Evaluation of logical equivalence using auto-generated corpus on BERT

Shunjie Wang, Jize Cao, Yuanbo Xu, Sicong Huang

University of Washington, Seattle
{shunjiew, caojize, yuanboxu, huangs33}@uw.edu

Abstract

Many existing Natural Language Inference (NLI) datasets are easily exploited by spurious statistical cues, making NLI a trivial task to solve and are not suitable for capturing structural information in the data. Previous work targeting specific semantic phenomenon has attempted using formal language templates to generate synthesized data for exploiting structural forms. This work employs a similar data generation approach for assessing semantic knowledge in neural language models. specifically, learning logical equivalences. We proposed a dataset to test the ability of BERT in distinguishing the logic relation of two sentences. The data generation process is linguistically-motivated and governed by transformation rules in logic. Results show that upon fine-tuning, BERT is very capable at distinguishing the logical relation of two sentences, but pre-trained BERT cannot capture such information without fine-tuning.

1 Introduction

Transformer-based language models pre-trained on large datasets and fine-tuned on specific tasks have achieved many successes on a wide range of NLP tasks. However, Niven and Kao (2019) recently pointed out that the success of such models may not be due to their ability to generalize and understand natural language, but rather the models are exploiting spurious statistical cues in the datasets. To find out whether the models have ability to truly understand natural language semantics, we propose a new method that evaluates the model's ability to judge *logical equivalence* in a Nature Language Inference (NLI) task.

NLI is an important problem of Nature Language Understating, aiming at detecting inferential relationships between two sentences. Given a pair of textual inputs, A and A', we need to determine if A *entails* A', A *contradicts* A', or A and A' have no logical relationship (they are *neutral*). Also, if A implies A', and A' implies A, we can say A and A' are *logically equivalent*.

Take the following sentence pair as an example:

1. "If it is raining, then you take an umbrella."
2. "If you don't take umbrella, then it is not raining."

These two sentences are logically equivalent, because (1) can imply (2) and (2) can imply (1).

Logical equivalence can also be represented in *first-order logic rules*. In the above example, if we use P to represent statement "it is raining" and use Q to represent "you take an umbrella", the first-order logic rule is $(P \Rightarrow Q) \leftrightarrow (\neg Q \Rightarrow \neg P)$.

In this paper, we propose a new dataset containing logically equivalent sentences pairs, that enables us to evaluate the models' ability in distinguishing logical equivalence.

2 Related Work

Annotation artifacts have been common in large-scale NLI datasets (Gururangan et al., 2018), making NLI tasks trivial to solve. Salvatore et al. (2019) has argued that many existing NLI datasets do not properly address the structural feature as classification accuracies significantly greater than random can be achieved using a Bag-of-Words classifier. In order to assess neural language models' competence in logic, Salvatore et al. proposed to use formal logic for generating synthesized data. The authors used a template language composed of two basic entities, namely people and places, as well as three binary relations between the two entities. The template language is then realized through mapping

entities to nouns and relational operators and logical connectives to corresponding natural language forms. Six tasks were studied, including negation, boolean coordination, quantification, definite description, comparatives, and counting. Inspired by the method of the above work, Richardson et al. (2019) applied the data generation approach to semantic fragments targeting other semantic task such as monotonicity reasoning.

Our work follows these work as we also employ the method of generating data using logical templates, in order to target specific semantic phenomenon while having control over the complexity and diversity of the data. Our work is different in that instead of having several target phenomenon, we focus on learning logical equivalence from synthesized natural language data. Each of our template is motivated by existing transformation rules in logic such as DeMorgan's law, while the choices of templates are more arbitrary in the above work.

3 Methods

3.1 Experiment

The main task that we consider is equivalence identification, where we want to identify whether two given sentences s_1, s_2 are logically equivalent.

Since we focus on analyzing neural language models, especially BERT-based model (Devlin et al., 2019), we need a specific approach to evaluate BERT. Take $\text{BERT}_{\text{base}}$ as an example. Consider two sentences p_1, h_1 that we try to determine whether they are logically equivalent. As an analogy to the regular way for BERT performing pair sentence classification, we feed BERT with input in the form of "$[CLS]$ p_1 $[SEP]$ h_1 $[SEP]$", and use the first token's ([CLS]) representation as the whole pair of sentence's representation, denoted as e. Then we classify whether the pair is equivalent, contradicting, or neutral through a linear classifier over e. The objective is the cross-entropy loss in 3-class (`neutral`, `equivalence`, `contradiction`). As mentioned before, we claim that BERT has evaluated correctly that two sentences are logically equivalent only when it claims the entailment in both directions through this paradigm. We will treat the accuracy of this task as our evaluation metric. Also, we can identify the course of the mistake of these language models by tracing back which direction of entailment it predicted incorrectly.

	# pairs	Vocab.	Avg. len.
DeMorgan's	6000	384	11.95
Double negation	3000	376	6.87
Absorption	6000	384	12.97
Commutative	6000	376	9.52
Associative	6000	384	18.46
Idempotent	6000	374	7.17
Distributive	6000	387	24.49
Conditional	6000	374	10.64

Table 1: Statistics of generated data.

3.2 Data

The data is generated using the template language described in Salvatore et al. (2019). For each rule in the pair of laws described below, we define four statements A, B, C, D such that $A \leftrightarrow B$ and $C \leftrightarrow D$ are tautologies, and $A \leftrightarrow C$ and $B \leftrightarrow D$ are contradictions, as illustrated by Figure 2, where R refers to the governing law.

For example, we first define a set of first-order logic pairs where each form implies each other, thus logically equivalent. For example, for pair (A, B) where $A = (P \Rightarrow Q)$, and $B = (\neg P \vee Q)$, A implies B. Then, in the reverse direction, for pair (B, A), B also implies A. We thus say A and B are logically equivalent, or A $\leftrightarrow$ B. For pairs like (A, B) and (B, A), two cases of entailment in both directions result in a logical equivalence. For each A above, we also define some C such that $A \leftrightarrow C$ is a contradiction. For example, $C = (\neg P \wedge Q)$, forbidding the two statements from being logically equivalent.

Our templates are governed by pairs of transformation rules in logic, including `DeMorgan's law`, `double negation law`, `absorption law`, `communicative law`, `associative law`, `distributive law`, as well as an additional set of equivalences involving conditional statements.

For each of A, B, C, D, we define a corresponding first-order logical template, and a corresponding natural language template. The natural language this work studies is English. An example is given in Figure 1.

These templates are then realized as English sentences, where predicates p, q, r are drawn from a set of 278 predefined English verb phrases, and the variable x is drawn from a set of 50 common English names. Our predefined English verb phrases are generated through concatenating verbs from

Rules	Label	P logical Form	H logical Form	P NL Form	H NL Form
DeMorgan's	equivalence	¬ (p(x) ∧ q(x))	¬ p(x) ∨ ¬ q(x)	It is not true that Miranda$_x$ both drinks squash$_p$ and visits Luxembourg$_q$	Either Miranda$_x$ does not drinks squash$_p$ or Miranda$_x$ didn't visit Luxembourg$_q$
	neutral	¬ (p(x) ∧ q(x))	¬ p(x') ∨ ¬ q(x')	It is not true that Miranda$_x$ both drinks squash$_p$ and visits Luxembourg$_q$	Either Melody$_{x'}$ does not drinks squash$_p$ or Melody$_{x'}$ didn't visit Luxembourg$_q$
	contradiction	¬ (¬ p(x) ∨ ¬ q(x))	¬ p(x) ∨ ¬ q(x)	It's false that either Miranda$_x$ does not visits Luxembourg$_p$ or Miranda$_x$ does not visit Bucharest$_q$	Either Miranda$_x$ wasn't visiting Luxembourg$_p$ or Miranda$_x$ does not visit Bucharest$_q$

Figure 1: An example of data generation process using formal logic template.

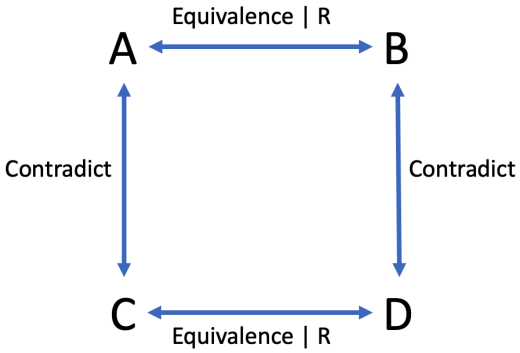

Figure 2: Unbiased data generation process using formal logic template. Given a formal expression A and an equivalence relation R (e.g.: DeMorgan's), we generate A's equivalent formal expression B according to R. Next, we attain formal expression C by taking A's negation and simplify it. Finally, we generate C's equivalent expression D using relation R. Then, for each pair of rule and A, we have 4 data instances: $(A = B)$, $(C = D)$, $(A \perp B)$, $(C \perp D)$ Note that in this design, each formal logic expression is possible to contradict or be equivalent to another expression, which eliminates the potential that the model makes decision by checking whether the premise/hypothesis is a specific formal logic expression. This generation format does decrease the bias in the premise/hypothesis, shown in Table 2, 3

{*eat, drink, go, play, do, visit*} and corresponding nouns in noun sets of musical instruments, food, drinks, sports, and cities. The statistics of our generated data is shown in Table 1.

Equivalence and contradiction data are manually defined in our 3-class classification templates, while neutral cases are generated for each equivalence case by randomly changing one of p, q, r, x to some other p', q', r', x' in either premise or hypothesis.

Since our data have a simple structure, in order to avoid introducing spurious cues, we experiment with two approaches to mitigating effects of potential cues, including paraphrasing and premise-hypothesis swapping. For paraphrasing, we use a script inspired by PARABANK by Hu et al. (2019) to dissimilate the syntactic structures while maintaining the semantic content of the data. For example, *it is not the case* → *it is false*. In this way, we introduce diversity of sentence structures while keeping content words as faithful as possible. As for swapping, for each transformation rule, half of the generated data have A, C as premises and B, D as hypotheses, while the other half have the other way around. We hope to eliminate potential cues from hypotheses, preventing the data from being solved trivially by looking at the hypothesis-only data, as previous work such as (Poliak et al., 2018) suggested.

3.3 Baseline Systems

We devise two baseline systems for this task to understand how difficult our generated dataset is to solve. One system leverages unlexicalized features described in Bowman et al. (2015), and the other system uses average word-embeddings as features inspired by Iyyer et al. (2015). Both systems use the support vector machine (SVM) as a classifier through `Scikit-learn` (Pedregosa et al., 2011) which is based on the `LIBSVM` (Chang and Lin, 2011) implementation.

Unlexicalized features: These are shallow features extracted from one pair of sentence, s_1 and s_2, without considering any other sentences in the corpus. According to Bowman et al., unlexicalized features include the following 3 types with 15 entries in total:

1. The BLEU score of s_1 with respect to s_2, using an n-gram length between 1 and 4 (4 entries).
2. The absolute length difference (by token) between s_1 and s_2 (1 entry).
3. The overlap between tokens in s_1 and s_2, both as an absolute count and a percentage of possible overlap, and both over all words and over just nouns, verbs, adjectives, and adverbs (10 entries).

Average embedding vectors: Similar to Iyyer et al.'s approach, we create vector representation of a sentence by averaging all word embedding vectors in that sentence, resulting in a 300-D vector

(word vectors come from the SpaCy (Honnibal and Montani, 2017) implementation). Then we concatenate the vectors of s_1 and s_2 to form a 600-D feature vector. Unlike Iyyer et al., we do not pass the resulting 600-D vector to any feed-forward layers, since we want such feature to stay "shallow".

4 Results

4.1 BERT analysis

Report results are shown in Table 2. The results show that the pre-trained BERT model's [CLS] token does not contain rich information about the equivalence relationship in the formal logic level, which is even worse than the performance of unlexicalized/lexicalized models. This suggests that the current unsupervised pre-trained paradigm fails to capture the equivalence relationship in logical level in its original knowledge aggregation. (The [CLS] token is regard as the token that aggregates the whole sequence information.)

To find out the reason of the poor performance of pre-trained BERT, we use the mean of the whole sequence's representations instead of the $[CLS]$ token to infer the equivalence relations. This setup can verify whether the pre-trained BERT does not capture the logically equivalent relationship or only the $[CLS]$ token does not capture such information. The result suggests that the whole sequence's information does capture more clues about inferring the logical equivalence. However, the performance is still much lower than that of the baselines. This strengthens the claim that the unsupervised pre-training of BERT does not help it learn logical equivalence in text.

4.2 Dataset analysis

According to Table 2, 3, without debiasing methods in section 3.2, the classifers can achieve high performance with only the hypothesis sentence (`h-only`). This suggests that there is significant structure bias in the hypothesis sentences since the system should check the pair of sentence for making predictions. Note that since the premise formal expression can be existed in any class (contradiction/equivalent/neutral), there's no evidence of the premise bias' existance. However, after debiasing, the low performance `h-only` model imply that the hypothesis now suffer from much less bias than before.

	P only	H only	P, H pair
*Rand. guess**	33.3	33.3	33.3
Baseline (feat.)	-	-	69.6
Baseline (embed)	25.7	40.8	71.6
$\text{BERT}_{\text{base}}$ (CLS)	34.9	38.0	40.0
+ finetune	37.7	44.2	**99.1**
$\text{BERT}_{\text{base}}$ (mean)	31.7	40.0	47.8
+ finetune	43.6	33.7	98.7

Table 2: % Test accuracy of 2 baseline systems and 2 ways of using BERT. Feature baseline is omitted because extracting features requires both p and h (P only: only using premise, H only: only using hypothesis).

	P only	H only	P, H pair
Baseline (feat.)	-	-	81.8
Baseline (embed)	34.4	65.4	86.1
$\text{BERT}_{\text{base}}$ (CLS)	35.7	39.0	41.1
+ finetune	36.4	**77.3**	99.7

Table 3: % Test accuracy before debiasing generated data. The data is simply generated by attaining the contradict and equivalent formal expression given a equivalence relation R described in Figure 1

5 Discussion

From the result, we can see that the test accuracy of pre-trained BERT is lower than any baseline, and is just a little higher than random guess. It means that the pre-trained BERT cannot capture the information of the logical equivalence.

We can see some pros and cons of our dataset from the result. The test accuracy of both the premise-only dataset and hypothesis-only dataset is low, no matter which model is being used. This means the dataset we created has little artifacts. However, BERT fine-tuned model has high accuracy, indicating that the complexity of our dataset is low.

By comparing the result before and after the debiasing process, we can see the accuracy on premise-only and hypothesis-only dataset have significant drop. It shows that the debiasing process is effective.

This dataset can also be used in other tasks. For future directions, we can use the dataset to evaluate other models and make comparison on BERT. Also, we can try to use this dataset to evaluate the hidden layers of models. We can visualize the dataflow in the hidden representations (aka. attention maps/internal layers) of models and try to figure out how the model can capture the logical

equivalence information.

Furthermore, it is better to increase the complexity of dataset, using more phrases and sentence structures. Also, the template-generated sentences do not fully resemble human language. We may work on making our data more diverse and more natural.

6 Conclusion

We proposed a new way to auto-generate a dataset that is capable to diagnose the ability of BERT about identifying the logical equivalence between two sentences. Instead of the usual template generation in (Salvatore et al., 2019; Richardson et al., 2019), we introduce the specific debiasing mechanism that does alleviate the structure bias in hypothesis sentences significantly. From the evaluation side, our result presents that the pre-trained BERT cannot capture the logical equivalence information without fine-tuning, which suggests the inefficiency of the pre-training paradigm towards learning such relationships. Moreover, the high accuracy of fine-tuned BERT indicates BERT is able to learn these relationships with supervised signals. As a follow-up, we will extend the current work towards three directions. First, we want to evaluate different models on this dataset and compare their ability to capture logical equivalence clues. Second, we can probe over the internal representations of BERT to understand how the model resolve the equivalence relation between sentences under the hood. Third, based on the high performance of fine-tuned BERT, we may look for ways to increase the complexity of the dataset to elevate the difficulty for models to find the structural clues.

References

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*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

J. Edward Hu, Rachel Rudinger, Matt Post, and Benjamin Van Durme. 2019. Parabank: Monolingual bitext generation and sentential paraphrasing via lexically-constrained neural machine translation. In *AAAI*.

Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. 2015. Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1681–1691, Beijing, China. Association for Computational Linguistics.

Timothy Niven and Hung-Yu Kao. 2019. Probing neural network comprehension of natural language arguments. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4658–4664, Florence, Italy. Association for Computational Linguistics.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191, New Orleans, Louisiana. Association for Computational Linguistics.

Kyle Richardson, Hai Hu, Lawrence S. Moss, and Ashish Sabharwal. 2019. Probing natural language inference models through semantic fragments. *ArXiv*, abs/1909.07521.

Felipe Salvatore, Marcelo Finger, and Roberto Hirata Jr. 2019. A logical-based corpus for cross-lingual evaluation. In *Proceedings of the 2nd*

Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019), pages 22–30, Hong Kong, China. Association for Computational Linguistics.