A Datasheet for SITUATEDGEN

A.1 Motivation

1. For what purpose was the dataset created? Was there a specific task in mind? Was there a specific gap that needed to be filled? Please provide a description.

This dataset aims to probe the commonsense reasoning ability of generative language models through the lens of keyword generation tasks. The task requires machines to compose a pair of contrastive sentences with a given set of keywords containing geographical or temporal entities. Current models lack the ability to correctly reason for the relationship among these entities and thus generate sentences that contradict commonsense knowledge. We hope our dataset could stir more research to fill this gap of generative commonsense reasoning.

2. Who created the dataset (e.g., which team, research group) and on behalf of which entity (e.g., company, institution, organization)?

The dataset is created by Yunxiang Zhang and Xiaojun Wan on behalf of the Text Mining and Linguistic Computing Group, Wangxuan Institute of Computer Technology, Peking University. Most part of this paper is done when the first author is at Peking University before moving to University of Michigan.

3. Who funded the creation of the dataset? If there is an associated grant, please provide the name of the grantor and the grant name and number.

It is funded by the Text Mining and Linguistic Computing Group, Wangxuan Institute of Computer Technology, Peking University.

A.2 Composition

1. What do the instances that comprise the dataset represent (e.g., documents, photos, people, countries)? Are there multiple types of instances (e.g., movies, users, and ratings; people and interactions be tween them; nodes and edges)? Please provide a description.

The dataset is comprised of pure text data in English, presented in a Jsonline format. The file is composed of a list of instances containing input keywords and targeted outputs.

2. How many instances are there in total (of each type, if appropriate)?

Our dataset consists of 8,268 instances. Please refer to Table 2 for detailed information.

3. Does the dataset contain all possible instances or is it a sample (not necessarily random) of instances from a larger set? If the dataset is a sample, then what is the larger set? Is the sample representative of the larger set (e.g., geographic coverage)? If so, please describe how this representativeness was validated/verified. If it is not representative of the larger set, please describe why not (e.g., to cover a more diverse range of instances, because instances were withheld or unavailable).

This dataset does not cover all aspects of commonsense knowledge so it does not contain all possible instances. We focus on geographical and temporal commonsense in this work since they provide testbeds for evaluating machines' reasoning ability under different extra-linguistic contexts.

4. What data does each instance consist of? "Raw" data (e.g., unprocessed text or images) or features? In either case, please provide a description.

Each instance is a dictionary has the following fields:

- "keywords": a list of keywords as input
- "statement": a string concatenation of two sentences as the target generations

- "ids": the origins of the commonsense statement (from which (train/dev/test) split of which source datasets/corpora) represented in the format of "{src_dataset}::{split}::{id}"
- "statements": a list of the two sentences in "statement" field.

5. Is there a label or target associated with each instance? If so, please provide a description.

Yes. It is represented as the "statement" filed in each instance.

6. Is any information missing from individual instances? If so, please provide a description, explaining why this information is missing (e.g., because it was unavailable). This does not include intentionally removed information, but might include, e.g., redacted text.

No. All instances are complete.

7. Are relationships between individual instances made explicit (e.g., users' movie ratings, social network links)? If so, please describe how these relationships are made explicit.

Individual instances are independent of each other. The train/dev/test splits do not overlap in any single sentence.

8. Are there recommended data splits (e.g., training, development/validation, testing)? If so, please provide a description of these splits, explaining the rationale behind them.

Yes, see Tabel 2 for details. The splitting process makes sure that the train/dev/test splits do not overlap in any single sentence. See Appendix D.3 for details.

9. Are there any errors, sources of noise, or redundancies in the dataset? If so, please provide a description.

There is noise in the train and dev set. We manually filter out unqualified examples in the test set. See more analysis in Section 5.1 and Appendix E.2.

10. Is the dataset self-contained, or does it link to or otherwise rely on external resources (e.g., websites, tweets, other datasets)? If it links to or relies on external resources, a) are there guarantees that they will exist, and remain constant, over time; b) are there official archival versions of the complete dataset (i.e., including the external resources as they existed at the time the dataset was created); c) are there any restrictions (e.g., licenses, fees) associated with any of the external resources that might apply to a dataset consumer? Please provide descriptions of all external resources and any restrictions associated with them, as well as links or other access points, as appropriate.

The SITUATEDGEN dataset is self-contained and we welcome practitioners to consider additional knowledge sources.

11. Does the dataset contain data that might be considered confidential (e.g., data that is protected by legal privilege or by doctor–patient confidentiality, data that includes the content of individuals' non-public communications)? If so, please provide a description.

No.

12. Does the dataset contain data that, if viewed directly, might be offensive, insulting, threatening, or might otherwise cause anxiety? If so, please describe why.

No.

A.3 Collection Process

1. How was the data associated with each instance acquired? Was the data directly observable (e.g., raw text, movie ratings), reported by subjects (e.g., survey responses), or indirectly inferred/derived from other data (e.g., part-of-speech tags, model-based guesses for age or language)? If the data was reported by subjects or indirectly inferred/derived from other data, was the data validated/verified? If so, please describe how.

The data is sourced from several commonsense related datasets and corpora. We design an automatic pipeline to convert and filter data into our desired format.

2. What mechanisms or procedures were used to collect the data (e.g., hardware apparatuses or sensors, manual human curation, software programs, software APIs)? How were these mechanisms or procedures validated?

We first convert instances from other datasets as commonsense statements. Then we match these statements into pairs and extract keywords from them. We further manually filter out invalid examples in the test set.

3. Who was involved in the data collection process (e.g., students, crowdworkers, contractors) and how were they compensated (e.g., how much were crowdworkers paid)?

We hired crowdworkers and compensated them with 0.1 yuan for each entry they checked, which is higher than the statutory minimum wage.

4. Over what timeframe was the data collected? Does this timeframe match the creation timeframe of the data associated with the instances (e.g., recent crawl of old news articles)? If not, please describe the time frame in which the data associated with the instances was created.

Our dataset was built in 2022 while the original source data is published between 2018-2021. Usually, commonsense statements are not changing over time.

5. Were any ethical review processes conducted (e.g., by an institutional review board)? If so, please provide a description of these review processes, including the outcomes, as well as a link or other access point to any supporting documentation.

No.

A.4 Preprocessing/cleaning/labeling

1. Was any preprocessing/cleaning/labeling of the data done (e.g., discretization or bucketing, tokenization, part-of-speech tagging, SIFT feature extraction, removal of instances, processing of missing values)? If so, please provide a description. If not, you may skip the remaining questions in this section.

Yes. We use templated-based and neural-based models to convert and filter the source data into our desired format. See details in Appendix D.

2. Was the "raw" data saved in addition to the preprocessed/cleaned/labeled data (e.g., to support unanticipated future uses)? If so, please provide a link or other access point to the "raw" data. Yes. The raw data is available on the corresponding dataset websites (CREAK – `https://github.com/yasumasaonoe/creak`, OpenbookQA – `https://allenai.org/data/open-book-qa`, StrategyQA – `https://allenai.org/data/strategyqa`, CommonsenseQA – `https://www.tau-nlp.sites.tau.ac.il/commonsenseqa`, ARC – `https://allenai.org/data/arc`).

3. Is the software that was used to preprocess/clean/label the data available? If so, please provide a link or other access point.

Yes. Please see `https://github.com/yunx-z/situated_gen`.

A.5 Uses

1. Has the dataset been used for any tasks already? If so, please provide a description.

Not yet.

2. Is there a repository that links to any or all papers or systems that use the dataset? If so, please provide a link or other access point.

There has not been such a paper or system yet.

3. What (other) tasks could the dataset be used for?

It can be used to develop better language models for commonsense reasoning. It can be used to evaluate language models, especially their understanding of commonsense knowledge. It could potentially benefit many downstream applications such as document summarization [44], story writing [51] and dialogue response generation [31].

4. Is there anything about the composition of the dataset or the way it was collected and preprocessed/cleaned/labeled that might impact future uses? For example, is there anything that a dataset consumer might need to know to avoid uses that could result in unfair treatment of individuals or groups (e.g., stereotyping, quality of service issues) or other risks or harms (e.g., legal risks, financial harms)? If so, please provide a description. Is there anything a dataset consumer could do to mitigate these risks or harms?

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 [16] 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.

5. Are there tasks for which the dataset should not be used? If so, please provide a description.

The dataset can only be used for research purposes.

A.6 Distribution

1. Will the dataset be distributed to third parties outside of the entity (e.g., company, institution, organization) on behalf of which the dataset was created? If so, please provide a description.

The dataset is already publicly available.

2. How will the dataset will be distributed (e.g., tarball on website, API, GitHub)? Does the dataset have a digital object identifier (DOI)?

The dataset is available at `https://github.com/yunx-z/situated_gen`.

3. When will the dataset be distributed?

It has already been distributed.

4. Will the dataset be distributed under a copyright or other intellectual property (IP) license, and/or under applicable terms of use(ToU)? If so, please describe this license and/or ToU, and provide a link or other access point to, or otherwise reproduce, any relevant licensing terms or ToU, as well as any fees associated with these restrictions.

This dataset is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License (CC BY-NC-SA 4.0). The full text of the license can be accessed at the following link: `https://creativecommons.org/licenses/by-nc-sa/4.0/`.

5. Have any third parties imposed IP-based or other restrictions on the data associated with the instances? If so, please describe these restrictions, and provide a link or other access point to, or otherwise reproduce, any relevant licensing terms, as well as any fees associated with these restrictions.

No.

6. Do any export controls or other regulatory restrictions apply to the dataset or to individual instances? If so, please describe these restrictions, and provide a link or other access point to, or otherwise reproduce, any supporting documentation.

No.

A.7 Maintenance

1. Who will be supporting/hosting/maintaining the dataset?

The first author, Yunxiang Zhang, is hosting and maintaining the dataset.

2. How can the owner/curator/manager of the dataset be contacted (e.g., email address)?

Email: yunxiang@umich.edu

3. Is there an erratum? If so, please provide a link or other access point.

No.

4. Will the dataset be updated (e.g., to correct labeling errors, add new instances, delete instances)? If so, please describe how often, by whom, and how updates will be communicated to dataset consumers (e.g., mailing list, GitHub)?

We are interested to collect more data using our automatic pipelines and conduct manual filtering as future work. We also welcome interested parties to point out errors in the dataset via contact email or github issues so we could correct them. If there is a plan for systematic updates, we will announce it at the earliest opportunity.

5. If others want to extend/augment/build on/contribute to the dataset, is there a mechanism for them to do so? If so, please provide a description. Will these contributions be validated/verified? If so, please describe how. If not, why not? Is there a process for communicating/distributing these contributions to dataset consumers? If so, please provide a description.

People can use this repository following the licenses and cite our paper.

B Limitations

Since our dataset is derived from existing commonsense benchmarks, we may inherit their annotation artifacts [18] and contain certain types of spurious lexical patterns (e.g., "A lived in B"). We could also conduct an extra manual evaluation on the machine generations, so as to gauge its correlation with automatic metrics, though this has been verified by [25] on the original generative commonsense reasoning task. Recently, a lot of work has developed new retrieval-augmented commonsense text generation models [54, 19], which could also be included as baseline models for a more comprehensive benchmark.

C 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 [16] 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.

Table 5: Source dataset examples. **Correct answers** are in bold and underlined.

Dataset	Size	Format	Raw Data → Statement Conversion Example
CREAK [34]	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 [17]	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 [46]	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 canals in Venice, Southern Europe.
ARC [11]	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 [30]	6,493	Commonsense Statement	You wear shorts in the summer. → You wear shorts in the summer.

D Additional Details of Dataset Collection

D.1 Commonsense Statement Collection

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

- **CREAK** [34] is a commonsense fact verification dataset featuring entity commonsense, which includes 13,418 true or false statements about entity knowledge written by crowdworkers.
- **StrategyQA** [17] 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** [46] is a commonsense question answering dataset of 12,247 five-way multiple-choice questions with a focus on knowledge in everyday life.
- **ARC** [11] is a commonsense question answering dataset. It has 7,787 four-way multiple-choice natural science questions collected from grade-school standardized tests.
- **OpenbookQA** [30] 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`[8] to transform a pair of question and Yes/No answer into a statement.

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

- 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 [36], which is a BART [22] model finetuned on QA2D [13], a dataset of human-annotated statements for QA pairs.

Converting QA pair to statement is not a difficult task for pretrained seq2seq models. We observe that the generated statements are mostly fluent and faithful to the input. Additionally, we have manually filtered out unnatural examples in the test set. We summarize the basic information of these datasets and provide an example of statement conversion for each dataset in Table 5.

D.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[9] 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 overlap (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`[10] released by SentenceTransformers [41] 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 [41].

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[11], which indicates parallel sentence structure of antithesis.
- The number of masked keywords (`[UNK]`) of every single sentence should not be more than 5 and less than 2, which controls for a reasonable 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 both sentences expressing the information of the same entity, while 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).

D.3 Dataset Splitting

We treat dataset splitting as a community structure [7] 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

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

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

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

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 belonging to different communities) 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 [7] community structure detection algorithm[12] and divide our graph into 79 communities. The largest community contains 3,273 edges, accounting for about 26% of the total data. We remove edges connecting different communities and then randomly divide the communities of contrastive sentence pairs into training set, validation set or test set.

E Dataset Quality Analysis

E.1 Manual Filtering of the Test Set

To ensure the high quality of the dataset, we manually filter out invalid examples in the test set that are not fluent antitheses or context-dependent. This process is important for the very high human performance shown in Table 3. Table 6 shows the instructions for annotators. We first ask two graduate students with proficiency in English to annotate 100 examples as valid or invalid. They agree with each other (i.e., give the same label) on 88% of examples. The inter-annotator agreement in terms of Cohen's Kappa [12] is 0.76, which indicates substantial agreement [21]. Since the agreement ratio is satisfactory, we ask one of the annotators to complete the rest of the filtering process.

E.2 Error Cases Analysis

In Section 5.1, we annotate 100 random examples for whether it is actually 1) (fluent) antithesis and 2) context-dependent. Below, we analyze the bad cases in detail, including non-contrastive and non-context-dependent sentence pairs.

The main explanation that accounts for the production of 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 [14], 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.*[13] *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. However, we would like to point out that contextual dependence is not an absolutely strict condition. Although this example still holds after swapping the GEO entities, it is not the optimal answer, because "brownstone" is more a typical thing in New York City and thus more suitable for a match with "New York City".

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

[13] 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`.

Table 6: Annotator instructions for manual filtering of our dataset.

Goal: The objective of our project is to generate high-quality contrastive sentence pairs (antithesis) that incorporate geographical and temporal contexts. These sentence pairs will serve as a means to evaluate machines' commonsense reasoning abilities under different extra-linguistic contexts. We aim to create sentences that require a deep understanding of real-world geographical and temporal entities but can be reasonably confirmed without resorting to external sources like Google or Wikipedia.

Instructions: We show a set of keywords and a pair of sentences containing these keywords. Your task is to determine whether this sentence pair satisfies *all* of the following criteria:

1. The sentence pair includes all of the given keywords.
2. Each sentence has at least one entity related to geography or time.
3. Each sentence is fluent and adheres to commonsense knowledge.
4. The two sentences have similar syntactic structures and create a contradiction in semantics.
    - Intuitively, the qualifying two sentences can be connected into a coherent sentence via a conjunction word such as "while", "yet", and "whereas" (e.g., "*July is summer in the United States, while July is winter in Australia.*").
5. Swapping any of the geographical or temporal entities between the two sentences could lead to a 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 geographical 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.*"

Examples:
Keywords: *morning, night, sunrise, sunset*
Sentence 1: "The sky is bright with the sunrise in the early morning."
Sentence 2: "The sky is dark with the sunset in the late night."

Criterion 1: Both sentences include the keywords "morning" and "night."
Criterion 2: Each sentence contains a geographical or temporal entity ("sunrise" and "sunset") related to the context.
Criterion 3: Both sentences are fluent and adhere to commonsense knowledge.
Criterion 4: The sentences have a similar syntactic structure and create a semantic contradiction: "The sky is bright with the sunrise in the early morning, while the sky is dark with the sunset in the late night."
Criterion 5: Swapping the temporal entities "early morning" and "late night" would result in a contradiction: "The sky is bright with the sunrise in the late night, while the sky is dark with the sunset in the early morning."

This example demonstrates how the sentence pairs satisfy the specified criteria of the task.

F Experimental Setup

F.1 Baseline Models

We use HuggingFace [50] implementations of the BART and T5 models. For the decoding method, we adopt the standard beam search with a beam size of 4 for all baseline models. As for checkpoint selection, we save a checkpoint for each epoch and select the checkpoint with the highest `ROUGE-2` on the validation set. Other default hyperparameters are shown in Table 7.

Table 8 shows an example of GPT prompt format, consisting of a fixed instruction ("*Generate a pair of contrastive sentences with the given set of keywords.*") and a few in-context demonstrations ("*Keywords:* $c_1, ..., c_k$ \n Sentences: s_1 s_2").

F.2 Evaluation Metrics

We use the standard implementation of BLEU, ROUGE, METEOR, CIDEr, and SPICE in `pycocoevalcap`[14]. As recommended, we adopt the Recall score of BERTScore[15] and the hash code for evaluation setting is "roberta-large_L17_no-idf_version=0.3.12(hug_trans=4.21.3)-rescaled_fast-

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

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

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 8: An example of InstructGPT prompt format. We only show two in-context demonstrations here for brevity.

Generate a pair of contrastive sentences with the given set of keywords.

Keywords: Kansas, steakhouses, New York City, city, pizzerias
Sentences: Kansas city is known for its steakhouses. New York City is known for its pizzerias.
...
Keywords: seven days, one day, 1,440 minutes, a week
Sentences: There are 1,440 minutes in one day. There are seven days a week.

Keywords: axis, one day, one month, Earth, Moon
Sentences:

tokenizer". 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 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/`