How Fast can BERT Learn Simple Natural Language Inference?

Yi-Chung Lin and **Keh-Yih Su**
Institute of Information Science
Academia Sinica, Taiwan
{lyc,kysu}@iis.sinica.edu.tw

Abstract

This paper empirically studies whether BERT can really learn to conduct natural language inference (NLI) without utilizing hidden dataset bias; and how efficiently it can learn if it could. This is done via creating a simple entailment judgment case which involves only binary predicates in plain English. The results show that the learning process of BERT is very slow. However, the efficiency of learning can be greatly improved (data reduction by a factor of 1,500) if task-related features are added. This suggests that domain knowledge greatly helps when conducting NLI with neural networks.

1 Introduction

Entailment judgment (Dagan et al., 2006; Marelli et al., 2014a) is a common test for *natural language inference* (NLI) (Camburu et al., 2018; Conneau et al., 2018) as it possesses the simplest form in related tasks such as question and answering (Bowman and Zhu, 2019). Also, SNLI dataset (Bowman et al., 2015) is frequently adopted for NLI evaluation because it is the first corpus to show the power of neural networks for the task that specifically targets NLI.

Recently, BERT (Devlin et al., 2019), XLNet (Yang et al., 2019), and RoBERTa (Liu et al., 2019) have all shown excellent performances on SNLI, surpassing even human performance (Gong et al., 2017). However, Tsuchiya (2018) and Gururangan et al. (2018) show that SNLI contains *hidden bias*. Also, deep neural networks (DNNs) have been shown to predominantly capture the *statistical irregularities* that go unnoticed by humans (Poliak et al., 2018); this surprisingly yields 69% accuracy on SNLI without being provided the associated *premise* (i.e., the required *supporting evidence*). Furthermore, some studies (Naik et al., 2018; McCoy et al., 2019; Jiang and Marneffe, 2019) have shown that BERT mainly conducts NLI with surface clues/patterns, but not those clues actually adopted by humans.

To the best of our knowledge, none of previous studies carefully removes the dataset bias, and then investigates how efficiently that BERT could learn NLI if the surface clues are *completely* removed. In this paper, we empirically study whether BERT is capable of learning NLI without surface clues/bias appearing in the dataset; and if it is, whether it can learn NLI efficiently. We carefully created absolutely unbiased datasets, in which the premises and hypotheses simply describe the relative position of two objects in plain English. That is, given a premise "*John is on the left side of Mary*" (abbreviated as a predicate "left(John, Mary)" from now on), the hypotheses predicates "left(John, Mary)", "left(Mary, John)" and "left(John, Helen)" should be labeled as *entailment*, *contradictory* and *neutral* respectively. Our experiment results show BERT is very slow in learning this simple NLI.

We then further study if the learning efficiency can be improved with domain knowledge (Gülçehre and Bengio, 2016). Inspired by Chen et al. (2017), we think whether two entities are *exactly the same* is quite crucial in making the above NLI. So, the task-related features, such as whether the first/second argument of the premise predicate exactly matches that of hypothesis, are fed into BERT. The obtained results show that such task-related features are able to greatly benefit BERT (reducing the data needed by a factor of 1,500), which is important as it is difficult to acquire enough data in many real-world applications.

Our main contributions are: (1) We are the first to quantitatively study how efficiently BERT can learn to conduct NLI without available surface clues/bias. (2) We design experiments to completely eliminate hidden bias while evaluating

Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics, pages 626–633
April 19 - 23, 2021. ©2021 Association for Computational Linguistics

the inference capability of BERT. (3) We show that adding task-related features greatly enhances the learning efficiency of BERT.

2 Teaching BERT Binary Predicates

The sentence "*John is on the left side of Mary*" describes a positional relation between two people. For conciseness, it will be denoted by a binary predicate "left(John,Mary)" from now on, where "left" is the predicate name, "John" is the first argument and "Mary" is the second argument. We seek to determine how much data are required to teach BERT to truly understand this simple binary predicate (i.e., premise) and correctly judge that the hypothesis "left(Mary,John)" is contradictory and that the hypothesis "left(John,Helen)" is neutral. Besides, we also seek to determine whether BERT is also able to learn the antonymous predicate "right(·,·)", and judge that the hypothesis "right(Mary,John)" is entailed by the above premise.

2.1 Entity Names and Datasets

The arguments of the above binary predicates "left(·,·)" and "right(·,·)" actually can be the names of any objects. However, in this paper, we simply trained BERT with *personal names*. To avoid dividing a personal name into sub-words, we collected 1,696 male and female first names which appear in the vocabulary of the pre-trained "BERT-Base, Uncased" model (Devlin et al., 2019). These names were randomly partitioned into three sets: $\boldsymbol{\eta}_{\mathbf{T}}$, $\boldsymbol{\eta}_{\mathbf{V}}$ and $\boldsymbol{\eta}_{\mathbf{E}}$. The subscripts **T**, **V** and **E** indicate these name sets will be used for **T**raining, **V**alidation and **E**valuation respectively. Sets $\boldsymbol{\eta}_{\mathbf{T}}$ and $\boldsymbol{\eta}_{\mathbf{V}}$, which consist of 1,356 and 170 names respectively, are used to generate the training and validation datasets for fine-tuning BERT; and Set $\boldsymbol{\eta}_{\mathbf{E}}$, consisting of 170 names, is used to generate a dataset for evaluating the performance of BERT in understanding the predicates with personal names.

Furthermore, we also seek to ascertain whether the BERT model trained by the predicates with personal names also understands the predicates with names of other object types. Therefore, in addition to personal names, we also collected 30 common fruit and vegetable names to create an additional set $\mathbf{f}_{\mathbf{E}}$, which will be used to generate another dataset for performance evaluation.

We conduct a number of experiments on *recognizing textual entailment* (RTE) (Dagan et al., 2006; Marelli et al., 2014a) to study the learning curves of BERT in understanding binary predicates. In each experiment, four datasets—**Name-T**, **Name-V**, **Name-E** and **Fruit-E**—are created by filling experiment-specific templates with names randomly chosen from $\boldsymbol{\eta}_{\mathbf{T}}$, $\boldsymbol{\eta}_{\mathbf{V}}$, $\boldsymbol{\eta}_{\mathbf{E}}$ and $\mathbf{f}_{\mathbf{E}}$ respectively. The training set **Name-T** and the validation set **Name-V** are used to fine-tune the BERT model. The other two sets (**Name-E** and **Fruit-E**) are test sets. They are used to assess the performances of the BERT model. Each dataset is generated via iteratively and sequentially adding one entailment example, one contradictory example, and one neutral example until it reaches the desired size. Thus a dataset of a size 100 will consist of 34 entailment examples, 33 contradictory examples, and 33 neutral examples. The experiment-specific templates are described in the following sections.

2.2 One Simple Binary Predicate

We first conducted *EXP-SP* (SP: **S**imple **P**redicate) experiment to show if BERT can be taught to understand the simple binary predicate "left(·,·)". In this experiment, a template has the form "*premise* [s] *hypothesis*", where "[s]" is a separator token between the premise and the hypothesis. The templates are partitioned into entailment, contradictory, and neutral template sets. The entailment template set has only one template "left(x, y) [s] left(x, y)", where "left(x, y)" represents the token sequence "x is on the left side of y" and the variables x and y indicate the names to be filled in. Likewise, the contradictory template set has only one template "left(x, y) [s] left(y, x)". However, the neutral template set consists of four templates:

"left(x, y) [s] left(x, z)",
"left(x, y) [s] left(z, x)",
"left(x, y) [s] left(y, z)",
"left(x, y) [s] left(z, y)",

where the variable z indicates a name to be filled in. Names and templates were randomly selected during dataset creation. For example, when generating a neutral example for **Name-T**, we randomly chose one template from the neutral template set and randomly chose three distinct names (for variables x, y, z) from name set $\boldsymbol{\eta}_{\mathbf{T}}$. Obviously, the generated datasets do not have any hidden bias, as all examples share the same token sequence except the argument tokens which are randomly chosen. Therefore, no annotation

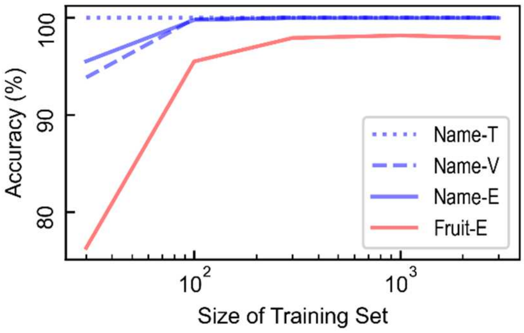

Figure 1: RTE performances of EXP-SP.

artifacts (Gururangan et al., 2018) in the data can provide hints for predicting final inference answers.

To quantitatively study the learning efficiency, we generated various **Name-T** sets with different sizes to study how much data would be required to teach BERT to understand the simple binary predicate "left(·,·)". We set **Name-V**, **Name-E**, and **Fruit-E** to 1,000 examples each. The solid lines in Figure 1 show that the accuracies of BERT on both test sets (i.e., **Name-E** and **Fruit-E**) increase when the training set size increases. However, even if we train BERT with 3,000 examples, it still cannot achieve 100% accuracy on the test sets. In fact, all sentences in this experiment have the form "*x is on the left side of y*", where x and y are object names. Therefore, only 6 different words could appear in the context of object names. In other words, given 3,000 examples, BERT still cannot fully understand the meaning of the context "*is on the left side of*".

Furthermore, training BERT to reach over 99% accuracy on **Name-E** requires 100 training examples. However, even given 30 times the training data, BERT is still unable to achieve 99% accuracy on **Fruit-E**. That is, BERT is not able to well generalize what it has learned from the examples with person names to the examples with fruit and vegetable names.

2.3 One Antonymous Predicate

Antonyms are frequently used in natural language and play an important role in natural language inference. Given the premise "*John is on the left side of Mary*", we can easily infer that the hypothesis "*Mary is on the right side of John*" is entailed by the premise, and that the hypothesis "*John is on the right side of Mary*" contradicts to the premise. This inference is easy for humans; but is it also easy for BERT? Therefore, we conducted another test named *EXP-AP* (AP: **A**ntonymous **P**redicate), a more complicated RTE experiment in which we added the antonymous predicate "right(·,·)".

In this experiment, the entailment template set consists of the following four templates:

"left(x,y) [s] left(x,y)",
"left(x,y) [s] right(y,x)",
"right(x,y) [s] right(x,y)",
"right(x,y) [s] left(y,x)".

The contradictory template set consists of the following four templates:

"left(x,y) [s] left(y,x)",
"left(x,y) [s] right(x,y)",
"right(x,y) [s] right(y,x)",
"right(x,y) [s] left(x,y)".

Likewise, the neutral template set consists of 16 templates. In brief, adding one antonymous predicate "right(·,·)" enlarges the number of all possible templates for dataset generation from 6 to 24.

The solid lines in Figure 2 are the accuracies on the test sets **Name-E** and **Fruit-E** in EXP-AP. For ease of comparison, we also plot the EXP-SP counterpart accuracies with dotted lines. Although EXP-AP uses only four times as many templates as EXP-SP uses, we must provide more than 30 times the training data (from 100 to 3,000) for BERT to reach 99% accuracy on **Name-E**. That is, by adding a single antonymous predicate, the RTE task of EXP-AP becomes 30 times harder than that of EXP-SP. It seems teaching BERT to "almost understand" two simple binary predicates requires more than 3,000 examples. This result could be also interpreted in another view. In the EXP-AP, the context of object names in each sentence is either "*is on the left side of*" or "*is on the right side of*". BERT requires 3,000 examples to learn the meanings of the 7 different words that appear in these two contexts. This represents a quite inefficient learning curve.

3 Incorporating Human Knowledge

The previous section showed that many training examples are needed to teach BERT for understanding two binary predicates "left(·,·)" and "right(·,·)" in a simple RTE task. It thus naturally leads to a conjecture that training BERT to understand more complicated predicates would very likely require infeasible amount of training

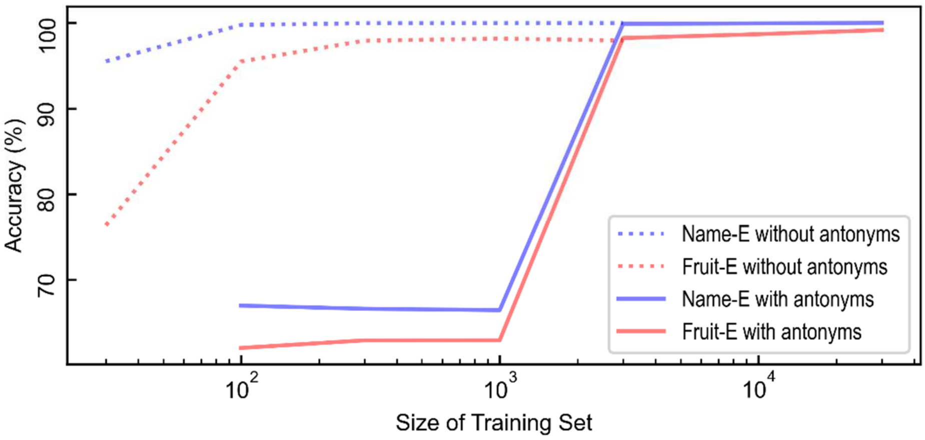

Figure 2: RTE performances of EXP-SP (dotted lines) and EXP-AP (solid lines).

data. One possible solution is to improve the learning curve of BERT by feeding it useful/obvious features that have been identified in human/domain knowledge.

3.1 Simple Features (SF)

Accordingly, we conducted an experiment *EXP-SF*, in which we appended features to the input tokens of EXP-AP to improve the learning curve. In this experiment, a template has the form "*premise* [s] *hypothesis* [s] *features*". Obviously, humans directly compare the predicate name and the predicate arguments in the premise against their counterparts in the hypothesis. Perhaps this knowledge about which two fields should be compared would be helpful when training BERT on the EXP-AP task. Let N_{P} and N_{H} indicate the predicate names in the premise and the hypothesis respectively; also, let $A_{i,\mathrm{P}}$ and $A_{i,\mathrm{H}}$ indicate the i-th predicate arguments in the premise and the hypothesis respectively. In EXP-SF, every example in the datasets of EXP-AP will be augmented by the following three indicator features:

$$f_1 = I(N_{\mathrm{P}} = N_{\mathrm{H}}),$$
$$f_2 = I\left(A_{1,\mathrm{P}} = A_{1,\mathrm{H}}\right),$$
$$f_3 = I\left(A_{2,\mathrm{P}} = A_{2,\mathrm{H}}\right).$$

Here the indicator function $I(x)$ returns the word "*true*" if x is true and "*false*" if it is not. EXP-SF templates thus could be directly transformed from the corresponding EXP-AP templates. To illustrate this transformation, we show an entailment template, a contradictory template and a neutral template in EXP-SF as follows:

"left(x, y) [s] right(y, x) [s] false false false",
"left(x, y) [s] left(y, x) [s] true false false",
"left(x, y) [s] right(x, z) [s] false true false".

The dashed lines in Figure 3 are the accuracies on the test sets **Name-E** and **Fruit-E** in EXP-SF. For ease of comparison, we also plot the counterpart accuracies of the baseline (i.e., EXP-AP) with dotted lines. The fact that the dashed lines lie far on the left side of the dotted lines indicates that much fewer training examples are required to train BERT after adding these three simple features [1]. This represents a greatly improved learning curve. Specifically, given merely 100 EXP-SF examples, BERT achieves over 99% accuracy on **Name-E**; in contrast, BERT requires 3,000 examples to surpass 99% accuracy on **Name-E** in EXP-AP. This represents a greatly improved learning curve.

3.2 Discriminant Features (DF)

In the previous experiment EXP-SF, features f_2 and f_3 do not precisely indicate the situation in which the arguments in the premise match the arguments in the hypothesis after swapping. For example, for both "left(x, y) [s] left(y, x)" and "left(x, y) [s] left(z, x)", features f_2 and f_3 are all *false* in EXP-SF. However, the former is a contradictory case and the latter is a neutral case. We thus conducted the last experiment named *EXP-DF*, in which we replaced f_2 and f_3 with a

[1] Note that it does not matter which words were chosen to represent the values of features. We had randomly selected *fortification* and *mississippi* from BERT's vocabulary to replace the words *true* and *false*. The experimental results were similar to those in Figure 3. In other words, the initial embeddings of the feature values are not crucial.

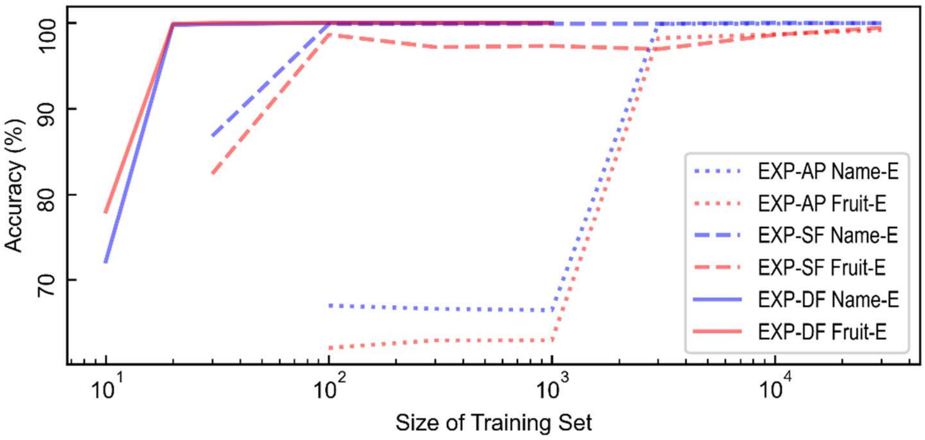

Figure 3: RTE performances of EXP-AP (dotted lines), EXP-SF (dashed lines) and EXP-DF (solid lines). More detailed data values are listed in Appendix.

new discriminant feature f_2'. Words *true*, *false* and *fuzzy* were used to indicate the three possible values of this new feature as follows:

$$f_2' = \begin{cases} true, & A_{1,\mathrm{P}} = A_{1,\mathrm{H}} \wedge A_{2,\mathrm{P}} = A_{2,\mathrm{H}}; \\ false, & A_{1,\mathrm{P}} = A_{2,\mathrm{H}} \wedge A_{2,\mathrm{P}} = A_{1,\mathrm{H}}; \\ fuzzy, & \text{otherwise.} \end{cases}$$

The solid lines in Figure 3 show that the learning curve of BERT is further improved after adopting this discriminant feature. Given only 20 EXP-DF examples, BERT achieves 99.9% accuracy on **Fruit-E**. However, in EXP-AP, 1,500 times amount of data (i.e., 30,000 examples) is needed for reaching 99.1% accuracy on **Fruit-E**.

4 Related Work

Dagan et al. (2006) first initiated the task of recognizing textual entailment about fifteen years ago. This task continued until 2011 (Bentivogli et al., 2011). Afterwards, conducting inference with BERT was studied in (Clark et al., 2019; Zellers et al., 2019; Tenney et al., 2019; Aken et al., 2019; Coenen et al., 2019; Michel et al., 2019).

On the other hand, various corpora have been created for conducting NLI for different purposes: SICK (Marelli et al., 2014b), SNLI (Bowman et al., 2015), MNLI (Willams et al., 2018), MPE (Lai et al., 2017), JOCI (Zhang et al., 2017), XNLI (Conneau et al., 2018), and SciTail (Khot et al., 2018). Corpora such as HOTPOTQA (Yang et al., 2018), Breaking-NLI (Glockner et al., 2018), CommonSense QA (Talmor et al., 2019), DROP (Dua et al., 2019) and ROPES (Lin et al., 2019) have also been created recently to evaluate more diverse and difficult NLI cases. However, all those corpora are not created for keeping the data absolutely unbiased. We believe using bias-free simple binary predicates would be more suitable to assess the true inference capability of BERT (or even other DNNs).

Recently, Ribeiro et al. (2020) proposed a new evaluation methodology, named CheckList, to check general linguistic capabilities of a given NLI model. They generated a large number of diverse test cases to identify the critical failures hidden behind state-of-art models. In contrast, our work mainly targeted the learning efficiency of BERT on various unbiased datasets. Besides, we also studied how adding useful domain-specific features would affect the learning curve of BERT.

5 Conclusion

This paper is the first quantitative study on whether BERT could really learn to conduct NLI without implicitly utilizing hidden dataset bias, and how quickly it does so if it could. We conduct experiments to evaluate the capability of BERT on making inference without hidden bias, and show that BERT learns NLI inefficiently even for a simple case. We further add task-related features to greatly enhance BERT's learning efficiency. As a result, it suggests that domain knowledge may be essential in conducting NLI with neural networks (at least for BERT).

References

Betty van Aken, Benjamin Winter, Alexander Löser, and Felix A. Gers. 2019. How Does BERT Answer Questions: A Layer-Wise Analysis of Transformer

Representations. *Computing Research Repository,* arXiv:1909.04925.

Luisa Bentivogli, Peter Clark, Ido Dagan, and Danilo Giampiccolo. 2011. The Seventh PASCAL Recognizing Textual Entailment Challenge. In *Proceedings of the 4th Text Analysis Conference*. Gaithersburg: National Institute of Standards and Technology, pages 1–16.

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

Samuel R. Bowman and Xiandan Zhu. 2019 Deep Learning for Natural Language Inference. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials*.

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-SNLI: Natural Language Inference with Natural Language Explanations. In *Proceedings of the 32nd Conference on Neural Information Processing Systems (NeurIPS 2018)*.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to Answer Open-Domain Questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pages 1870–1879.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What Does BERT Look At? An Analysis of BERT's Attention. *Computing Research Repository,* arXiv:1906.04341.

Andy Coenen, Emily Reif, Ann Yuan, Been Kim, Adam Pearce, Fernanda Viégas, and Martin Wattenberg. 2019. Visualizing and Measuring the Geometry of BERT. *Computing Research Repository,* arXiv:1906.02715.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating Cross-lingual Sentence Representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 2475–2485.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The PASCAL Recognising Textual Entailment Challenge. In *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, Springer, pages 177–190.

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*. Association for Computational Linguistics, pages 4171–4186.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A Reading Comprehension Benchmark Requiring Discrete Reasoning Over Paragraphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, pages 2368–2378.

Max Glockner, Vered Shwartz and Yoav Goldberg. 2018. Breaking NLI Systems with Sentences that Require Simple Lexical Inferences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pages 650–655.

Yichen Gong, Heng Luo and Jian Zhang. 2017. Natural Language Inference Over Interaction Space. *Computing Research Repository,* arXiv:1709.04348.

Çağlar Gülçehre and Yoshua Bengio. 2016. Knowledge Matters: Importance of Prior Information for Optimization. *Journal of Machine Learning Research*, 17(8): 1-32.

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*. Association for Computational Linguistics, pages 107–112.

Nanjiang Jiang and Marie-Catherine de Marneffe, 2019. Evaluating BERT for natural language inference: A case study on the Commitment Bank. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing.* Association for Computational Linguistics, pages 6086–6091.

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2018. SCITAIL: A Textual Entailment Dataset from Science Question Answering. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI-18)*. Association for the Advancement of Artificial Intelligence, pages 5189–5197.

Alice Lai, Yonatan Bisk, and Julia Hockenmaier. 2017. Natural Language Inference from Multiple Premises. In *Proceedings of the 8th International Joint Conference on Natural Language Processing*. Asian Federation of Natural Language Processing, pages 100–109.

Kevin Lin, Oyvind Tafjord, Peter Clark, and Matt Gardner. 2019. Reasoning Over Paragraph Effects in Situations. *Computing Research Repository,* arXiv:1908.05852.

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. *Computing Research Repository,* arXiv:1907.11692.

Marco Marelli, Luisa Bentivogli, Marco Baroni, Raffaella Bernardi, Stefano Menini and Roberto Zamparelli. 2014a. SemEval-2014 Task 1: Evaluation of Compositional Distributional Semantic Models on Full Sentences through Semantic Relatedness and Textual Entailment. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*. Association for Computational Linguistics, pages 1–8.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi and Roberto Zamparelli. 2014b. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the 9th International Conference on Language Resources and Evaluation*. European Language Resources Association (ELRA), pages 216–223.

R. Thomas McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the Wrong Reasons: Diagnosing Syntactic Heuristics in Natural Language Inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pages 3428–3448.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are Sixteen Heads Really Better than One? In *Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*. arXiv:1905.10650.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 2018. Stress Test Evaluation for Natural Language Inference. In *Proceedings of the 27th International Conference on Computational Linguistics*. Association for Compu-tational Linguistics, pages 2340–2353.

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*. Association for Computational Linguistics, pages 180–191.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin and Sameer Singh. 2020. Beyond Accuracy: Behavioral Testing of NLP Models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pages 4902–4912.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CoomonSenseQA: A Question Answering Challenge Targeting Commonsense Knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, pages 4149–4158.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT Rediscovers the Classical NLP Pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pages 4593–4601.

Masatoshi Tsuchiya. 2018. Performance Impact Caused by Hidden Bias of Training Data for Recognizing Textual Entailment. In *Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC 2018).* https://www.aclweb.org/anthology/L18-1239

Adina Willams, 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*. Association for Computational Linguistics, pages 1112–1122.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HOTPOTQA: A Dataset for Diverse Explainable Multi-hop Question Answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 2369–2380.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. *Computing Research Repository,* arXiv:1906.08237.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a Machine Really Finish Your Sentence? In

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, pages 4791–4800.

Sheng Zhang, Rachel Rudinger, Kevin Duh, and Benjamin Van Durme. 2017. Ordinal Commonsense Inference. Transactions of the Association for Computational Linguistics, 5:379–395.

Appendix

To reduce the performance assessment error, each accuracy reported in this paper is the mean of accuracies obtained from multiple simulations. Each simulation uses a unique random seed to fine-tune the BERT model. The accuracies plotted in the figures for EXP-AP, EXP-SF and EXP-DF are listed in the following table, where μ denotes the accuracy mean and σ denotes the standard deviation of μ.

Training Set Size	Test Set Accuracy (%)			
	Name-E		**Fruit-E**	
	μ	σ	μ	σ
EXP-AP				
100	67.0	0.6	62.1	1.8
300	66.6	0.3	62.9	0.9
1000	66.5	0.4	62.9	1.4
3000	99.9	0.0	98.2	0.3
10000	99.9	0.0	98.7	0.2
30000	100.0	0.0	99.1	0.1
EXP-SF				
30	86.8	0.9	82.4	3.1
100	99.9	0.1	98.6	0.4
300	99.9	0.0	97.2	0.4
1000	99.9	0.0	97.3	0.6
3000	99.9	0.0	96.9	0.5
10000	100.0	0.0	98.6	0.9
30000	100.0	0.0	99.4	0.1
EXP-DF				
10	72.1	1.3	78.0	3.4
20	99.8	0.2	99.9	0.1
30	99.9	0.1	100.0	0.0
100	100.0	0.0	100.0	0.0
300	100.0	0.0	100.0	0.0
1000	100.0	0.0	100.0	0.0