay, Luheng He, Yejin Choi, and Manaal Faruqui. 2021. TIME-DIAL: temporal commonsense reasoning in dialog. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 7066–7076. Association for Computational Linguistics.

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. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 3980–3990. Association for Computational Linguistics.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019. Social iqa: Commonsense reasoning about social interactions. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 4462–4472. Association for Computational Linguistics.

Stefan Schweter and Alan Akbik. 2020. FLERT: document-level features for named entity recognition. *CoRR*, abs/2011.06993.

Lei Sha. 2020. Gradient-guided unsupervised lexically constrained text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 8692–8703. Association for Computational Linguistics.

Shikhar Singh, Nuan Wen, Yu Hou, Pegah Alipoormolabashi, Te-Lin Wu, Xuezhe Ma, and Nanyun Peng. 2021. COM2SENSE: A commonsense reasoning benchmark with complementary sentences. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 883–898. Association for Computational Linguistics.

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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4149–4158. Association for Computational Linguistics.

Lifu Tu, Garima Lalwani, Spandana Gella, and He He. 2020. An empirical study on robustness to spurious correlations using pre-trained language models. *Trans. Assoc. Comput. Linguistics*, 8:621–633.

Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 4566–4575. IEEE Computer Society.

Peifeng Wang, Jonathan Zamora, Junfeng Liu, Filip Ilievski, Muhao Chen, and Xiang Ren. 2021. Contextualized scene imagination for generative commonsense reasoning. *CoRR*, abs/2112.06318.

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.

Lili Yao, Nanyun Peng, Ralph M. Weischedel, Kevin Knight, Dongyan Zhao, and Rui Yan. 2019. Plan-and-write: Towards better automatic storytelling. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational*

Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019, pages 7378–7385. AAAI Press.

Da Yin, Hritik Bansal, Masoud Monajatipoor, Liunian Harold Li, and Kai-Wei Chang. 2022. Geomlama: Geo-diverse commonsense probing on multilingual pre-trained language models. *CoRR*, abs/2205.12247.

Da Yin, Liunian Harold Li, Ziniu Hu, Nanyun Peng, and Kai-Wei Chang. 2021. Broaden the vision: Geo-diverse visual commonsense reasoning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 2115–2129. Association for Computational Linguistics.

Michael J. Q. Zhang and Eunsol Choi. 2021. Situatedqa: Incorporating extra-linguistic contexts into QA. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 7371–7387. Association for Computational Linguistics.

Ben Zhou, Daniel Khashabi, Qiang Ning, and Dan Roth. 2019. "going on a vacation" takes longer than "going for a walk": A study of temporal commonsense understanding. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 3361–3367. Association for Computational Linguistics.

A Additional Details of Dataset Collection

A.1 Commonsense Statement Collection

We briefly introduce the nature of each source datasets in Section 4.1.

- **CREAK** (Onoe et al., 2021) is a commonsense fact verification dataset featuring entity commonsense, which includes 13,418 true or false statements about entity knowledge written by crowdworkers.
- **StrategyQA** (Geva et al., 2021) is a commonsense question answering dataset that requires multi-hop implicit reasoning. It consists of 5,111 questions whose answers are either Yes or No. Machines need to decompose a question into multiple atomic questions to arrive at an answer.
- **CommonsenseQA** (Talmor et al., 2019) is a commonsense question answering dataset of 12,247 five-way multiple-choice questions with a focus on knowledge in everyday life.
- **ARC** (Clark et al., 2018) is a commonsense question answering dataset. It has 7,787 four-way multiple-choice natural science questions collected from grade-school standardized tests.
- **OpenbookQA** (Mihaylov et al., 2018) is a commonsense question answering dataset that simulates openbook test. The data set is made up of 5,957 multiple-choice questions, accompanied by 6,493 commonsense statements about science facts. Since there is a significant overlap between the knowledge in questions and statements, we only use the statements data for simplicity.

We now detail the specific preprocessing method for each source dataset to convert them (i.e., question-answer pairs) into statements.

- If the raw data comes in the statement format (CREAK and OpenbookQA), we obtain the true statements (part of CREAK and all of OpenbookQA) without extra processing.
- If the raw data comes in Yes/No question format (StrategyQA), we leverage a POS-rule-based open-sourced system `question_to_statement`[10] to transform a pair of question and Yes/No answer into a statement.

[10] `https://github.com/SunnyWay/question_to_statement`

Dataset	Size	Format	Raw Data → Statement Conversion Example
CREAK (Onoe et al., 2021)	13,418	True/False statement	In the calendar year, May comes after April and before June. (**True**/False) → In the calendar year, May comes after April and before June.
StrategyQA (Geva et al., 2021)	5,111	Yes/No Question	Are more watermelons grown in Texas than in Antarctica? (**Yes**/No) → More watermelons are grown in Texas than in Antarctica.
CommonsenseQA (Talmor et al., 2019)	12,247	Multiple-choice Question	Where in Southern Europe would you find many canals? (A) Michigan (B) New York (C) Amsterdam **(D) Venice** (E) Sydney → You would find many cannals in Venice, Southern Europe.
ARC (Clark et al., 2018)	7,787	Multiple-choice Question	How long does it take for Earth to rotate on its axis seven times? (A) one day **(B) one week** (C) one month (D) one year → It takes one week for Earth to rotate on its axis seven times.
OpenbookQA (Mihaylov et al., 2018)	6,493	Commonsense Statement	You wear shorts in the summer. → You wear shorts in the summer.

Table 6: Source dataset examples. **Correct answers** are in bold and underlined.

- If the raw data comes in multiple-choice format (CommonsenseQA and ARC), we utilize a neural model to convert a pair of question and correct choice (q, a) into a statement in a sequence-to-sequence fashion. Concretely, we use the QA-to-statement model checkpoint released by Pan et al. (2021), which is a BART (Lewis et al., 2020) model finetuned on QA2D (Demszky et al., 2018), a dataset of human-annotated statements for QA pairs.

We summarize the basic information of these datasets and provide an example of statement conversion for each dataset in Table 6.

A.2 Antithesis Mining

Keyword Masking. We use entities and other nouns as the keywords of sentences, because as a pilot study, we only consider the relationships between spatio-temporal contexts and nouns and ignore the influence of other part of speech categories such as verbs, adjectives and prepositions. We use the same NER tagger in Section 4.2 to extract entities. We leverage spaCy[11] to extract all the nouns (including proper nouns) from a sentence. We merge the entities and nouns as keywords after removing duplicates. In particular, if a noun and an entity partly overlaps (e.g., "**month**" and "a lunar **month**"), we retain the entity when deduplicating.

Masked Sentence Similarity Matching. We use the pretrained language model `all-MiniLM-L6-v2`[12] released by Sentence-Transformers (Reimers and Gurevych, 2019) to obtain high-quality embeddings of keyword-masked sentences. We calculate the cosine similarity to pair highly similar masked sentences. Computing the similarity of all possible sentence pairs requires $\mathcal{O}(n^2)$ time complexity. To accelerate this process, we use the `paraphrase_mining` API of SentenceTransformers (Reimers and Gurevych, 2019).

Rule-based Filtering. We devise the following rules to filter invalid sentence pairs based on iterative observation of the data:

- The masked sentence similarity exceeds a certain threshold[13], which indicates parallel sentence structure of antithesis.
- The number of masked keywords (`[UNK]`) of each single sentence should not be more than 5 and less than 2, which controls for a reason-

[11] `https://spacy.io/models/en#en_core_web_sm`

[12] `https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2`

[13] We set the threshold as 0.8 via manual inspection.

able difficulty of the keyword-to-text generation task.

- Any entity in one sentence does not appear in the other sentence within a pair (including the deformation of entity words, such as singular/plural form, upper/lower case, etc.). This is to avoid that both sentences express the information of the same entity, while the contrastive sentences should describe two opposite things.
- Both of the two sentences contain either GEO entities or TEMP entities (GEO+GEO or TEMP+TEMP), which avoids sentences comparing GEO context to a non-parallel TEMP context (GEO+TEMP).

A.3 Dataset Splitting

We treat dataset splitting as a community structure (Blondel et al., 2008) discovery problem. Community structure refers to a group of tightly connected nodes that have a high density of internal connections and a low density of external connections. We regard a single sentence as a node in the graph. If two single sentences can be matched into a pair of contrastive sentences, an undirected edge will connect the corresponding nodes of these two single sentences. In this way, we obtain an undirected graph describing the dataset structure. A subset of a dataset (such as a training set) is equivalent to a subgraph containing all sentence pairs (edges) and single sentences (nodes) of that subset.

In order to prevent the same sentence from appearing across different sets, we require that the subgraph node sets of the training set, validation set, and test set are disjoint. We use a community structure detection algorithm to meet this requirement. We use the community as the basic unit of dataset splitting, putting all the edges (sentence pairs) in one community into a certain dataset split. Connecting edges between communities (two vertices belong to different community) are removed. We note that sentences with similar syntactic structures tend to be connected to each other in the graph and thus fall into the same community, which ensures the syntactic variability between train/dev/test splits.

We use the Louvain (Blondel et al., 2008) community structure detection algorithm[14] and divide our graph into 79 communities. The largest community contains 3,273 edges, accounting for about 26% of the total data. After removing a total of 3,311 edges connecting different communities (about 26% of the total), we obtained 9,378 contrastive sentence pairs with geographical or temporal contexts. We then randomly divide the communities into training set, validation set or test set.

[14] `https://github.com/shobrook/communities`

Parameter	Value
epoch	10
batch size	32
beam size	4
max input length	64
max output length	128
learning rate	3e-5
warm-up steps	500

Table 7: Hyper-parameter settings for all baseline models.

B Experimental Setup

B.1 Baseline Models

We use HuggingFace (Wolf et al., 2020) implementations of the BART and T5 models. For decoding method, we adopt the standard beam search with a beam size of 4 for all baseline models. As for checkpoint selection, we save checkpoint for each epoch and select the checkpoint with highest `ROUGE-2` on the validation set. Other default hyper-parameters are shown in Table 7.

B.2 Evaluation Metrics

We use the standard implementation of BLEU, ROUGE, METEOR, CIDEr, SPICE in `pycocoevalcap`[15]. In addition, we design and implement `MATCH` to evaluate how well the machines solve the challenge of situated semantic matching (Section 3.2). We now define the keyword matching accuracy `MATCH` based on mathematical notations introduced in Section 3.1.

$t = (t_1, ..., t_k), t_i \in \{0, 1\}$ indicates that each keyword c_i appears in which sentence in the answer pair $y^{true} = \{s_1^{true}, s_2^{true}\}$. In other words, if c_i *should* appear in s_1, then $t_i = 0$; if c_i *should* appear in s_2, then $t_i = 1$. $p = (p_1, ..., p_k), p_i \in \{-1, 0, 1\}$ indicates that each keyword c_i appears in which sentence in the output pair $y^{pred} = \{s_1^{pred}, s_2^{pred}\}$. In other words, if c_i *actually* appear in s_1, then $p_i = 0$; if c_i *actually* appear in s_2, then $p_i = 1$; if c_i does

[15] `https://github.com/salaniz/pycocoevalcap`

not *actually* appear in both s_1 and s_2, then $p_i = -1$[16]. We define the matching accutacy of a sentence pair match(y^{true}, y^{pred}) as the proportion of correctly matched keywords, which is calculated as $\frac{1}{k}\max(\sum_{i=1}^{k}\mathbb{1}_{t_i=p_i}, \sum_{i=1}^{k}\mathbb{1}_{1-t_i=p_i}) \in [0,1]$. Here $\mathbb{1}$ is the indicator function. The formula includes both $1-t$ and t in a symmetric way because the sentence pair is unordered. For the whole test set, we take the average matching accuracy of all examples as `MATCH`.

We illustrate the computing process of matching accuracy with a simple example. Given `[July, China, winter, Australia, summer, July]`, the answer could be "*July is summer in China. July is winter in Australia.*" So $t = (0,0,1,1,0,1)$. If the generated output is "*July is summer in Australia. July is winter in China.*", then $p = (0,1,1,0,0,1)$. As a result, the matching accuracy is $4/6 = 0.67$.

As for the implementation, we utilize `NLTK`[17] to split the output into two sentences. In particular, if there is only one sentence in the output, we append an empty string as the second one; if there are more than two sentences, we only take the former two sentences into consideration. We lemmatize the sentence before determining keyword appearance.

[16] By defining $p_i = -1$, `MATCH` can also reflect the coverage of keywords in the output.

[17] `https://www.nltk.org/`