Evaluating Paraphrastic Robustness in Textual Entailment Models

Dhruv Verma **Yash Kumar Lal** **Shreyashee Sinha**
Stony Brook University
{dhverma,ylal,shrsinha}@cs.stonybrook.edu

Benjamin Van Durme
Johns Hopkins University
vandurme@jhu.edu

Adam Poliak
Bryn Mawr College
apoliak@brynmawr.edu

Abstract

We present $\hat{P}aRTE$, a collection of 1,126 pairs of Recognizing Textual Entailment (RTE) examples to evaluate whether models are robust to paraphrasing. We posit that if RTE models understand language, their predictions should be consistent across inputs that share the same meaning. We use the evaluation set to determine if RTE models' predictions change when examples are paraphrased. In our experiments, contemporary models change their predictions on 8-16% of paraphrased examples, indicating that there is still room for improvement.

1 Introduction

Recognizing Textual Entailment (RTE), the task of predicting whether one sentence (*hypothesis*) would likely be implied by another (*premise*), is central to natural language understanding (NLU; Dagan et al., 2005), as this task captures "all manners of linguistic phenomena and broad variability of semantic expression" (MacCartney, 2009). If an RTE model has a sufficiently high *capacity for reliable, robust inference necessary for full NLU* (MacCartney, 2009), then the model's predictions should be consistent across paraphrased examples.

We introduce $\hat{P}aRTE$, a test set to evaluate how *reliable* and *robust* models are to paraphrases (Table 1 includes an example). The test set consists of examples from the Pascal RTE1-3 challenges (Dagan et al., 2006; Bar-Haim et al., 2006; Giampiccolo et al., 2007) rewritten with a lexical rewriter and manually verified to preserve the meaning and label of the original RTE sentence-pair. We use this evaluation set to determine whether models change their predictions when examples are paraphrased.

While this may not be a sufficient test to determine whether RTE models *fully understand* language, as there are many semantic phenomena that RTE models should capture (Cooper et al., 1996; Naik et al., 2018), it is *necessary* that any NLU system be robust to paraphrases.

P	The cost of security when world leaders gather near Auchterarder for next year 's G8 summit, is expected to top $150 million.
P'	The cost of security when world leaders meet for the G8 summit near Auchterarder next year will top $150 million.
H	More than $150 million will be probably spent for security at next year's G8 summit.
H'	At the G8 summit next year more than $150 million will likely be spent on security at the event.

Table 1: An original and paraphrased RTE example. The top represents an original premise (P) and its paraphrase (P'). The bottom depicts an original hypothesis (H) and its paraphrase (H'). A model robust to paraphrases should have consistent predictions across the following pairs: P-H, P'-H, P-H', and P'-H'.

Our experiments indicate that contemporary models are robust to paraphrases as their predictions do not change on the overwhelmingly large majority of examples that are paraphrased. However, our analyses temper this claim as models are more likely to change their predictions when both the premise and hypothesis are phrased compared to when just one of the sentences is rewritten. We release $\hat{P}aRTE$[1] to encourage others to evaluate how well their models perform when RTE examples are paraphrased.

2 Related Work

With the vast adoption of human language technology (HLT), systems must understand when different expressions convey the same meaning (paraphrase) and support the same inferences (entailment). Paraphrasing and entailment are closely connected as the former is a special case of the latter where two sentences entail each other (Nevěřilová, 2014; Fonseca and Aluísio, 2015; Víta, 2015; Ravichander et al., 2022). Para-

[1] https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/HLMI23

phrasing has been used to improve RTE predictions (Bosma and Callison-Burch, 2006; Sun et al., 2021) and RTE has been used for paraphrase identification (Seethamol and Manju, 2017) and generation (Arora et al., 2022). Furthermore, both phenomena are key to NLU (Androutsopoulos and Malakasiotis, 2010) and work such as Zhao et al. (2018); Hu et al. (2019) have explored rewriting RTE examples to create more robust models.

We follow a long tradition of evaluating linguistic phenomena captured in RTE models (Cooper et al., 1996). Recent tests focus on evaluating how well contemporary RTE models capture phenomena such as monotonicity (Yanaka et al., 2019a,b), verb veridicality (Ross and Pavlick, 2019; Yanaka et al., 2021), presuppositions (Parrish et al., 2021) implicatures (Jeretic et al., 2020), basic logic (Richardson et al., 2020; Shi et al., 2021), figurative language (Chakrabarty et al., 2021), and others (Naik et al., 2018; Poliak et al., 2018a; Vashishtha et al., 2020). Unlike many of those works that evaluate models' accuracy on examples that target specific phenomena, we use a contrastive approach (Prabhakaran et al., 2019; Gardner et al., 2020) to determine whether RTE models' predictions change when examples are paraphrased.

3 $\hat{P}aRTE$

To explore whether these RTE models are robust to paraphrases, we create $\hat{P}aRTE$, a modified version of the Pascal RTE1-3 challenges (Dagan et al., 2005; Bar-Haim et al., 2006; Giampiccolo et al., 2007). $\hat{P}aRTE$ contains 1,126 examples of an original unmodified RTE sentence-pair grouped with a sentence-pair with a modified premise, hypothesis, or both. We use the examples in RTE1-3 to create our test set, as opposed to other RTE datasets due to its long-standing history.

3.1 Paraphrase Generation & Verification

For each RTE premise-hypothesis pair (P-H), we created three paraphrased premises (P') and hypotheses (H') using a T5-based paraphraser[2] fine-tuned on the Google PAWS dataset (Zhang et al., 2019). To ensure lexically diverse paraphrases, we filter out any paraphrases that have high lexical overlap with the original sentences using Jaccard index threshold of 0.75. Out of 14,400 generated sentences, 2,449 remained - 956 paraphrased premises (P') and 1,493 paraphrased hypotheses (H'). Next, we retained 550 paraphrased premises and 800 paraphrased hypotheses paraphrases that crowdsource workers identified as grammatical and similar in meaning to the original sentences.[3] We include a grammatical check since an existing RTE evaluation set focused on paraphrases (White et al., 2017) contains hypothesis-only biases related to grammaticality (Poliak et al., 2018b).

If at least one P' or one H' passes this filtering process, we retain the original RTE example and pair it with a corresponding paraphrased example (i.e. P'-H', P'-H, or P-H'). In the case where more than one P' or H' passes the filtering, we retained the P' or H' that crowdsource workers deemed most similar to the original sentence. Out of the original 2,400 RTE test pairs, we retain 914 pairs with a high-quality P' or H', resulting in 1,178 original and paraphrased RTE pairs.[4]

3.2 Overcoming Semantic Variability

MacCartney (2009) argues that in addition to being *reliable* and *robust*, RTE models must deal with the *broad variability of semantic expression*. In other words, though two sentences may be semantically congruent, it is possible that small variations in a paraphrased sentence contain enough semantic variability to change what would likely, or not likely be inferred from the sentence. Despite all P' and H' being deemed to be semantically congruent with their corresponding original sentences, the semantic variability of paraphrases might change whether H or H' can be inferred from P' or P.

Therefore, propagating an RTE label from an original sentence pair to a modified sentence pair might be inappropriate. We manually determined that this issue occurs in just 52 (4%) examples, and retained 1,126 examples. This ensures an evaluation set of high-quality examples that can be used to determine whether models are sensitive to paraphrases and change their prediction on paraphrased examples. Our dataset contains 402 examples with just a paraphrased premise P', 602 with just a paraphrased hypothesis H', and 122 with both a paraphrased premise and hypothesis.

[2]We manually verified the quality of this paraphraser. See Appendix B for more details.

[3]See Appendix B for a detailed description of this filtering process, including annotation guidelines.

[4]415 pairs where the premise is paraphrased, 631 pairs where the hypothesis is paraphrased, and 132 pairs where both are paraphrased.

Model \ Testset	MNLI	RTE	$\hat{P}aRTE$	% Δ $\hat{P}aRTE$
BoW	67.97	53.99	54.70	15.27
BiLSTM	66.68	51.59	51.24	16.69
BERT	90.04	72.11	72.55	9.50
RoBERTa	92.68	83.83	82.59	7.99
GPT-3	-	80.90	79.12	10.12

Table 2: Each row represents a model. The columns MNLI, RTE, $\hat{P}aRTE$ report the model's accuracy on those test sets. The last column (% Δ $\hat{P}aRTE$) reports the percentage of examples where the model changed its prediction.

4 Experimental Setup

We explore models built upon three different classes of sentence encoders: bag of words (BoW), LSTMs, and Transformers. Our BoW model represents premises and hypotheses as an average of their tokens' 300 dimensional GloVe embeddings (Pennington et al., 2014b). The concatenation of these representations is fed to an MLP with two hidden layers. For the BiLSTM model, we represent tokens with GloVe embeddings, extract sentence representations using max-pooling, and pass concatenated sentence representations to an MLP with two hidden layers.

Our transformer-based models are pre-trained BERT (Devlin et al., 2019) and Roberta (Liu et al., 2020) encoders with an MLP attached to the final layer. Additionally, we use GPT-3 in a zero-shot setting where we ask it to label the relationship between a premise and hypothesis.[5]

The RTE training sets do not contain enough examples to train deep learning models with a large number of parameters. We follow the common practice of training models on MNLI and using our test set to evaluate how well they capture a specific phenomenon related to NLU. During testing, we map the MNLI 'contradiction' and 'neutral' labels to the 'not-entailed' label in RTE, following common practice (Wang et al., 2018; Yin et al., 2019; Ma et al., 2021; Utama et al., 2022, *inter ailia*).

5 Results

Table 2 report the results. The RTE and $\hat{P}aRTE$ columns respectively report the models' accuracy on the 1,126 unmodified and paraphrased sentence pairs.[6] Comparing the difference in accuracy between unmodified and paraphrased examples can be misleading. If the number of times a model changes a correct prediction is close to the number of times it changes an incorrect prediction, then the accuracy will hardly change. Figure 1 demonstrates why the accuracies do not change by much when models' predictions change on paraphrased examples. Furthermore, if a model is robust to paraphrases, then it should not change its predictions when an example is paraphrased, even if the prediction on the original unmodified example was incorrect. Hence, our test statistic is the percentage of examples where a model's predictions change (% Δ $\hat{P}aRTE$ column in Table 2) rather than a change in accuracy.

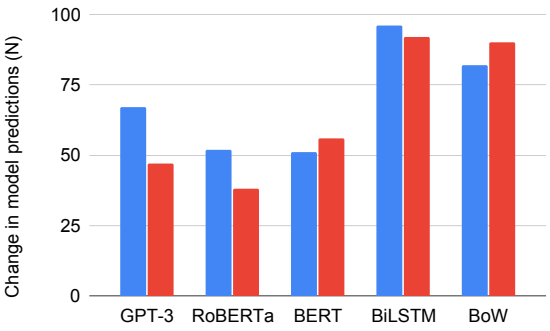

Figure 1: Number of times a model changes its predictions from correct to incorrect (left blue bar) or incorrect to correct (right red bar).

Compared to the Transformer based models, the BoW and BiLSTM models seem to be more sensitive, and less robust to paraphrasing, as they change their predictions on 15.27% and 16.69% respectively of the 1,126 examples. However, this might be associated with how word embedding models only just outperform random guesses in and perform much worse on RTE compared to the Transformer models.

[5] See Appendix A for more details, including hyperparameters, model sizes, and GPT-3 prompt design and configurations. Our code is available at https://github.com/stonybrooknlp/parte

[6] Although there are just 914 unmodified sentence pairs, for the sake of a head-to-head comparison, we retain all instances of the unmodified sentence pairs when computing accuracy.

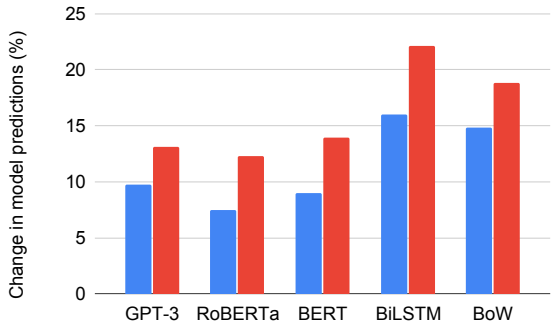

Figure 2: Percentage of examples with one paraphrased sentence (left blue bar) or two paraphrased sentences (right red bar) where models' predictions change.

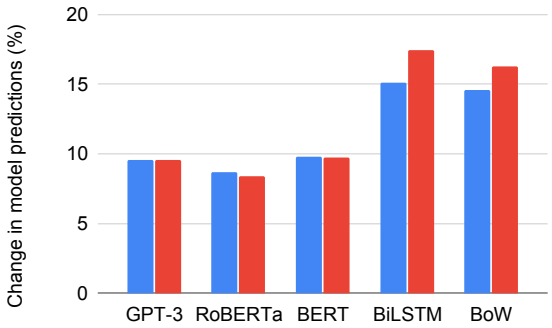

Figure 3: Percentage of examples where the models' predictions changed when the gold label is entailed (blue left bar) or not-entailed (right red bar).

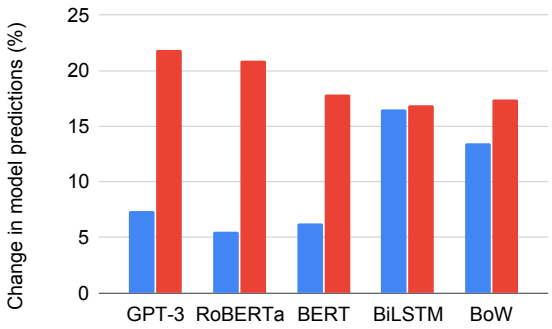

Figure 4: Percentage of examples where models' predictions change when the original prediction was correct (left blue bar) or incorrect (right red bar).

Focusing on the Transformer models, we noticed that RoBERTa performs the best on the datasets and is the most robust to paraphrasing - changing its predictions on just under 8% of paraphrased examples. Interestingly, when the models are trained specifically to perform this task, the models change their predictions on fewer paraphrased examples as these models' accuracy increases. However, improving performance alone might not automatically improve models' robustness to paraphrases. GPT-3's accuracy noticeably outperforms BERT's accuracy, but GPT-3 changes its predictions on more paraphrased examples compared to BERT.

P'-H' compared to P-H' or P'-H Figure 2 shows noticeable increases in the percentage of changed predictions when both premise and hypothesis are paraphrased compared to when just one of the sentences is paraphrased. Specifically, for BoW and BiLSTM we see an increase of 4.01 and 6.01 percentage points respectively, and for BERT, Roberta, GPT-3 increases of 4.97, 4.83, and 3.55. As the transformer-based models changed their predictions on 12-14% of examples where both sentences are paraphrased compared to 9-11% in general, this analysis further suggests that these models are not as robust to paraprhases as desired.

Entailed vs Not-entailed examples RTE analyses often differentiate how models perform on entailed vs not entailed examples (Liu et al., 2022). In Figure 3, we do not see meaningful differences in how models' predictions change on paraphrased examples based on the gold label. This might suggest that our dataset does not contain statistical irregularities based on the RTE labels.

Correct vs Not-Correct Predictions Figure 4 shows that the Transformer models' predictions is more likely to change when it's prediction on an original example was incorrect (right red bars) compared to when the prediction for an original example was correct (left blue bars). For example, when RoBERTa's prediction for an original RTE example was correct, the model changed its prediction on just 5.5% of the corresponding paraphrased examples. When RoBERTa's predictions for an original RTE example were incorrect, RoBERTa's predictions changed for 20.88% corresponding paraphrased examples. Analyzing differences in models' confidences assigned to predictions might provide more insight (Marcé and Poliak, 2022). We leave this for future work.

Source Task RTE1-3 examples originated from multiple domains and downstream tasks, e.g. question-answering (Moldovan et al., 2006), information extraction (Grishman and Sundheim, 1996), and summarization (Evans et al., 2004; Radev et al., 2001). This enables researchers to evaluate how

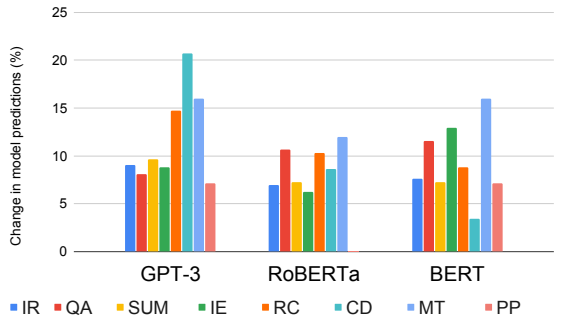

Figure 5: Percentage of examples where models predictions change their predictions depending on the examples' sources. We omit Bow and BiLSTM for space.

RTE models perform on examples that contain different aspects of *open domain inference* necessary for the task (MacCartney, 2009). Figure 5 reports the changes in models' predictions across the different sources of examples. We do not see consistent trends across the original data sources.

6 Conclusion

We introduced $\hat{P}aRTE$, a high-quality evaluation set of RTE examples paired with paraphrased RTE examples. We use our evaluation set to determine whether RTE models are robust to paraphrased examples. Our experiments indicate that while these models predictions are usually consistent when RTE examples are paraphrased, there is still room for improvement as models remain sensitive to changes in input (Jia and Liang, 2017; Belinkov and Bisk, 2018; Iyyer et al., 2018). We hope that researchers will use $\hat{P}aRTE$ to evaluate how well their NLU systems perform on paraphrased data.

Limitations

Our results nor evaluation set cannot be used to indicate whether RTE models trained for other languages are robust to paraphrases. However, researchers can apply the methods we used to develop $\hat{P}aRTE$ to build evaluation sets in other languages to test whether non-English NLU systems are robust to paraphrases.

Ethics Statement

In conducting our research on RTE model robustness to paraphrasing, we take great care to ensure the ethical and responsible use of any data and models involved. We adhere to the principles of fairness, transparency, and non-discrimination in our experimentation and analysis. Furthermore, we take measures to protect the privacy and confidentiality of any individuals crowdsource workers. We also strive to make our evaluation set and methods openly available to the research community to promote further study and advancement in the field of Natural Language Processing.

References

Ion Androutsopoulos and Prodromos Malakasiotis. 2010. A survey of paraphrasing and textual entailment methods. *Journal of Artificial Intelligence Research*, 38:135–187.

Simran Arora, Avanika Narayan, Mayee F. Chen, Laurel Orr, Neel Guha, Kush Bhatia, Ines Chami, Frederic Sala, and Christopher Ré. 2022. Ask me anything: A simple strategy for prompting language models.

Roy Bar-Haim, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, and Bernardo Magnini. 2006. The second pascal recognising textual entailment challenge.

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation. In *International Conference on Learning Representations*.

Wauter Bosma and Chris Callison-Burch. 2006. Paraphrase substitution for recognizing textual entailment. In *Workshop of the Cross-Language Evaluation Forum for European Languages*, pages 502–509. Springer.

Tuhin Chakrabarty, Debanjan Ghosh, Adam Poliak, and Smaranda Muresan. 2021. Figurative language in recognizing textual entailment. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3354–3361, Online. Association for Computational Linguistics.

Robin Cooper, Dick Crouch, Jan Van Eijck, Chris Fox, Johan Van Genabith, Jan Jaspars, Hans Kamp, David Milward, Manfred Pinkal, Massimo Poesio, et al. 1996. Using the framework. Technical report.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *Machine Learning Challenges Workshop*, pages 177–190. Springer.

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*, pages 177–190. Springer.

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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

David Kirk Evans, Judith L. Klavans, and Kathleen R. McKeown. 2004. Columbia newsblaster: Multilingual news summarization on the web. In *Demonstration Papers at HLT-NAACL 2004*, pages 1–4, Boston, Massachusetts, USA. Association for Computational Linguistics.

Erick R. Fonseca and Sandra Maria Aluísio. 2015. Semi-automatic construction of a textual entailment dataset: Selecting candidates with vector space models. In *Proceedings of the 10th Brazilian Symposium in Information and Human Language Technology*, pages 201–210, Natal, Brazil. Sociedade Brasileira de Computação.

Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hannaneh Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, Sameer Singh, Noah A. Smith, Sanjay Subramanian, Reut Tsarfaty, Eric Wallace, Ally Zhang, and Ben Zhou. 2020. Evaluating models' local decision boundaries via contrast sets. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1307–1323, Online. Association for Computational Linguistics.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. 2007. The third pascal recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing*, pages 1–9. Association for Computational Linguistics.

Ralph Grishman and Beth Sundheim. 1996. Message Understanding Conference- 6: A brief history. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*.

J. Edward Hu, Abhinav Singh, Nils Holzenberger, Matt Post, and Benjamin Van Durme. 2019. Large-scale, diverse, paraphrastic bitexts via sampling and clustering. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 44–54, Hong Kong, China. Association for Computational Linguistics.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1875–1885, New Orleans, Louisiana. Association for Computational Linguistics.

Paloma Jeretic, Alex Warstadt, Suvrat Bhooshan, and Adina Williams. 2020. Are natural language inference models IMPPRESsive? Learning IMPlicature and PRESupposition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8690–8705, Online. Association for Computational Linguistics.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark. Association for Computational Linguistics.

Alisa Liu, Swabha Swayamdipta, Noah A. Smith, and Yejin Choi. 2022. WANLI: Worker and AI collaboration for natural language inference dataset creation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6826–6847, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Ro{bert}a: A robustly optimized {bert} pretraining approach.

Tingting Ma, Jin-Ge Yao, Chin-Yew Lin, and Tiejun Zhao. 2021. Issues with entailment-based zero-shot text classification. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 786–796, Online. Association for Computational Linguistics.

Bill MacCartney. 2009. *Natural language inference*. Ph.D. thesis, Stanford University.

Sanjana Marcé and Adam Poliak. 2022. On gender biases in offensive language classification models. In *Proceedings of the 4th Workshop on Gender Bias in Natural Language Processing (GeBNLP)*, pages 174–183, Seattle, Washington. Association for Computational Linguistics.

Dan I. Moldovan, Mitchell Bowden, and M. Tatu. 2006. A temporally-enhanced poweranswer in trec 2006. In *TREC*.

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*, pages 2340–2353, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Zuzana Nevěřilová. 2014. Paraphrase and textual entailment generation. In *International Conference on Text, Speech, and Dialogue*, pages 293–300. Springer.

Alicia Parrish, Sebastian Schuster, Alex Warstadt, Omar Agha, Soo-Hwan Lee, Zhuoye Zhao, Samuel R. Bowman, and Tal Linzen. 2021. NOPE: A corpus of naturally-occurring presuppositions in English. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 349–366, Online. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014a. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014b. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Adam Poliak, Aparajita Haldar, Rachel Rudinger, J. Edward Hu, Ellie Pavlick, Aaron Steven White, and Benjamin Van Durme. 2018a. Collecting diverse natural language inference problems for sentence representation evaluation. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 337–340, Brussels, Belgium. Association for Computational Linguistics.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018b. 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.

Vinodkumar Prabhakaran, Ben Hutchinson, and Margaret Mitchell. 2019. Perturbation sensitivity analysis to detect unintended model biases. 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)*, pages 5740–5745, Hong Kong, China. Association for Computational Linguistics.

Dragomir R. Radev, Sasha Blair-Goldensohn, Zhu Zhang, and Revathi Sundara Raghavan. 2001. NewsInEssence: A system for domain-independent, real-time news clustering and multi-document summarization. In *Proceedings of the First International Conference on Human Language Technology Research*.

Abhilasha Ravichander, Matt Gardner, and Ana Marasovic. 2022. Condaqa: A contrastive reading comprehension dataset for reasoning about negation. In *EMNLP 2022*.

Kyle Richardson, Hai Na Hu, Lawrence S. Moss, and Ashish Sabharwal. 2020. Probing natural language inference models through semantic fragments. In *AAAI*, volume abs/1909.07521.

Alexis Ross and Ellie Pavlick. 2019. How well do NLI models capture verb veridicality? 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)*, pages 2230–2240, Hong Kong, China. Association for Computational Linguistics.

S. Seethamol and K. Manju. 2017. Paraphrase identification using textual entailment recognition. In *2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)*, pages 1071–1074.

Jihao Shi, Xiao Ding, Li Du, Ting Liu, and Bing Qin. 2021. Neural natural logic inference for interpretable question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3673–3684, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jiao Sun, Xuezhe Ma, and Nanyun Peng. 2021. AESOP: Paraphrase generation with adaptive syntactic control. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5176–5189, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Prasetya Utama, Joshua Bambrick, Nafise Moosavi, and Iryna Gurevych. 2022. Falsesum: Generating document-level NLI examples for recognizing factual inconsistency in summarization. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2763–2776, Seattle, United States. Association for Computational Linguistics.

Siddharth Vashishtha, Adam Poliak, Yash Kumar Lal, Benjamin Van Durme, and Aaron Steven White. 2020. Temporal reasoning in natural language inference. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4070–4078, Online. Association for Computational Linguistics.

Martin Víta. 2015. Computing semantic textual similarity based on partial textual entailment. In *Doctoral Consortium on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, volume 2, pages 3–12. SCITEPRESS.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355.

Aaron Steven White, Pushpendre Rastogi, Kevin Duh, and Benjamin Van Durme. 2017. Inference is everything: Recasting semantic resources into a unified evaluation framework. In *Proceedings of the Eighth International Joint Conference on Natural Language*

Processing (Volume 1: Long Papers), pages 996–1005, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Adina Williams, Nikita Nangia, and Samuel 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, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Hitomi Yanaka, Koji Mineshima, Daisuke Bekki, Kentaro Inui, Satoshi Sekine, Lasha Abzianidze, and Johan Bos. 2019a. Can neural networks understand monotonicity reasoning? In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 31–40.

Hitomi Yanaka, Koji Mineshima, Daisuke Bekki, Kentaro Inui, Satoshi Sekine, Lasha Abzianidze, and Johan Bos. 2019b. Help: A dataset for identifying shortcomings of neural models in monotonicity reasoning. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (* SEM 2019)*, pages 250–255.

Hitomi Yanaka, Koji Mineshima, and Kentaro Inui. 2021. Exploring transitivity in neural NLI models through veridicality. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 920–934, Online. Association for Computational Linguistics.

Wenpeng Yin, Jamaal Hay, and Dan Roth. 2019. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. 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)*, pages 3914–3923, Hong Kong, China. Association for Computational Linguistics.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. Paws: Paraphrase adversaries from word scrambling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1298–1308.

Zhengli Zhao, Dheeru Dua, and Sameer Singh. 2018. Generating natural adversarial examples. In *International Conference on Learning Representations*.

Model	Acc(original)	Acc(modified)
GPT-3	82.27%	79.12%
BERT	72.53%	72.55%
RoBERTa	84.35%	82.59%
BiLSTM	53.06%	51.24%
BoW	53.71%	54.7%

Table 3: Accuracy of models on original as well as modified pairs in $\hat{P}aRTE$.

Model	Acc(original)	Acc(modified)
GPT-3	60.21	41.22
BERT	45.74	52.33
RoBERTa	60	42.22
BiLSTM	49.04	48.93
BoW	47.61	52.32

Table 4: Accuracy of models on original as well modified pairs in $\hat{P}aRTE$ for cases where their predictions change upon encountering paraphrases.

A Experimental Implementation Details

This section describes the model implementations for our experiments. For our work we trained/fine-tuned three different models - Bag of Words (BoW), BiLSTM, BERT-large with a classification head and RoBERTa-large with a classification head. Each model was trained on the MultiNLI training dataset (Williams et al., 2018) and validated on the paraphrased RTE dev set we created. Each model was implemented using PyTorch. All transformer based models were downloaded from HuggingFace.

A.1 BoW

The BoW model consisted of GloVe (300 dimension embeddings trained on 840B CommonCrawl tokens) (Pennington et al., 2014b) vectors as the embedding layer. The average of all word vectors for the input sequence is treated as its final representation. The representations for the hypothesis and premises were concatenated and passed through three fully connected layers with ReLU activation units after each layer. We concatenate the premise, hypothesis, their absolute difference and their product and pass it into the first layer of the classifier. This input to the first layer is of 4 * embedding dimension and the output is of embedding dimension. Each subsequent hidden layer's input and output dimensions are embedding dimension * embedding dimension.

The model was trained with a vocabulary size of 50,000, a learning rate of 0.005, the maximum sequence length was 50 and a batch size of 32. We force all sentences to be of maximum sequence length using truncation or padding where applicable. We train the model for 15 epochs and select the one that achieves highest validation accuracy for our experiments.

A.2 BiLSTM

The BiLSTM model consisted of GloVe (300 dimension embeddings trained on 840B CommonCrawl tokens) (Pennington et al., 2014a) vectors as the embedding layer. The average of all word vectors for the input sequence is treated as its final representation. The word vectors were passed through an LSTM unit. This unit was bidirectional, with 64 hidden units and 2 stacked LSTM layers. The representations for the hypothesis and premises were concatenated and passed through three fully connected layers with ReLU activation units after each layer. We concatenate the premise, hypothesis, their absolute difference and their product and pass it into the first layer of the classifier. This input to the first layer is of hidden units * embedding dimension and the output is of embedding dimension. Each subsequent hidden layer's input and output dimensions are embedding dimension * embedding dimension.

The model was trained with a vocabulary size of 50,000, a learning rate of 0.005, the maximum sequence length was 50 and a batch size of 32. We force all sentences to be of maximum sequence length using truncation or padding where applicable. We train the model for 15 epochs and select the one that achieves highest validation accuracy for our experiments.

A.3 BERT

We fine tuned the BERT-large model available on HuggingFace [7]. We added a classification head on top of the model using the AutoModel API on HuggingFace. The model was trained for 5 epochs with a learning rate of 3e-6 using the Adam optimizer. In order to simulate larger batch sizes on smaller GPUs, we used gradient accumulation as well. We simulated a batch-size of 32 by accumulating gradients over two batches of size 16. The model which achieved the highest validation accuracy was used for our experiments.

[7] https://huggingface.co/bert-large-uncased

A.4 RoBERTa

We fine tuned the RoBERTa-large model available on HuggingFace [8]. We added a classification head on top of the model using the AutoModel API on HuggingFace. The model was trained for 5 epochs with a learning rate of 3e-6 using the Adam optimizer. In order to simulate larger batch sizes on smaller GPUs, we used gradient accumulation as well. We simulated a batch-size of 32 by accumulating gradients over 8 batches of size 4. The model which achieved the highest validation accuracy was used for our experiments.

A.5 GPT-3

We used a temperature of 0.0 for all the experiments to select the most likely token at each step, as this setting allow for reproducibility.

```
response = openai.Completion.create(
            model="text-davinci-003",
            prompt=prompt,
            temperature=0,
            max_tokens=1,
            top_p=1.0,
            frequency_penalty=0.1,
            presence_penalty=0.0
)
```

We restricted the model outputs to just one token. Only “yes" or “no" are considered valid answers. The model did not generate any output apart from these in all our experiments. We used the following prompt template:

```
Premise: {sentence1}
Hypothesis: {sentence2}

Does the premise entail the hypothesis?
Answer:
```

B Dataset Creation

The following process describes how we create a vetted, paraphrased version of the RTE dataset that tests whether models' are robust to paraphrased input. First, we use a strong T5-based paraphraser to create three re-written sentences for each premise and hypothesis in the 2,400 pairs in the RTE1-3 test sets, resulting in 14,400 new sentences. To generate these paraphrases, we use top-k sampling during decoding.[9] The re-writer model was fine-tuned on the Google PAWS dataset and can be found on Huggingface [10]. To evaluate its ability to generate gramatically correct paraphrases, we sampled 100 sentence pairs with at least one valid paraphrase and manually went through them. Upon checking for grammaticality, we found a grammatical error in <8% of the sentences.

Since we want to test paraphrastic understanding beyond simple lexical replacement, we discarded the re-written sentences that had at most a 25% lexical overlap with the corresponding original sentence. We use Jaccard index as a measure of lexical similarity (1) where τ_s are the tokens in the original sentence and τ_p are the the tokens in the paraphrase.

$$Score = \frac{\tau_s \cap \tau_p}{\tau_s \cup \tau_p} \quad (1)$$

To ensure that the re-written sentences are indeed sentence-level paraphrases for the original sentences, we relied on crowdsource workers to remove low quality paraphrases. The Amazon Mechanical Turk HIT is described in detail in subsection B.2. We retain any paraphrases that get a similarity score above 75 out of 100.

B.1 Manual Verification

Before crowd sourcing to get the best paraphrase generated for a given sentence, we conducted manual evaluation to understand the average error rate of the paraphraser model used. As mentioned above, we sampled 100 sentence pairs with each pair having atleast one valid paraphrase. The paraphrases for these sentences were evaluated for grammatical errors. Any semantic errors are handled during crowd-sourcing.

The errors can roughly be classified into roughly three categories - repetition errors, tense errors and incorrect punctuation. Examples of each type can be found in Figure 6. Overall, we found the error rate to be small enough to continue using the paraphraser. We also asked MTurk workers to mark paraphrases as grammatically incorrect to ensure that the final dataset does not have any grammatically incorrect sentences.

B.2 MTurk HIT

We used Amazon Mechanical Turk to identify ungrammatical paraphrases rate how well a generated paraphrase preserved the meaning of the original sentence. No filtering criteria was applied to crowdsource workers and were paid roughly $14.20 an hour.

Each annotator was presented with a reference sentence, a corresponding paraphrased sentences, and tasked to judge on a scale of 0 to 100 how

[8] https://huggingface.co/roberta-large

[9] k=120; top-p=0.95

[10] https://huggingface.co/Vamsi/T5_Paraphrase_Paws

Original sentence	Paraphrase	Error
British servicemen detained	British servicemen detained by British servicemen detained	Repetition in the sentence
The state charges against Nichols are for 160 victims and one victim 's fetus .	The state charges against Nichols are for 160 victims and one victims'fetus.	Incorrect apostrophe after "victims"
The engine can answer specific queries directly .	The engine can direct answer specific queries.	Adjective changed to "direct"

Figure 6: Types of errors made by the paraphraser model

closely a paraphrased sentence retains the meaning of the reference sentence. A similarity score of 100 means that the paraphrase is the exactly the same in meaning as the reference, while a similarity score of 0 means that the meaning of the paraphrase is irrelevant or contradicts the reference sentence. Additionally, the MTurk workers were asked to judge the grammaticality of the paraphrase by selecting whether the paraphrase was grammatically correct or now. Figure 7 includes the instructions we showed crowdsource workers for judging similarity between sentences.

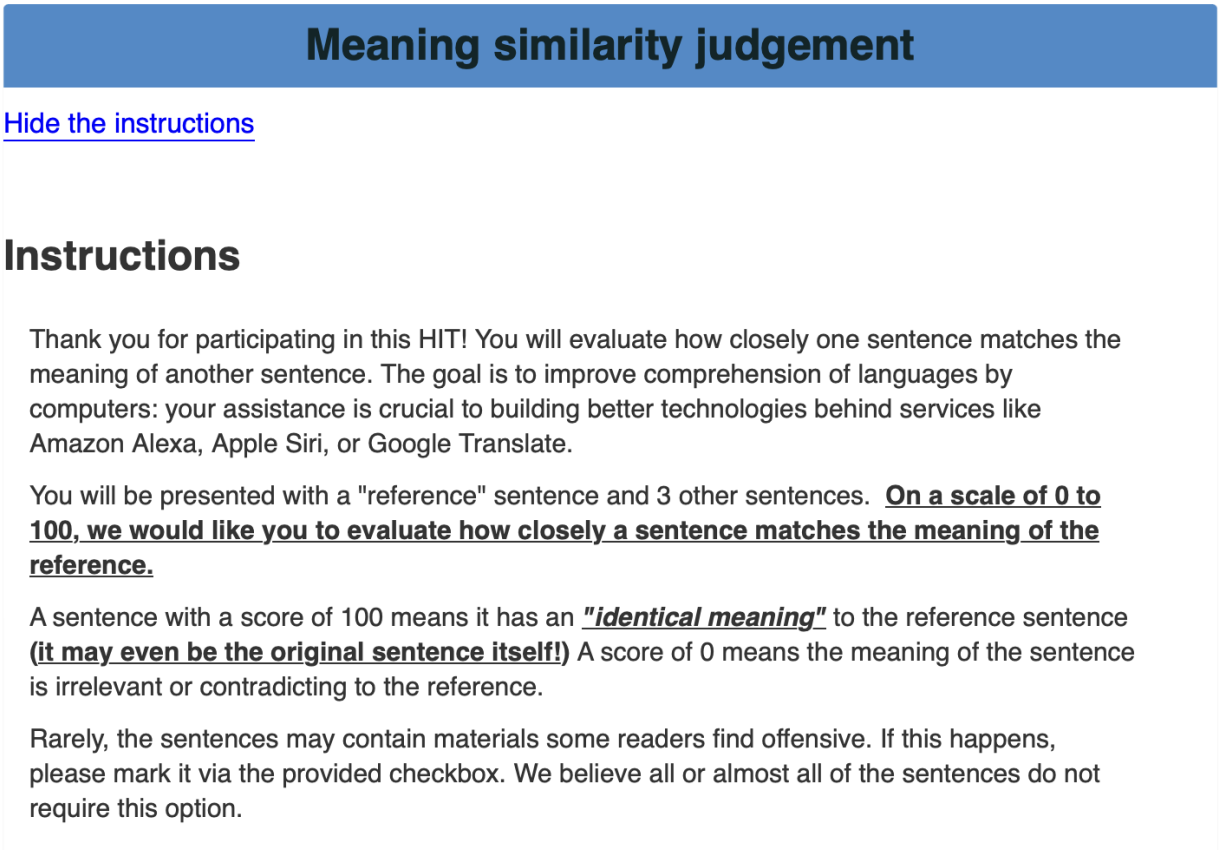

Figure 7: Instructions for semantic similarity and grammatically check.