Hence, Socrates is mortal: A Benchmark for Natural Language Syllogistic Reasoning

Anonymous ACL submission

Abstract

Syllogistic reasoning, a typical form of deductive reasoning, is a critical capability widely required in natural language understanding tasks, such as text entailment and question answering. To better facilitate research on syllogistic reasoning, we develop a benchmark called SylloBase that differs from existing syllogistic datasets in three aspects: (1) Covering a complete taxonomy of syllogism reasoning patterns; (2) Containing both automatically and manually constructed samples; and (3) Involving both the generation and understanding tasks. We automatically construct 250k template-based syllogism samples by mining syllogism patterns from Wikidata and ConceptNet. To improve our dataset's naturalness and challenge, we manually rewrite 1,000 samples from template-based data by adding distracting noise and paraphrasing as the test set. State-of-the-art pre-trained language models can achieve the best generation ROUGE-L of 38.06 by BART and the best multi-choice accuracy of 77% by RoBERTa on SylloBase, which indicates the great challenge of learning diverse syllogistic reasoning types on SylloBase.

1 Introduction

Reasoning, as a typical way for human beings to obtain new knowledge and understand the world, is also an ultimate goal of artificial intelligence (Newell and Simon, 1956; Lenat et al., 1990). Reasoning skills, *i.e.*, examine, analyze, and critically evaluate arguments as they occur in ordinary language, have been required by many natural language processing tasks, such as machine reading comprehension (Liu et al., 2020; Yu et al., 2020), open-domain question answering (Kwiatkowski et al., 2019; Huang et al., 2019), and text generation (Dinan et al., 2019).[1] According to different mental processes, reasoning can be categorized as deductive, inductive, abductive, etc.[2] In Piaget's theory of cognitive development (Huitt and Hummel, 2003), these logical reasoning processes are necessary to manipulate information, which is required to use language and acquire knowledge. Therefore, the study of logical reasoning is worthy of our attention because it is so prevalent and essential in our daily lives.

Table 1: An example of categorical syllogism. The colored terms correspond to the symbols in the pattern.

Major premise	Human is mortal.
Minor premise	Socrates is human.
Conclusion	Hence, Socrates is mortal.
Pattern	All m are p, all s are m → All s are p.

In this study, we focus on syllogism, which is a typical form of reasoning and has been studied for a long time (it was initially defined in Aristotle's logical treatises *Organon*, composed around 350 BCE). As shown in Table 1, a syllogism often contains two premises and a conclusion, where the conclusion can be inferred based on the given premises through a deductive reasoning process.[3] Though reasoning-required tasks (such as question answering) have been widely studied, the thorough study to test the deductive reasoning capabilities of a model or system is rare. In the study of syllogism, there are only a few datasets, and they have several limitations: (1) They focus merely on categorical syllogism (shown in Table 1) (Dames et al., 2020; Dong et al., 2020; Aghahadi and Talebpour, 2022). Even though it is the most common type, syllogisms come in a variety of forms. They involve different reasoning processes and are also beneficial. (2) Some datasets (Dames et al., 2020; Dong et al., 2020) are not in natural language, which are difficult to adapt to inference requirements in real

[1] The definition of logical reasoning, https://www.lsac.org/lsat/taking-lsat/test-format/logical-reasoning.

[2] https://en.wikipedia.org/wiki/Reason

[3] There can also be three or more premises. More details are given in Section 3.2.4.

natural language scenarios. (3) More severely, all of them have less than 10k samples, which are not enough for training modern deep neural networks.

To support further study on syllogistic reasoning, in this work, we build a new natural language benchmark – SYLLOBASE, which has the following features (some examples are shown in Table 3): **First**, it is a more complete benchmark that covers five types of syllogisms. Therefore, it can support more fine-grained research on certain types, their interrelationships, and their combined effect on other tasks. **Second**, all premises and conclusions are written in natural language. It more closely resembles real-world application settings in which natural language descriptions rather than categorized inputs are provided. In addition, the power of large-scale pre-trained language models can also be harnessed effectively. **Third**, with our proposed automatic construction process, we collect a large number of samples (250k in total). They can support the training of deep neural networks. In order to validate the performance on actual human syllogism, we also manually annotate 1,000 samples as the test set. This test set may also be used independently to assess the reasoning capability of models in a zero-/few-shot manner. **Finally**, to promote a more comprehensive investigation of syllogistic reasoning, we develop both a generation and an understanding task.

The experimental results indicate that there is a great deal of room for improvement in the syllogistic reasoning capabilities of existing models. Our additional experiments demonstrate the efficacy of transferring knowledge learned from our automatically constructed syllogism to actual human reasoning.

2 Background and Related Work

2.1 Syllogism

Syllogism is a common form of deductive reasoning. Basic syllogism can be categorized as categorical syllogism, hypothetical syllogism, and disjunctive syllogism. They can be further combined into polysyllogisms. In this section, we use the most common categorical syllogism to introduce the term and structure of syllogism. Other types of syllogism will be introduced in Section 3.

Table 1 shows a well-known categorical syllogism about "Socrates is mortal". We can see a categorical syllogism usually contains two premises and a conclusion. A common term (*e.g.*, "human") links two premises, and the premises respectively define the relationship between "human" and "mortal" or "Socrates". The reasoning process is to draw a conclusion based on the two premises. A syllogism can also be described by a pattern, as shown in the last row of Table 1.

2.2 Related Work

Syllogistic Reasoning Dataset Several syllogistic reasoning datasets have been introduced to promote the development of this field. CCOBRA (Dames et al., 2020) is a dataset with around 10k triplets (major premise, minor premise, conclusion). The task is formed as a single-choice question, and the ground-truth conclusion is shuffled with several distractors. ENN (Dong et al., 2020) is another similar dataset, but the syllogism is constructed from WordNet (Miller, 1995). SylloFigure (Peng et al., 2020) and Avicenna (Aghahadi and Talebpour, 2022) are two natural language text-based syllogism reasoning datasets, but they are designed for different tasks. SylloFigure annotates the data in SNLI (Bowman et al., 2015), restores the missing premise, and transforms each syllogism into a specific figure.[4] The target is to predict the correct figure type of a syllogism. Avicenna is a crowdsourcing dataset, and the syllogism is extracted from various sources, such as books and news articles. These syllogisms are used for both natural language generation and inference tasks.

Different from existing datasets that focus only on categorical syllogism, our SYLLOBASE covers more types and patterns of syllogism and is significantly larger than existing datasets. More detailed comparisons are shown in Table 2.

Logic Reasoning in NLP There are several tasks and datasets related to logical reasoning in NLP. The task of natural language inference (NLI) (Bos and Markert, 2005; Dagan et al., 2005; MacCartney and Manning, 2009; Bowman et al., 2015; Williams et al., 2018), also known as recognizing textual entailment, requires model to classify the relationship types (*i.e.*, contradicted, neutral, and entailment) between a pair of sentences. However, this task only focuses on sentence-level logical reasoning, and the relationships are constrained to only a few types. Another NLP task related to logical reasoning is machine reading comprehension (MRC). There are several MRC datasets designed specifi-

[4] Figures in syllogism, https://en.wikipedia.org/wiki/Syllogism.

Table 2: Comparison of existing syllogism datasets.

Dataset	#Types	Natural Language	Complete Patterns	Source	Size
CCOBRA	1 (Categorical)	✗ (Triplet)	✓	Crowdsourcing	10k
ENN	1 (Categorical)	✗ (Triplet)	✓	WordNet	7k
SylloFigure	1 (Categorical)	✓	✗	SNLI	8.6k
Avicenna	1 (Categorical)	✓	✗	Crowdsourcing	6k
SylloBase (Our)	5	✓	✓	Knowledge Base & Crowdsourcing	250k

Table 3: Examples of syllogisms from our test set.

Categorical Syllogism **Premise 1:** Carbon dioxide is a chemical compound. **Premise 2:** Chemical compounds are considered pure substances. **Conclusion:** Pure substances include carbon dioxide.
Hypothetical Syllogism **Premise 1:** When you make progress in your project, you may want to celebrate. **Premise 2:** Having a party is a good choice if you want to celebrate. **Conclusion:** You may want to have a party if you achieve great progress in your project.
Disjunctive Syllogism **Premise 1:** Newspapers are generally published daily or weekly. **Premise 2:** Some newspapers are not published weekly. **Conclusion:** Some newspapers are daily newspapers.
Polysyllogism **Premise 1:** Some movies are not cartoon movies. **Premise 2:** Science fiction animations belong to animated films. **Premise 3:** Remake films are also films. **Conclusion:** Some remakes are out of scope of science fiction cartoons.
Complex Syllogism **Premise 1:** If Jack has computer skills *and* programming knowledge, he could write programs. **Premise 2:** Jack cannot write computer programs, but he can use computers. **Conclusion:** Jack does not have programming knowledge.

cally for logical reasoning, such as LogiQA (Liu et al., 2020) and ReClor (Yu et al., 2020). A paragraph and a corresponding question are given, and the model is asked to select a correct answer from four options. This task requires models to conduct paragraph-level reasoning, which is much more difficult than NLI.

The above logic reasoning NLP tasks attempt to improve models' general logic reasoning capability, but they pay little attention to different types of reasoning processes, such as deductive reasoning or inductive reasoning. In this work, we study a specific form of deductive reasoning, *i.e.*, syllogism. We hope our benchmark can support more in-depth studies on the reasoning process.

3 Data Construction

Our target is to develop a large-scale benchmark and support research on several typical kinds of syllogistic reasoning. It is straightforward to collect data through human annotation, as most existing datasets have explored (Dames et al., 2020; Aghahadi and Talebpour, 2022). However, this method is impracticable for obtaining large-scale data due to the high cost of human annotation. Therefore, we propose constructing a dataset automatically from existing knowledge bases and manually rewriting 1,000 samples as the test set.

3.1 Data Source

Inspired by existing studies (Dong et al., 2020) that collect data from knowledge bases, we choose Wikidata (Vrandecic and Krötzsch, 2014) and ConceptNet (Speer et al., 2017) as our data sources because they contain large-scale high-quality entities and relations.

Wikidata is an open-source knowledge base, serving as a central storage for all structured data from Wikimedia projects. The data model of Wikidata typically consists of two components: *items* and *properties*. Items represent all things in human knowledge. Each item corresponds to a clearly identifiable concept or object, or to an instance of a concept or object. We use entities in the top nine categories, including human, taxon, administrative territorial, architectural structure, occurrence, chemical compound, film, thoroughfare, and astronomical object.[5] Then, we use the relationship of *instance of*, *subclass of*, and *part of* to extract triplets. An example triplet is (*human*, *organisms known by a particular common name*, *Socrates*). These triplets will be used to construct syllogisms.

ConceptNet is another open-source semantic network. It contains a large number of knowledge graphs that connect words and phrases of natural

[5] The full list and the statistics are available at: https://www.wikidata.org/wiki/Wikidata:Statistics.

language with labeled edges (relations). Its knowledge is collected from many sources, where two entities are connected by a closed class of selected relations such as *IsA*, *UsedFor*, and *CapableOf*. We use ConceptNet to extract the descriptive attributes of the entities obtained from Wikidata. By this means, we can obtain another group of triplets, which are also used for constructing syllogism. For example, we can get a triplet for the entity *human* like (*human*, *CapableOf*, *die*), representing the fact that a human will die.

3.2 Data Processing

In this section, we introduce the construction process of five types of syllogism data, respectively. Some examples of different types are shown in Table 3.

3.2.1 Categorical Syllogism

As shown in Table 1, a categorical syllogism is composed of a major premise, a minor premise, and a corresponding conclusion. We first construct premises and then use them to infer the conclusion and form syllogisms.

The premise in a categorical syllogism can be summarized as four propositions according to different quantifiers and copulas:

(1) All S are P; (2) No S are P;
(3) Some S are P; (4) Some S are not P;

where S and P are two entities. With different combinations of the four propositions, categorical syllogisms can be categorized into 24 valid patterns. The first part of Table 3 shows an example of Dimatis syllogism, which is one of the valid patterns.[6] To construct premises, we use the extracted triplets from Wikidata and ConceptNet. To obtain a proposition which contains negative relationship, we can use the *Antonym* and *DistinctFrom* relationship in ConceptNet to construct it. Taking the triplets (*chemical compound*, *subclass of*, *pure substance*) and (*chemical compound*, *Antonym*, *mixture*) as an example, we have:

(1) All chemical compounds are pure substances;
(2) No chemical compounds are mixture;
(3) Some pure substances are chemical compounds;
(4) Some pure substances are not mixture.

By this means, we can obtain various premises, which will be used for constructing syllogisms.

For each pattern of syllogism, we first sample triplets to construct major premises. Then, we use the middle term (*i.e.*, the second entity in the major premise) to sample the minor premises. Finally, the conclusion can be inferred from the major and minor premises directly.

Considering the example in Table 3, which is a Dimatis syllogism, we first sample a triplet (*carbon dioxide*, *IsA*, *chemical compound*). Then, we use the middle term *chemical compound* to sample another triplet (*chemical compound*, *subclass of*, *pure substance*), which forms the minor premise. Finally, we can generate a conclusion based on the pattern definition. All other different patterns of syllogisms can be constructed in a similar way.

3.2.2 Hypothetical Syllogism

Similar to categorical syllogism, a hypothetical syllogism has two premises and a conclusion. The difference is that the premises have one or more hypothetical propositions. A hypothetical Syllogism has three valid patterns (the full list is in Appendix A), and we use five relations (*i.e.*, *Causes*, *HasSubevent*, *HasPrerequisite*, *MotivatedByGoal*, and *CausesDesire*) in ConceptNet to construct hypothetical propositions.

The following pattern is used as an example to illustrate the data construction process:

Premise 1: If P is true, then Q is true.
Premise 2: If Q is true, then R is true.
Conclusion: If P is true, then R is true.

Specifically, we extract a triplet pair where the tail entity of one triplet is the head entity of another triplet, *e.g.*, (*success*, *CausesDesire*, *celebrate*) and (*celebrate*, *CausesDesire*, *have a party*). This triplet pair can construct premises as *success makes you want to celebrate*, and *celebration makes you want to have a party*. Then, we can build a hypothetical syllogism according to the pattern, and the corresponding conclusion is *success makes you want to have a party*. Hypothetical syllogism with other patterns can be constructed in a similar way.

3.2.3 Disjunctive Syllogism

A disjunctive syllogism has two premises: One of them is a compound proposition, which tells that at least one proposition is true; The other premise tells that one proposition in the former premise is false. Then, we can infer another proposition in the former premise is true. For example, if P and Q are two propositions, a disjunctive syllogism can

[6] Other patterns can be referred to in Appendix A.

be described as:

Premise 1: P is true or Q is true;
Premise 2: P is not true;
Conclusion: Q is true.

According to whether the two propositions can be both true, a disjunctive syllogism can be categorized as compatible or incompatible.

We use ten relations in ConceptNet to construct disjunctive syllogism, where eight of them (such as *PartOF* and *HasA*) are used for compatible disjunctive syllogism, and the rest two (*i.e.*, *Antonym* and *DistinctFrom*) are used for incompatible disjunctive syllogism (all relations we used are listed in Appendix B). Here, we use the incompatible disjunctive syllogism as an example to illustrate the construction process.

We first sample a triplet for an entity, such as (*newspapers*, *CapableOf*, *come weekly*) and (*newspapers*, *CapableOf*, *come daily*). Then, we can construct a premise as *newspapers can come weekly or come daily*. Next, we obtain another premise, such as *some newspapers cannot come weekly*. Finally, we can have the conclusion as *some newspapers come daily*. In this way, we can automatically construct various disjunctive syllogisms based on the triplets in ConceptNet.

3.2.4 Polysyllogism

A polysyllogism is a combination of a series of categorical syllogism. It usually contains three or more categorical propositions, and the conclusion is also a categorical proposition.

The construction of polysyllogism can be summarized as the following steps:

(1) We sample a categorical syllogism from our categorical syllogism repository (built in Section 3.2.1).

(2) According to the form of the conclusion, we can get its predicate term and subject term.

(3) We use these terms to traverse the repository and select a premise/conclusion that contains them.

(4) We use the conclusion obtained in the second step and the selected premise/conclusion in the third step as two new premises. Then, we can infer the conclusion and check if the generated syllogism follows a valid pattern.

(5) Repeat the above process, and we can obtain a series of syllogisms.

(6) We use both premises in the first syllogism and the minor premise in all other syllogisms as the premises of the polysyllogism. The conclusion is obtained from the last syllogism's conclusion. By this means, we can construct a polysyllogism.

Table 4: An example of categorical syllogism before manual rewriting.

Original	
Premise 1	Some chemical compounds are carbon dioxide.
Premise 2	All chemical compounds are pure substances.
Conclusion	Some pure substances are carbon dioxide.

We provide an example in the fourth row of Table 3 to illustrate the construction process.

3.2.5 Complex Syllogism

In addition to constructing the previous four types of syllogism, we investigate another new type of syllogism, which is called complex syllogism. A complex syllogism contains two premises and a conclusion, and the premises and conclusion are compound propositions, which contain one or more logical connectives (*i.e.*, *not*, *and*, *or*, and *if-then*). These logical connectives significantly increase the difficulty of the syllogism. An example of a complex syllogism is shown in the last row of Table 3. The construction steps can be summarized as:

(1) We randomly sample a pattern from hypothetical and disjunctive syllogism as a basic pattern.

(2) We replace the simple propositions in the basic pattern (such as P, Q, and R) by a compound proposition with the logical connectives *not*, *and*, and *or*, (*e.g.*, *not P*, *P or Q*, and *P and Q*).

(3) After the replacement, we can infer the conclusion (according to the pattern we derived, as shown in Appendix A) and construct a complex syllogism.

Rule of Replacement To replace a simple proposition by a compound proposition, we use the *Synonyms* relation in ConceptNet. For example, considering the proposition *something that might happen as a consequence of eating ice cream is pleasure*, we use the synonym of the entity *ice cream*, *i.e.*, *cone*, and construct a compound proposition as *something that might happen as a consequence of eating ice cream and cone is pleasure*.

3.3 Manual Rewriting for Test Set

To test the models' performance on (real) syllogisms and facilitate future in-depth research, we manually rewrite 1,000 samples (200 of each type) from our collected data as a test set. The rewriting process is as follows: First, 500 samples are

Table 5: Results of conclusion generation task.

	Metric	Transformer	GPT-2	T5	BART
Categorical	ROUGE-1	27.93	34.14	37.95	41.59
	ROUGE-2	6.91	8.10	9.93	11.98
	ROUGE-L	24.80	27.74	28.55	32.06
	BLEU-1	19.65	22.54	21.93	25.93
	BLEU-2	4.44	4.57	4.45	6.03
	BERT-Score	87.81	88.49	89.79	90.28
Hypothetical	ROUGE-1	21.49	29.06	35.29	35.12
	ROUGE-2	6.68	8.62	13.83	14.41
	ROUGE-L	19.51	24.84	30.97	30.73
	BLEU-1	19.21	19.75	26.01	25.5
	BLEU-2	6.43	5.37	10.70	10.17
	BERT-Score	89.84	89.29	91.75	91.63
Disjunctive	ROUGE-1	38.05	41.81	51.76	51.97
	ROUGE-2	17.55	15.61	29.29	30.93
	ROUGE-L	35.51	38.09	48.91	49.76
	BLEU-1	30.68	30.27	36.89	41.07
	BLEU-2	12.25	10.11	19.82	22.92
	BERT-Score	91.20	91.29	93.91	93.84
Polysyllogism	ROUGE-1	42.87	48.45	51.64	50.17
	ROUGE-2	16.10	21.39	24.94	22.20
	ROUGE-L	37.84	43.46	46.53	45.23
	BLEU-1	30.73	33.94	37.11	35.99
	BLEU-2	10.91	13.44	14.94	14.75
	BERT-Score	90.55	90.24	92.09	91.98
Complex	ROUGE-1	33.06	40.65	45.76	47.83
	ROUGE-2	14.19	15.31	20.37	23.51
	ROUGE-L	31.61	38.23	43.58	45.17
	BLEU-1	26.98	30.53	37.16	39.74
	BLEU-2	10.88	9.98	16.85	19.15
	BERT-Score	90.88	90.62	92.90	93.04
All	ROUGE-1	28.47	35.53	40.95	42.07
	ROUGE-2	9.71	12.18	17.51	18.78
	ROUGE-L	26.15	31.67	36.80	38.06
	BLEU-1	22.47	23.76	28.89	30.23
	BLEU-2	6.98	7.00	11.87	13.08
	BERT-Score	89.41	89.69	91.80	91.81

randomly collected from each type of syllogism, respectively. Then, we examine the semantics and filter out illogical syllogisms. Next, for the remaining ones, we correct the grammatical problems (if any). Finally, for each premise/conclusion, the language is painstakingly paraphrased. Table 4 displays an example before our manual rewriting, and it corresponds to the first example in Table 3. It is evident that the sample after rewriting is more diverse, fluent, and closer to real human language. Our experiments (see Section 4.5) will show that the test data are very challenging, whereas training on our automatically collected data is still effective.

4 Experiments

4.1 Task Formalization

Based on our collected data, we design two tasks:

Conclusion Generation It is a natural language generation task. The model should generate the correct conclusion based on two given premises. Premises and conclusions are natural language text, which can be represented as sequences of tokens. Formally, given two premises $P_1 = \{w_1^{P_1}, \cdots, w_m^{P_1}\}$ and $P_2 = \{w_1^{P_2}, \cdots, w_n^{P_2}\}$, the model is asked to generate the conclusion $C = \{w_1^C, \cdots, w_l^C\}$, where w is a token. Similar to other text generation tasks, the generation probability of the conclusion is determined by the product of the probability of each word, which can be described as: $P(C|P_1, P_2) = \prod P(w_i^C | w_{<i}^C, [P_1; P_2])$, where [;] is concatenation operation. More premises can be handled by concatenate all of them as a long sequence.

Conclusion Selection It is a natural language understanding task. The model is asked to select a correct conclusion from four options, where three of them are distractors.[7] With the above notations of premises and conclusion, we can define the conclusion selection task as:

$$S(C_i, [P_1; P_2]) = \frac{\exp(M(C_i, [P_1; P_2]))}{\sum_{j=1}^{4} \exp(M(C_j, [P_1; P_2]))},$$

where $S(C_i, [P_1; P_2])$ is the predicted probability of C_i as a correct conclusion, and $M(\cdot, \cdot)$ is the output logit of the model.

The statistics of our dataset for both tasks are given in Appendix D.

4.2 Baseline and Evaluation Metrics

We compare the performance of several models. For the conclusion generation task, we consider Transformer (Vaswani et al., 2017) and several pre-trained models, including GPT-2 (Radford et al., 2019), T5 (Raffel et al., 2020), and BART (Lewis et al., 2020). For the conclusion selection task, we employ BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), XLNet (Yang et al., 2019), and ELECTRA (Clark et al., 2020) as baseline methods. For all pre-trained models, we use the base version.

As for evaluation metrics, following previous studies (Aghahadi and Talebpour, 2022), we use ROUGE-1/2/L (Lin, 2004), BLEU-1/2 (Papineni et al., 2002), and BERT-Score (Zhang et al., 2020) to evaluate the performance of the conclusion generation task. ROUGE and BLEU are commonly used metrics for text generation, and they measure the n-grams overlap between the generated

[7] Detailed construction process is given in Appendix C.

text and the ground-truth one. BERT-Score is a recently proposed model-based metric. It leverages the pre-trained contextual embeddings from BERT and matches words in generated and ground-truth texts by cosine similarity. It has been shown to correlate with human judgment on sentence-level and system-level evaluation. For the conclusion selection task, we use Accuracy to evaluate the models' performance.

4.3 Implementation Details

We use PyTorch (Paszke et al., 2019) and Transformers (Wolf et al., 2019) to implement all models. They are trained on 8 Tesla V100 GPUs with 32GB memory. All hyperparameters (*e.g.*, learning rate) are tuned according to the performance (BLEU-1/Accuracy) on the validation set.

In the conclusion generation task, for the decoder-only model GPT-2, the major premise, minor premise, and conclusion are concatenated as a long sequence and fed into the model. The loss is only computed in the conclusion part. For the encoder-decoder structure (Transformer, T5, and BART), the two premises are concatenated and input to the encoder, while the conclusion is input to the decoder and used for generation. The maximum generation length is set as 128. The training batch size is set as 32. The AdamW (Loshchilov and Hutter, 2019) optimizer is applied with a learning rate of 5e-5. The learning rate decay mechanism is applied. All models are trained by 100 epochs, and the total training time is around 32.875 hours.

In the conclusion selection task, we concatenate two premises as one sequence, use the conclusion as another sequence, and transform them into the text-pair input format, which is commonly supported by pre-trained language models. For example, the input for BERT is: $X =$ `[CLS]` $P_1 P_2$ `[SEP]` C `[SEP]`. The representation of `[CLS]` is used for option selection. The maximum sequence length is set as 256. The training batch size is set as 64. A learning rate of 2e-5 with decay mechanism is used. The optimizer is also AdamW. All models are trained by ten epochs, and the total training time is around 28.625 hours.

4.4 Experimental Results

The results of all models on the conclusion generation task are shown in Table 5, while those on the conclusion selection task are reported in Table 6.

Table 6: Accuracy of conclusion selection task.

Type	BERT	RoBERTa	XLNet	ELECTRA
Categorical	49.50	47.00	49.50	43.50
Hypothetical	88.73	90.20	92.65	88.73
Disjunctive	96.52	97.01	96.02	97.50
Polysyllogism	57.14	64.53	60.10	62.56
Complex	88.12	92.08	93.07	93.07
All	74.55	77.82	76.93	75.15

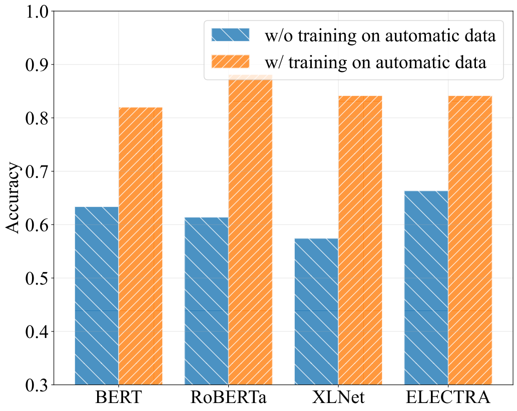

Figure 1: Results of the conclusion selection task with or without pre-training on automatic training data.

For the conclusion generation task, we can see that the overall performance in terms of word-overlap metrics (such as ROUGE and BLEU) is poor. Given that conclusions are often brief (7.79 tokens on average), these results show that the task is fairly challenging. In contrast, the BERT-Score is high, indicating that models are able to generate some key information but cannot organize it into a reasonable conclusion. Furthermore, the pre-trained language models perform significantly better than the vanilla Transformer. We attribute this to the natural language nature of our dataset, and these results suggest that our dataset can help future research on leveraging pre-trained language models to generate text that is logically reasonable.

For the conclusion selection task, the overall accuracy is around 75%, showing a significant deviation from perfection. Intriguingly, the performance on categorical syllogisms and polysyllogisms is extremely bad. A potential reason is that these two types of syllogisms contain more patterns (*e.g.*, categorical syllogisms have 24 valid patterns). As a comparison, the performance on hypothetical syllogisms is significantly higher since there are only three patterns. We also notice that the performance on polysyllogisms is higher than that on categorical syllogisms, despite the fact that the former is derived from the latter. We speculate the reason is that the polysyllogisms have more abundant information in premises (*i.e.*, multiple premises), which is

Table 7: Results (ROUGE-1/2/L) of the conclusion generation task with or without pre-training on automatic training data.

Model	*w/o* Automatic data	*w/* Automatic data
Transformer	24.26 / 7.32 / 22.4	32.79 / 11.59 / 30.06
GPT-2	32.96 / 11.62 / 29.64	44.26 / 17.19 / 39.59
T5	44.03 / 20.31 / 39.01	50.33 / 25.03 / 46.18
BART	45.96 / 22.79 / 41.98	52.27 / 26.81 / 47.08

Table 8: An example of syllogism with context. The vanilla premises are in red.

Premise 1: Carbon dioxide is a chemical compound composed of two oxygen atoms covalently bonded to a single carbon atom. CO2 exists in the earth's atmosphere as a gas and in its solid state it known as dry ice.
Premise 2: In a scientific context, "pure" denotes a single type of material. Ostensibly, compounds contain more than one type of material. Therefore, chemical compounds are considered pure substances. Pure compounds are created when elements combine permanently, forming one substance.
Conclusion: Pure substances include carbon dioxide.

Table 9: Impact of context for conclusion generation (ROUGE-1/2/L) and conclusion selection (Accuracy).

Model	*w/o* Context	*w/* Context
Transformer	28.47 / 9.71 / 26.15	14.19 / 2.58 / 13.13
GPT-2	35.53 / 12.18 / 31.67	21.82 / 5.67 / 19.89
T5	40.95 / 17.51 / 36.8	23.79 / 7.13 / 21.68
BART	42.07 / 18.78 / 38.06	23.38 / 7.27 / 21.31
BERT	74.55	65.84
RoBERTa	77.82	66.53
XLNet	76.93	69.01
ELECTRA	75.15	66.34

helpful for pre-trained language models to conduct reasoning.

4.5 Further Analysis

We also explore the following research questions:

Effect of Automatically Constructed Data In our benchmark, the training data are automatically constructed from knowledge bases, while the test data are human annotated.[8] To reveal the relationship between these two kinds of data, we conduct an additional experiment as: we split the test set as new training, validation, and test sets with a ratio of 8:1:1 (*i.e.*, they have 800, 100, and 100 samples respectively). Then, we train new models on the new training data and test their performance on the new test data. As a comparison, we train another model that has been pre-trained on the original training data (automatically constructed). The results are illustrated in Figure 1 and Table 7.

It is clear to see that training on automatically constructed data is beneficial for learning on manually rewritten data. This is due to the fact that the original dataset is large and contains sufficient training signals. This also validates the benefit of our dataset – the knowledge acquired from large-scale data can be transferred to more difficult problems.

Effect of Context in Premises Existing machine reading comprehension datasets often provide a paragraph for reasoning. Inspired by these tasks, we expand the premises in our generated syllogisms by adding more informative context so as to validate the models' capability of extracting effective clues and inferring conclusions. Specifically, for each premise in the manually rewritten dataset, we ask the annotators to further collect some relevant information through search engines and add it as the context. After this step, both premises are hidden in paragraphs, which makes it more difficult to infer a correct conclusion (as shown in Table 8). Results of both tasks shown in Table 9 indicate: (1) Existing models are still far from tackling reasoning problems in real life; and (2) Extracting clues (such as premises in our case) before reasoning is a promising solution for reasoning tasks, which could be explored in the future.

Appendix F shows the model generated conclusions of syllogisms in Table 3, and we analyze the limitation of this work in Appendix G.

5 Conclusion

In this work, we built a large-scale benchmark for natural language syllogistic reasoning. It covers five types of syllogism. The data were automatically constructed from knowledge bases by our proposed construction methods. To evaluate the models' performance on real human syllogism, we manually rewrote 1,000 samples as the test set. Experiments showed that syllogistic reasoning is a very challenging task for existing pre-trained language models. Moreover, our further study indicated that existing models are even farther from tackling syllogistic reasoning in real scenarios.

[8] We also perform a human evaluation on 100 automatically constructed samples (20 for each type of syllogisms). About 72% samples are grammatically perfect and logically correct. More details can be referred to at Appendix E.

Ethical Statement

This work constructs a new benchmark for syllogistic reasoning. The main dataset is automatically constructed using entities and their relations from Wikidata and ConceptNet. The construction template is predefined and manually reviewed, so the ethical concerns are avoided. For the human rewriting process, we hire five annotators and require them to avoid any social bias and privacy issues in the rewritten material. The results are randomly shuffled and sent back to them for an ethical review. We pay them roughly $15 per hour for annotation.

References

Zeinab Aghahadi and Alireza Talebpour. 2022. Avicenna: a challenge dataset for natural language generation toward commonsense syllogistic reasoning. *Journal of Applied Non-Classical Logics*, 0(0):1–17.

Johan Bos and Katja Markert. 2005. Recognising textual entailment with logical inference. In *HLT/EMNLP 2005, Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference, 6-8 October 2005, Vancouver, British Columbia, Canada*, pages 628–635. The Association for Computational Linguistics.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 632–642. The Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The PASCAL recognising textual entailment challenge. In *Machine Learning Challenges, Evaluating Predictive Uncertainty, Visual Object Classification and Recognizing Textual Entailment, First PASCAL Machine Learning Challenges Workshop, MLCW 2005, Southampton, UK, April 11-13, 2005, Revised Selected Papers*, volume 3944 of *Lecture Notes in Computer Science*, pages 177–190. Springer.

Hannah Dames, Clemens Schiebel, and Marco Ragni. 2020. The role of feedback and post-error adaptations in reasoning. In *Proceedings of the 42th Annual Meeting of the Cognitive Science Society - Developing a Mind: Learning in Humans, Animals, and Machines, CogSci 2020, virtual, July 29 - August 1, 2020*. cognitivesciencesociety.org.

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.

Emily Dinan, Varvara Logacheva, Valentin Malykh, Alexander H. Miller, Kurt Shuster, Jack Urbanek, Douwe Kiela, Arthur Szlam, Iulian Serban, Ryan Lowe, Shrimai Prabhumoye, Alan W. Black, Alexander I. Rudnicky, Jason Williams, Joelle Pineau, Mikhail S. Burtsev, and Jason Weston. 2019. The second conversational intelligence challenge (convai2). *CoRR*, abs/1902.00098.

Tiansi Dong, Chengjiang Li, Christian Bauckhage, Juanzi Li, Stefan Wrobel, and Armin B. Cremers. 2020. Learning syllogism with Euler neural-networks. *CoRR*, abs/2007.07320.

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.

William Huitt and John Hummel. 2003. Piaget's theory of cognitive development. *Educational psychology interactive*, 3(2).

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur P. Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: a benchmark for question answering research. *Trans. Assoc. Comput. Linguistics*, 7:452–466.

Douglas B. Lenat, Ramanathan V. Guha, Karen Pittman, Dexter Pratt, and Mary Shepherd. 1990. CYC: toward programs with common sense. *Commun. ACM*, 33(8):30–49.

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.

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.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. 2020. Logiqa: A challenge dataset for machine reading comprehension with logical reasoning. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 3622–3628. ijcai.org.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Bill MacCartney and Christopher D. Manning. 2009. An extended model of natural logic. In *Proceedings of the Eight International Conference on Computational Semantics, IWCS 2009, Tilburg, The Netherlands, January 7-9, 2009*, pages 140–156. Association for Computational Linguistics.

George A. Miller. 1995. Wordnet: A lexical database for English. *Commun. ACM*, 38(11):39–41.

Allen Newell and Herbert A. Simon. 1956. The logic theory machine-a complex information processing system. *IRE Trans. Inf. Theory*, 2(3):61–79.

Feng Nie, Jin-Ge Yao, Jinpeng Wang, Rong Pan, and Chin-Yew Lin. 2019. A simple recipe towards reducing hallucination in neural surface realisation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2673–2679. 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.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8024–8035.

Shiya Peng, Lu Liu, Chang Liu, and Dong Yu. 2020. Exploring reasoning schemes: A dataset for syllogism figure identification. In *Chinese Lexical Semantics - 21st Workshop, CLSW 2020, Hong Kong, China, May 28-30, 2020, Revised Selected Papers*, volume 12278 of *Lecture Notes in Computer Science*, pages 445–451. Springer.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

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.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 4444–4451. AAAI Press.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Denny Vrandecic and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 5754–5764.

Weihao Yu, Zihang Jiang, Yanfei Dong, and Jiashi Feng. 2020. Reclor: A reading comprehension dataset requiring logical reasoning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

A Patterns in Syllogism

We list all valid patterns in categorical (shown in Table 10), hypothetical (shown in Table 11), and complex syllogisms (shown in Table 12).

B Relations from Wikidata and ConceptNet

We list all relations that are used for constructing syllogisms in Table 13.

C Distractor Construction in Conclusion Selection Task

In the conclusion selection task (introduced in Section 4.1), we mix the correct conclusion with three distractors. Basically, these distractors are generated from the ground-truth conclusion by changing its quantifier, adding negative words, or exchanging its subject and object. Specifically, for different kinds of syllogisms, we show the distractor generation process by some examples.

Categorical Syllogism For a syllogism as follows:

Premise 1: All m are p.
Premise 2: All s are m.
Conclusion: All s are p.

We can generate distractors of the conclusion as:

(1) Some s are p. (*modify quantifiers*)
(2) All s are not p. (*add negative words*)
(3) All p are s. (*exchange subjects and predicates*)
(4) Some p are not s. (*others*)

Hypothetical Syllogism For a syllogism as follows:

Premise 1: If P is true, then Q is true.
Premise 2: If Q is true, then R is true.
Conclusion: If P is true, then R is true.

We can generate distractors of the conclusion as:

(1) If R is true, then P is true. (*exchange propositions*)
(2) If Q is true, then P is true. (*exchange propositions*)
(3) If R is true, then Q is true. (*exchange propositions*)
(4) P is true. (*remove a proposition*)
(5) Q is true. (*remove a proposition*)
(6) R is true. (*remove a proposition*)
(7) If P is true, then R is not true. (*add negative words*)

Disjunctive Syllogism For a syllogism as follows:

Premise 1: P is true or Q is true;
Premise 2: P is not true;
Conclusion: Q is true.

We can generate distractors of the conclusion as:

(1) Q is not true. *(add negative words)*
(2) P is true. *(change a proposition)*
(3) P is true or Q is not true. *(add a proposition)*

Polysyllogism Syllogism This kind of syllogism is built on several categorical syllogisms. Therefore, we can use the same distractor construction method as categorical syllogisms.

Complex Syllogism This kind of syllogism is constructed by adding one or model logical connectives to the original premises and conclusions. Therefore, to generate the distractors, we can (1) add or remove the negative connective (*i.e.*, *not*) from the original proposition; or (2) replace the connectives in the original proposition by others (*e.g.*, *and* → *or*). For example, given a syllogism as follows:

Premise 1: If P is true or if Q is true, then R is true;
Premise 2: If R is true, then S is true;
Conclusion: If P is true or if Q is true, then S is true.

We can generate distractors of the conclusion as:

(1) If P is true or if Q is true, then S is not true. *(add negative words)*
(2) If P is true or if S is true, then Q is true. *(change a proposition)*
(3) If P is true and if S is true, then Q is true. *(change the logical connective words)*

D Dataset Statistics

The statistics of our SylloBase is given in Table 14.

E Annotation of Automatic Data

To evaluate the quality of our automatically generated data, we conduct a human annotation for 100 random samples (20 for each type of syllogisms). The annotators are asked to label whether the samples have grammatical faults and incorrect logic. The overall accuracy is 72%. Concretely, the accuracy is 75%, 80%, 70%, 65%, and 70% for categorical syllogisms, hypothetical syllogisms,

Table 10: 24 valid patterns in categorical syllogisms.

Pattern	Figure	Major premise	Minor premise	Conclusion
Barbara (AAA)	1	All m are p	All s are m	All s are p
Barbari (AAI*)	1	All m are p	All s are m	Some s are p
Calarent (EAE)	1	No m is p	All s are m	No s is p
Celaront (EAO*)	1	No m is p	All s are m	Some s are not p
Darii (AII)	1	All m are p	Some s are m	Some s are p
Ferio (EIO)	1	No m is p	All s are m	Some s are not p
Camestres (AEE)	2	All p are m	No s is m	No s is p
Camestros (AEO*)	2	All p are m	No s is m	Some s are not p
Cesare (EAE)	2	No p is m	All s are m	No s is p
Cesaro (EAO*)	2	No p is m	All s are m	Some s are not p
Baroco (AOO)	2	All p are m	Some s are not m	Some s are not p
Festino (EIO)	2	No p is m	Some s are m	Some s are not p
Darapti (AAI)	3	All m are p	All m are s	Some s are p
Felapton (EAO)	3	No m is p	All m are s	Some s are not p
Datisi (AII)	3	All m are p	All m are s	Some s are p
Disamis (IAI)	3	Some m are p	All m are s	Some s are p
Bocardo (OAO)	3	Some m are not p	All m are s	Some s are not p
Ferison (EIO)	3	No m is p	Some m are s	Some s are not p
Bamalip (AAI)	4	All p are m	All m are s	Some s are p
Calemes (AEE)	4	All p are m	No m is s	No s is p
Calemos (AEO*)	4	All p are m	No m is s	Some s ara not p
Fesapo (EAO)	4	No p is m	All m are s	Some s are not p
Dimatis (IAI)	4	Some p are m	All m are s	Some s are p
Fresison (EIO)	4	No p is m	Some m are s	Some s are not p

Table 11: Three valid patterns in hypothetical syllogism. P, Q, and R are three propositions.

Original hypothetical syllogism		
Premise 1: If P is true, then Q is true.	Premise 2: If Q is true, then R is true.	Conclusion: If P is true, then R is true.
Modus ponens		
Premise 1: If P is true, then Q is true.	Premise 2: P is true.	Conclusion: Q is true.
Modus tollens		
Premise 1: If P is true, then Q is true.	Premise 2: Q is not true.	Conclusion: P is not true.

disjunctive syllogisms, polysyllogisms, and complex syllogisms, respectively. This result reflects: (1) Our automatic data have fairly good quality. Our experiments in Section 4.5 also validates this. (2) The polysyllogism is hard to construct as it concerns multiple syllogisms.

F Case Study

We show some results of BART in conclusion generation task to make a case study. They are shown in Table 15. We can see: (1) The model can generate conclusions that are different from the ground-truth but are also correct in logic (*e.g.*, the first, third, and fourth case). This indicates that pre-trained language models can indeed learn some logic reasoning skills from syllogisms rather than merely "remembering" some fixed patterns. (2) Syllogistic reasoning is still difficult for existing models, and the errors stem from several different aspects. As shown in the hypothetical syllogism, the model generates a semantically correct conclusion, but it is irrelevant to the premises. This problem is identified as "hallucination" of pre-trained language models (Nie et al., 2019), *i.e.*, the model cannot decide whether to generate a conclusion based on its learned parameters or the given context. We believe our dataset can contribute to the study of hallucinations in logical reasoning. As for the last case, the model generate a conclusion opposite to the ground-truth. This indicates that existing models may need additional reasoning modules to conduct complex reasoning problems.

G Limitations

We build a new benchmark for syllogistic reasoning. The limitations are mainly in the experiments part: (1) Due to the limited human resources, our test set is quite small, which may not support training large models directly. (2) We evaluate all models by comparing their predictions with the ground-

Table 12: 42 valid patterns in complex syllogisms.

Id	Premise 1	Premise 2	Conclusion
0	$\neg p \vee q$	p	q
1	$(p \wedge q) \vee r$	$\neg p \vee \neg q$	r
2	$(p \vee q) \vee r$	$\neg p \wedge \neg q$	r
3	$p \vee \neg q$	$\neg p$	$\neg q$
4	$p \vee (q \wedge r)$	$\neg p \wedge q$	r
5	$p \vee (q \wedge r)$	$\neg p \wedge r$	q
6	$p \vee (q \vee r)$	$\neg p \wedge \neg r$	q
7	$\neg p \vee q$	$\neg q$	$\neg p$
8	$p \vee (q \vee r)$	$\neg q \wedge \neg r$	p
9	$(p \wedge q) \vee r$	$p \wedge \neg r$	q
10	$(p \wedge q) \vee r$	$q \wedge \neg r$	p
11	$p \vee \neg q$	q	p
12	$p \vee (q \wedge r)$	$\neg q \vee \neg r$	p
13	$\neg q \rightarrow \neg p$	$\neg q$	$\neg p$
14	$(p \vee q) \rightarrow r$	$p \vee q$	r
15	$(p \wedge q) \rightarrow r$	$p \wedge q$	r
16	$p \rightarrow (q \vee r)$	p	$q \vee r$
17	$p \rightarrow (q \vee r)$	$p \wedge \neg q$	r
18	$p \rightarrow (q \vee r)$	$p \wedge \neg r$	q
19	$p \rightarrow (q \wedge r)$	p	$q \wedge r$
20	$p \rightarrow (q \wedge r)$	$p \wedge q$	r
21	$p \rightarrow (q \wedge r)$	$p \wedge r$	q
22	$(p \vee q) \rightarrow r$	$\neg r$	$\neg (p \vee q)$
23	$(p \vee q) \rightarrow r$	$\neg p \wedge \neg r$	$\neg q$
24	$(p \vee q) \rightarrow r$	$\neg q \wedge \neg r$	$\neg p$
25	$(p \wedge q) \rightarrow r$	$\neg r$	$\neg (p \wedge q)$
26	$(p \wedge q) \rightarrow r$	$p \wedge \neg r$	$\neg q$
27	$(p \wedge q) \rightarrow r$	$q \wedge \neg r$	$\neg p$
28	$p \rightarrow (q \vee r)$	$\neg q \wedge \neg r$	$\neg p$
29	$p \rightarrow (q \wedge r)$	$\neg q \vee \neg r$	$\neg p$
30	$\neg q \rightarrow \neg p$	$\neg r \rightarrow \neg q$	$\neg r \rightarrow \neg p$
31	$(p \vee q) \rightarrow r$	$r \rightarrow s$	$(p \vee q) \rightarrow s$
32	$(p \vee q) \rightarrow r$	$(r \rightarrow s) \wedge p$	s
33	$(p \vee q) \rightarrow r$	$(r \rightarrow s) \wedge q$	s
34	$(p \wedge q) \rightarrow r$	$r \rightarrow s$	$(p \wedge q) \rightarrow s$
35	$(p \wedge q) \rightarrow r$	$(r \rightarrow s) \wedge p \wedge q$	s
36	$p \rightarrow (q \vee r)$	$(q \vee r) \rightarrow s$	$p \rightarrow s$
37	$p \rightarrow (q \wedge r)$	$(q \wedge r) \rightarrow s$	$p \rightarrow s$
38	$p \rightarrow q$	$q \rightarrow (r \vee s)$	$p \rightarrow (r \vee s)$
39	$p \rightarrow q$	$(q \rightarrow (r \vee s)) \wedge p$	$r \vee s$
40	$p \rightarrow q$	$q \rightarrow (r \wedge s)$	$p \rightarrow (r \wedge s)$
41	$p \rightarrow q$	$(q \rightarrow (r \wedge s)) \wedge p$	$r \wedge s$

truth conclusions, but human performance is not evaluated. As a benchmark, it may be better to provide human performance and show the performance gap of existing models. (3) We have not tested the performance of pre-trained models in terms of logical correctness. This kind of automatic metrics has been rarely studied, which can be a potential direction of our future work.

Table 13: Relations used for syllogisms construction.

Type	Used Relations
Wikidata	
Categorical	academic degree subclass (human)
Categorical	ethnic subclass (human)
Categorical	field of work subclass (human)
Categorical	genre subclass (human)
Categorical	occupation subclass (human)
Categorical	language subclass (human)
Categorical	instance of (human)
Categorical	instance of (taxon)
Categorical	taxon subclass (taxon)
Categorical	film subclass (film)
Categorical	chemical compound subclass (chemical compound)
Categorical	administrative territorial subclass (administrative territorial)
Categorical	architectural structure subclass (architectural structure)
Categorical	astronomical object subclass (astronomical object)
Categorical	occurrence subclass (occurrence)
Categorical	thoroughfare subclass (thoroughfare)
ConceptNet	
Categorical / Disjunctive	/r/CapableOf
Categorical / Disjunctive	/r/HasProperty
Categorical / Disjunctive	/r/Antonym
Categorical / Disjunctive	/r/DistinctFrom
Disjunctive	/r/Part of
Disjunctive	/r/HasA
Disjunctive	/r/UsedFor
Disjunctive	/r/SymbolOf
Disjunctive	/r/MannerOf
Disjunctive	/r/MadeOf
Hypothetical	/r/Causes
Hypothetical	/r/HasSubevent
Hypothetical	/r/HasPrerequisite
Hypothetical	/r/MotivatedByGoal
Hypothetical	/r/CausesDesire

Table 14: Statistics of SylloBase.

Conclusion Generation	Training	Validation	Test (*w/o* context)	Test (*w/* context)
# Premises-Conclusion Pair	240,000	10,000	1,000	1,000
Avg./Max. # Tokens in Premises	23.52 / 232	23.50 / 232	27.59 / 75	183.92 / 726
Avg./Max. # Tokens in Conclusion	7.79 / 88	7.77 / 44	8.5 / 21	8.5 / 21
Conclusion Selection	**Training**	**Validation**	**Test (*w/o* context)**	**Test (*w/* context)**
# Premises-Question Pair	240,000	10,000	1,000	1,000
Avg./Max. # Tokens in Premises	23.52 / 232	23.50 / 232	27.59 / 75	183.92 / 726
Avg./Max. # Tokens in Question	12.41 / 16	12.39 / 16	12.38 / 16	12.38 / 16
Avg./Max. # Tokens in Candidate Answer	8.88 / 88	8.85 / 44	9.41 / 26	9.41 / 26

Table 15: Examples of conclusions generated by BART. The good results and bad results are in different colors.

Categorical Syllogism **Premise 1:** Carbon dioxide is a chemical compound. **Premise 2:** Chemical compounds are considered pure substances. **Conclusion:** Pure substances include carbon dioxide. **Prediction**: Some completely natural substances include carbon dioxide.
Hypothetical Syllogism **Premise 1:** When you make progress in your project, you may want to celebrate. **Premise 2:** Having a party is a good choice if you want to celebrate. **Conclusion:** You may want to have a party if you achieve great progress in your project. **Prediction**: Celebrating has the effect of bringing people together.
Disjunctive Syllogism **Premise 1:** Newspapers are generally published daily or weekly. **Premise 2:** Some newspapers are not published weekly. **Conclusion:** Some newspapers are daily newspapers. **Prediction**: Some newspapers are generally published daily.
Polysyllogism **Premise 1:** Some movies are not cartoon movies. **Premise 2:** Science fiction animations belong to animated films. **Premise 3:** Remake films are also films. **Conclusion:** Some remakes are out of scope of science fiction cartoons. **Prediction**: It exists some remakes are out of scope of science fiction cartoon.
Complex Syllogism **Premise 1:** If Jack has computer skills *and* programming knowledge, he could write programs. **Premise 2:** Jack cannot write computer programs, but he can use computers. **Conclusion:** Jack does not have programming knowledge. **Prediction**: He can write computer programs.