Systematic review of effect of data augmentation using paraphrasing on named entity recognition

Saket Sharma[1¶] **Aviral Joshi**[1¶] **Namrata Mukhija**[1§*] **Yiyun Zhao**[¶]

Hanoz Bhathena[¶] **Prateek Singh**[¶] **Sashank Santhanam**[‖†] **Pritam Biswas**[¶]

[¶]Machine Learning Center of Excellence, JPMorgan Chase
[§]New York University
[‖]Apple
saket.sharma@jpmchase.com

Abstract

While paraphrasing is a promising approach for data augmentation in classification tasks, its effect on named entity recognition (NER) is not investigated systematically due to the difficulty of span-level label preservation. In this paper, we utilize simple strategies to annotate entity spans in generations and compare established and novel methods of paraphrasing in NLP such as back translation, specialized encoder decoder models such as Pegasus, and GPT-3 variants for their effectiveness in improving downstream performance for NER across different levels of gold annotations and paraphrasing strength on 5 datasets. We also analyze the quality of generated paraphrases based on their entity preservation and paraphrasing language quality. We find that the choice of the paraphraser greatly impacts NER performance, with one of the larger GPT-3 variants exceedingly capable at generating high quality paraphrases, improving performance in most cases, and not hurting others, while other paraphrasers show more mixed results. We also find inline auto annotations generated by larger GPT-3 to be strictly better than heuristic based annotations. We find diminishing benefits of paraphrasing as gold annotations increase for most datasets. While larger GPT-3 variants perform well by both entity preservation and human evaluation of language quality, those two metrics do not necessarily correlate with downstream performance for other paraphrasers.

1 Introduction

Named entity recognition (NER) seeks to extract entity mentions (e.g., Shakespeare, Warwickshire) from a text (Shakespeare was born and raised in Warwickshire) for predefined categories of interest (such as people and locations). It is a critical component underpinning many industrial pipelines for a variety of downstream natural language processing applications such as search, recommendation, and virtual assistant systems. However, in real-world applications, there is often a scarcity of labeled data for training advanced deep neural models because span-level NER annotations are costly, and domain expertise may be needed to annotate data from domains such as finance, biomedical sciences, etc.

Data augmentation is often used as an alternative to address the data scarcity issue in many NLP tasks (see an NLP data augmentation survey by Feng et al. (2021)). However, data augmentation

[*]Work done while interned at JPMorgan Chase.
[†]Work done while at JPMorgan Chase.
[1]Equal contribution.

for NER imposes additional challenges because NER requires token/span level label preservation. Therefore, most existing works on NER data augmentation primarily focus on local replacement for entity mentions (Dai and Adel, 2020; Zhou et al., 2022; Liu et al., 2022; Zhu et al., 2021) as well as context words (Dai and Adel, 2020; Li et al., 2020). The replacements can be other mentions with the same labels (Dai and Adel, 2020), synonyms from an external lexical resource such as wordnet (Dai and Adel, 2020), or tokens generated by the pretrained language models such as BERT via masked token task (Zhou et al., 2022; Liu et al., 2022; Zhu et al., 2021). However, to enhance the reliability of masked token prediction, the language model usually needs to be fine-tuned on the NER training data and label information is often inserted close to the [MASK]s (Zhou et al., 2022; Zhu et al., 2021), which requires a decent amount of labeled training data.

This work primarily focuses on the less-studied data augmentation method for NER – paraphrasing – which has the potential to introduce structural and lexical replacement and does not assume many labeled examples. Specifically, we compare established, and novel paraphrasing methods and propose simple ways to preserve span-level labels. Unlike most existing studies that focus on the influence of the amount of gold data only, we systematically investigate the effect of different levels of paraphrasing on downstream performance, at different levels of gold annotations across 5 datasets. We investigate the quality of paraphrases from 6 different systems as augmentation data, as well as stand alone training data for NER; and also conduct quality analysis of paraphrases generated by different systems based on the NER preservation and language quality.

We find paraphrasing to be generally effective in low data regimes for most paraphrasers. However, the choice of paraphrases affects the magnitude, and direction of the change in performance across all levels of gold data. We find the use of LLMs to generate inline annotations[3] while paraphrasing to be superior to simpler heuristics, and GPT-3 Davinci variant with inline annotations to be a generally superior choice across datasets. While paraphrases generated by GPT-3 with inline annotations also score highest on NER preservation and language quality, we do not see broader correlation between paraphrase quality metrics and downstream performance.

2 Datasets and Paraphrasers

2.1 Datasets

NER datasets are chosen to have coverage across a variety of domains including news, Wikipedia, Twitter, biomedical research and search; while also having a diverse set of entity types (word phrases, alphanumeric, datetime, alphabetical etc.).

We choose 5 datasets based on the above principles: Ontonotes5 (Hovy et al., 2006), Tweebank (Jiang et al., 2022), WNUT 2017 (Derczynski et al., 2017), MIT Restaurant NER dataset (MIT-R) (Liu et al., 2013), BioCreative V CDR (BC5CDR) (Wei et al., 2016). Pre-formatted versions of all datasets are sourced from the TNER project (Ushio and Camacho-Collados, 2021) on Huggingface datasets(Wolf et al., 2020). Datasets such as WNUT also have rare entities by design, allowing us to probe robustness against entity memorization.

2.2 Paraphrasers and postprocessing

In our experiments, we compare six paraphrasing systems:(1) Back Translation, (2) Pegasus, (3) Ada (Prompt A), (4) Ada (Prompt B), (5) Davinci (Prompt A) and (6) Davinci (Prompt B). We generate a maximum of 4 unique paraphrases per seed gold sentence for each paraphraser and postprocess the paraphrases with simple re-annotation and filtering.

2.2.1 Back-translation; BT

Back translation has been widely used as a data augmentation method (Sugiyama and Yoshinaga, 2019; Corbeil and Ghadivel, 2020; Xie et al., 2020) including in phrase based systems like Bojar and Tamchyna (2011). For our experiments we use pre-trained English-German and German-English models available from Huggingface Hub [4] via Tiedemann and Thottingal (2020) and the model

[3]Inline annotation: [Shakespeare](PERSON) was born and raised in [Warwickshire](LOC)
[4]https://huggingface.co/models

	MIT-R	Onto-notes	BC5-CDR	Twee-bank	Wnut-17
BT	1	0	2	0	3
Pegasus	1	0	13	3	8
Ada-A	10	0	0	11	0
Ada-B	4	0	0	**16**	2
DaV-A	3	0	4	5	0
DaV-B	**26**	**35**	**26**	10	**27**

Table 1: Counts the configurations of G & P where a paraphraser shows highest improvement over no paraphrasing baseline for a given G. DaV-B short for DaVinci (B) outperforms other paraphrasers across most datasets

	MIT-R	Onto-notes	BC5-CDR	twee-bank	Wnut-17
BT	0.66	0.74	0.76	**0.41**	0.30
Pegasus	0.68	0.75	0.78	0.33	0.23
Ada-A	0.71	0.73	0.74	0.36	0.23
Ada-B	0.70	0.72	0.74	0.34	0.23
DaV-A	0.67	0.75	0.76	0.39	0.27
DaV-B	**0.73**	**0.80**	**0.82**	**0.41**	**0.32**

Table 2: Test micro-F1 when training using only paraphrases with P=1 for full dataset. Number in bold is the maximum for a given dataset. DaV-B short for DaVinci(B) outperforms all paraphrasers across datasets

architecture used is BART (Lewis et al., 2019). We use a temperature parameter of 0.8 with greedy decoding.

2.2.2 PEGASUS Paraphraser

PEGASUS, introduced in Zhang et al. (2020) for the purpose of summarization, is a large (568mn parameters) pre-trained transformer (Vaswani et al., 2017) based encoder-decoder model, pre-trained using a custom self-supervised objective. To use it as a paraphraser the model was fine-tuned on a paraphrasing task. We use an off-the-shelf version of PEGASUS fine-tuned for paraphrasing released on Huggingface model hub [5]

2.2.3 GPT-3 variants

GPT-3 Brown et al. (2020) is an auto-regressive decoder only transformer pre-trained for language modeling, showing impressive in-context learning, and instruction following ability ((Radford et al., 2019); (Sanh et al., 2021); (Wei et al., 2021); (Ouyang et al., 2022), (Campos and Shern, 2022)). We use the OpenAI API [6] to query the Ada ($\sim$350M parameters), and DaVinci ($\sim$175B parameters) variants of GPT-3. We prompt both GPT-3 variants with two versions of one shot prompts with a temperature of 0.8, max length of 100, and default values for other parameters (See Appendix 6.1):

- Prompt A – GPT-3 variant is instructed to generate paraphrases without specific instruction to retain inline annotation for entities
- Prompt B – GPT-3 variant is instructed to generate paraphrases, while also retaining inline annotation for entities

Post-processing & filtering of paraphrases We re-annotate outputs of all paraphrasers based on a case insensitive exact match search for the entity values present in seed sentence. In the case of LLMs generating inline annotations, this logic is used to supplement annotations generated by the model, relying on the model generated annotations in cases of conflicts. Further filtering is applied to the paraphrases from all models to remove paraphrases for seed sentences shorter than 15 characters, remove paraphrases that are a duplicate of the seed sentence or of another paraphrase, and when generation contains an entity not present in entity space of the dataset. We also retain only the first generation of multiline generations for paraphrasers generating a numbered list of paraphrases (common with prompt driven GPT-3 variants Appendix 6.2)

[5]https://huggingface.co/tuner007/pegasus_paraphrase
[6]https://beta.openai.com/

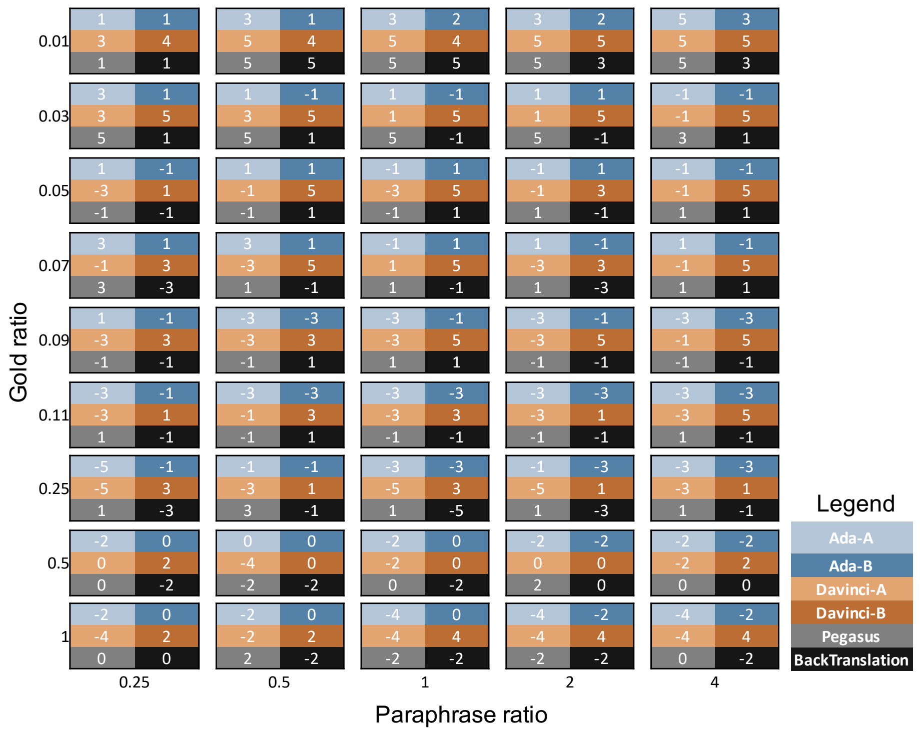


Figure 1: Matrix of scores of how F1 changed relative to the no paraphrasing (P=0) baseline after the addition of synthetic data across datasets for different G & P ratios. Improvement/worsening in any dataset at a given G/P ratio gets a score of +1/-1 respectively, and aggregation is then done across datasets. Higher numbers represent better performance across datasets. Score $\in [-5, 5]$

3 Experiments

3.1 Using paraphrases as augmentation data

Experimental setup In real world scenarios, we get annotated gold data incrementally. In our experiments, we simulate this by sampling gold data at various levels (1% to 100%) by building upon previous samples. For example, while generating gold sample for G=0.01 (first sample), we sample 1% of the total dataset, stratified by entities. However, when sampling for G=0.02, we retain the sample from the first step, and sample an additional 1% of the remaining dataset. This process is repeated until G=1. As a result, going from G=0.25 to G=0.5 does not actually double the gold dataset used in training. At each sampling step, we also sample an equivalent percentage of gold samples with no entities. Early experiments suggested increased benefit of paraphrasing at lower dataset size, so we explore more G ratios in this space. Additionally we only go upto G=0.25 for the really large Ontonotes dataset for speed.

For each gold to paraphrase ratio combination, we first sample gold data by the method described above. Then we randomly sample paraphrases for the gold IDs sampled until and including the current sample. This dataset is used to fine-tune a distilbert-base-cased base model for named entity recognition using the 1-step training described by (Okimura et al., 2022) using standard classification loss over hidden states of individual tokens. The models are trained to convergence with early stopping with a patience of 5.

We generate the overall, and entity specific micro F1 for each G/P combination along with standard deviation across three runs.

Results As Table 1 and Figure 1 suggest, choice of paraphraser strongly dictates the augmentation performance. GPT-3 Davinci (Prompt B) consistently outperforms, or matches other paraphrasers and is a safe default choice for paraphrasing across domains. Across the Davinci variants, inline annotations with Prompt B strictly outperform those introduced using heuristics. Davinci Prompt B

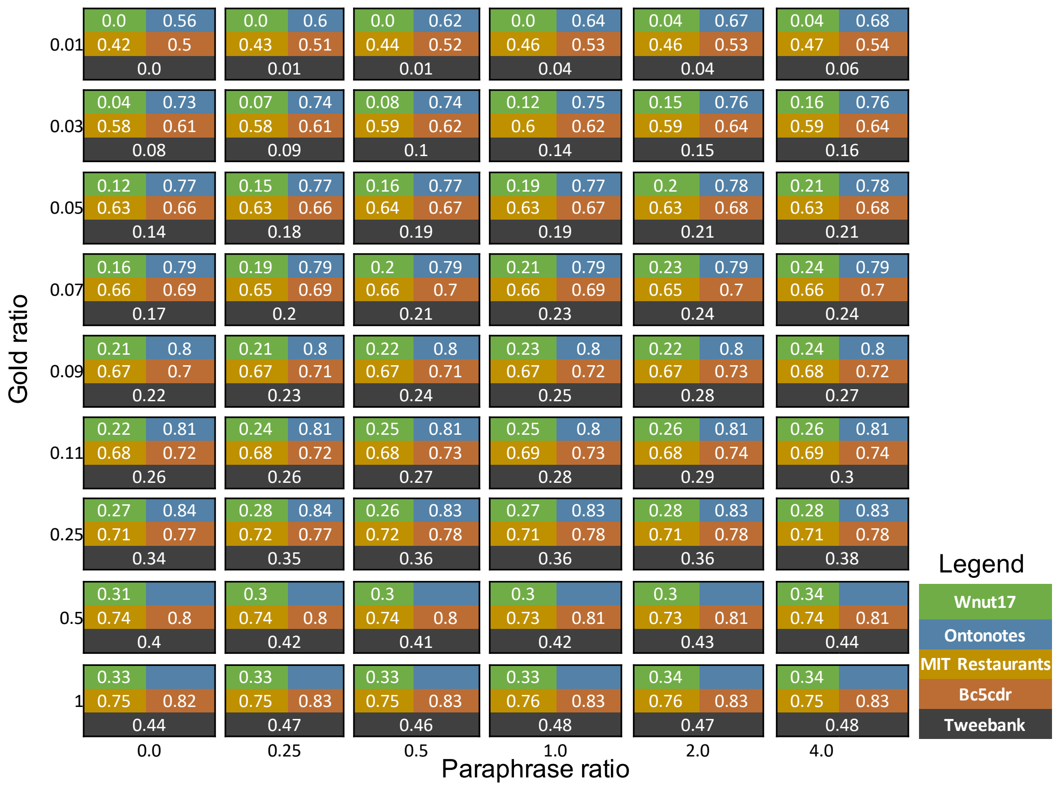


Figure 2: Micro F1 for Davinci (Prompt B) on datasets across gold and paraphrase ratios

also achieves or matches best performance at G=1 (0.25 for Ontonotes) and P=4 across all datasets. Ada variants show the most inconsistent results, with Backtranslation and Pegasus outperforming them as well as Davinci (Prompt A) in many cases. Full results are available in Appendix 6.4.

Increasing gold ratios While we run similar experiments on all paraphraser-dataset pairs, we share the results from DaVinci (Prompt B) on all datasets in Figure 2 because of its consistent improvements in NER performance. (Full results Appendix 6.4). We see consistent benefits of paraphrasing at lower gold ratios, and diminishing returns in relative performance bump as we go to higher values. Other paraphrasers show similar trends at low G ratios with some exceptions (Ada variants in BC5CDR, and Backtranslation on MIT-R) (See Figure 1, Appendix 6.4), although we see a lot more mixed results at medium to high G ratios.

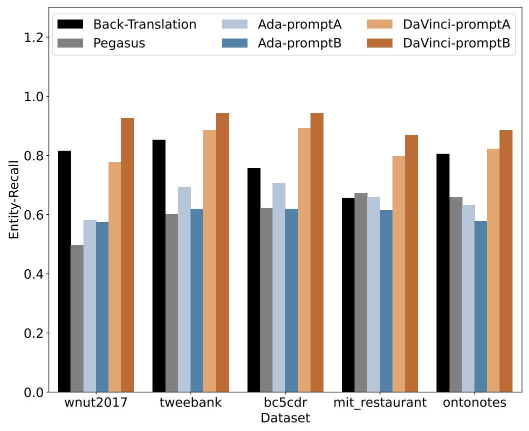
Increasing paraphrase ratios For DaVinci (Prompt B), as we increase paraphrasing, an initial bump in performance is seen followed by flattening performance, with a minor drop in performance in some cases (Figure 2). Other paraphrasers show more mixed results (Appendix: 6.4)

Finally, paraphrasing with DaVinci (Prompt B) also leads to improvements in performance for the WNUT17 dataset (Figure 2, Appendix 6.4) across all G ratios, showing signs of robustness against entity memorization, although deeper analysis on memorization is left for future work.

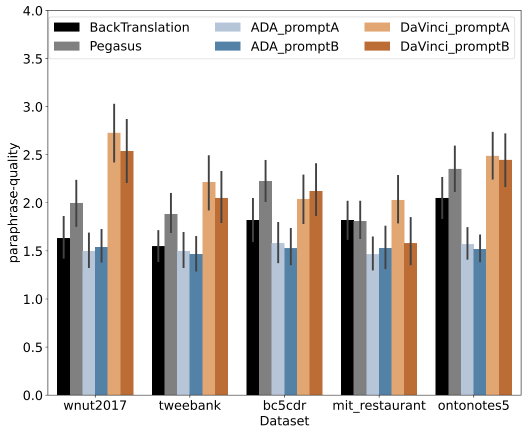
3.2 Using paraphrases as training data

Experimental setup We further evaluate quality of paraphrases directly by using **only** synthetic data to train NER models. These experiments are done only for P=1 for paraphrases generated for all training data (G=1).

Results As seen in Table 2, we find GPT-3 DaVinci Prompt B paraphrases performing best across all datasets. The trends among paraphrasers track augmentation performance in Figure 1 and Appendix 6.4.



(a) Entity recall evaluation



(b) Human evaluation of paraphrasing quality

Figure 3: Paraphrase Evaluation

3.3 Paraphrase generation quality Analysis

Besides assessing usefulness for NER with actual training, we investigate paraphrase generation quality directly from two perspectives – entity preservation and paraphrase quality to see to what extent these metrics correlate with NER performance.

As entities are central to NER, we hypothesize entity preservation to be important for performance. We count the number of gold entities that appear in paraphrases with correct annotations via a case insensitive string match (entity recall). This calculation sets a lower bound of the entity preservation accuracy.

Good paraphrases are also expected to introduce form variety while preserving the meaning faithfully, potentially helping downstream performance. We asked three human annotators to annotate paraphrases generated by the six systems for 50 training examples sampled for each dataset. Specifically, human annotators were instructed to ignore the entity accuracy and to score paraphrases from 1-5 based on the paraphrasing quality. Our guidelines are similar to (Niu et al., 2020) (Appendix 6.5)

3.3.1 Results

According to Figure 3(a), among all the paraphrase systems Davinci (Prompt B) has the highest entity recall rate, followed by Davinci (Prompt A) and backtranslation. While, Ada and Pegasus are more likely to lose gold entities. This suggests a large-sized GPT-3 model with an appropriate prompt can generate examples with high-quality inline entity annotations but a small-sized GPT-3 consistently underperforms even a simple Back-translation system. Figure 3(b) shows Davinci systems always have the best human evaluation scores across datasets followed by Pegasus and Back-translation, while Ada systems are consistently the worst.

In summary, we find that paraphrases generated by the Davinci (Prompt B) system often preserve entities and are of a good paraphrasing quality whereas Ada systems consistently underperform other systems in both metrics across datasets. These results are partially consistent with the downstream evaluations in that the augmentation data generated by Davinci (Prompt B) have reliably better downstream performance compared to other systems. However, broader trends in paraphrasing quality do not track with downstream NER performance.

4 Future work

While our work proposes a paraphrasing pipeline that performs consistently better than established paraphrasing pipelines in NER, we expect further benefits to come from more exhaustive tuning of prompts used to generate paraphrases. Another potential direction to improve downstream performance is to explore better (than random) sampling strategy for paraphrases (based on entity density, entity recall, or other metrics).

5 Conclusion

We study the effect of six paraphrasing systems on downstream NER performance across 5 datasets. We find that the choice of paraphraser system (model + prompt) strongly affects NER performance. GPT-3 Davinci (Prompt B) performs the best at both NER performance and paraphrasing quality metrics but other paraphrasers show mixed results, suggesting that GPT-3 Davinci (Prompt B) is a strong default choice. We further find that generating inline annotations using GPT-3 Davinci works superior to strictly heuristic based annotations. While we find paraphrasing to be more effective at lower amount of training data, it helps at higher levels depending on dataset, and paraphraser. We observe dataset dependent optimal paraphrase ratios for Davinci (Prompt B), with diminishing results as paraphrasing is increased; whereas other paraphrasers show mixed results. Paraphrases from Davinci (Prompt B) have the best quality, and downstream performance, but we do not find general correlation between paraphrase quality, and downstream performance.

References

Ondřej Bojar and Aleš Tamchyna. Improving translation model by monolingual data. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 330–336, Edinburgh, Scotland, July 2011. Association for Computational Linguistics. URL `https://aclanthology.org/W11-2138`.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Jon Ander Campos and Jun Shern. Training language models with language feedback. In *ACL Workshop on Learning with Natural Language Supervision. 2022.*, 2022.

Jean-Philippe Corbeil and Hadi Abdi Ghadivel. Bet: A backtranslation approach for easy data augmentation in transformer-based paraphrase identification context. *arXiv preprint arXiv:2009.12452*, 2020.

Xiang Dai and Heike Adel. An analysis of simple data augmentation for named entity recognition. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3861–3867, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.343. URL `https://aclanthology.org/2020.coling-main.343`.

Leon Derczynski, Eric Nichols, Marieke van Erp, and Nut Limsopatham. Results of the WNUT2017 shared task on novel and emerging entity recognition. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 140–147, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-4418. URL `https://aclanthology.org/W17-4418`.

Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. A survey of data augmentation approaches for NLP. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 968–988, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.84. URL `https://aclanthology.org/2021.findings-acl.84`.

Charles R Harris, K Jarrod Millman, Stéfan J Van Der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J Smith, et al. Array programming with numpy. *Nature*, 585(7825):357–362, 2020.

Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. OntoNotes: The 90% solution. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 57–60, New York City, USA, June 2006. Association for Computational Linguistics. URL `https://aclanthology.org/N06-2015`.

Hang Jiang, Yining Hua, Doug Beeferman, and Deb Roy. Annotating the tweebank corpus on named entity recognition and building NLP models for social media analysis. *CoRR*, abs/2201.07281, 2022. URL `https://arxiv.org/abs/2201.07281`.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

Kun Li, Chengbo Chen, Xiaojun Quan, Qing Ling, and Yan Song. Conditional augmentation for aspect term extraction via masked sequence-to-sequence generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7056–7066, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.631. URL `https://aclanthology.org/2020.acl-main.631`.

Jian Liu, Yufeng Chen, and Jinan Xu. Low-resource ner by data augmentation with prompting. In Lud De Raedt, editor, *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 4252–4258. International Joint Conferences on Artificial Intelligence

Organization, 7 2022. doi: 10.24963/ijcai.2022/590. URL `https://doi.org/10.24963/ijcai.2022/590`. Main Track.

Jingjing Liu, Panupong Pasupat, Yining Wang, Scott Cyphers, and Jim Glass. Query understanding enhanced by hierarchical parsing structures. In *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 72–77. IEEE, 2013.

Wes McKinney et al. pandas: a foundational python library for data analysis and statistics. *Python for high performance and scientific computing*, 14(9):1–9, 2011.

Tong Niu, Semih Yavuz, Yingbo Zhou, Nitish Shirish Keskar, Huan Wang, and Caiming Xiong. Unsupervised paraphrasing with pretrained language models. *arXiv preprint arXiv:2010.12885*, 2020.

Itsuki Okimura, Machel Reid, Makoto Kawano, and Yutaka Matsuo. On the impact of data augmentation on downstream performance in natural language processing. In *Proceedings of the Third Workshop on Insights from Negative Results in NLP*, pages 88–93, 2022.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*, 2022.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

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

Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*, 2021.

Amane Sugiyama and Naoki Yoshinaga. Data augmentation using back-translation for context-aware neural machine translation. In *Proceedings of the Fourth Workshop on Discourse in Machine Translation (DiscoMT 2019)*, pages 35–44, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-6504. URL `https://aclanthology.org/D19-6504`.

Jörg Tiedemann and Santhosh Thottingal. OPUS-MT — Building open translation services for the World. In *Proceedings of the 22nd Annual Conferenec of the European Association for Machine Translation (EAMT)*, Lisbon, Portugal, 2020.

Asahi Ushio and Jose Camacho-Collados. T-NER: An all-round python library for transformer-based named entity recognition. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 53–62, Online, April 2021. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/2021.eacl-demos.7`.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, 17(3):261–272, 2020.

Chih-Hsuan Wei, Yifan Peng, Robert Leaman, Allan Peter Davis, Carolyn J Mattingly, Jiao Li, Thomas C Wiegers, and Zhiyong Lu. Assessing the state of the art in biomedical relation extraction: overview of the biocreative v chemical-disease relation (cdr) task. *Database*, 2016, 2016.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/2020.emnlp-demos.6`.

Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. *Advances in Neural Information Processing Systems*, 33:6256–6268, 2020.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR, 2020.

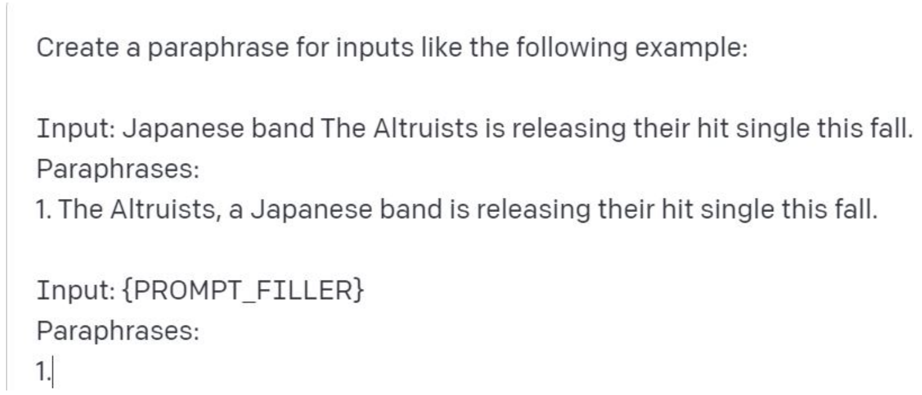
Ran Zhou, Xin Li, Ruidan He, Lidong Bing, Erik Cambria, Luo Si, and Chunyan Miao. MELM: Data augmentation with masked entity language modeling for low-resource NER. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2251–2262, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.160. URL `https://aclanthology.org/2022.acl-long.160`.

Wenjing Zhu, Liu Jian, Xu Jinan, Chen Yufeng, and Zhang Yujie. Improving low-resource named entity recognition via label-aware data augmentation and curriculum denoising. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 1131–1142, Huhhot, China, August 2021. Chinese Information Processing Society of China. URL `https://aclanthology.org/2021.ccl-1.101`.

6 Appendix

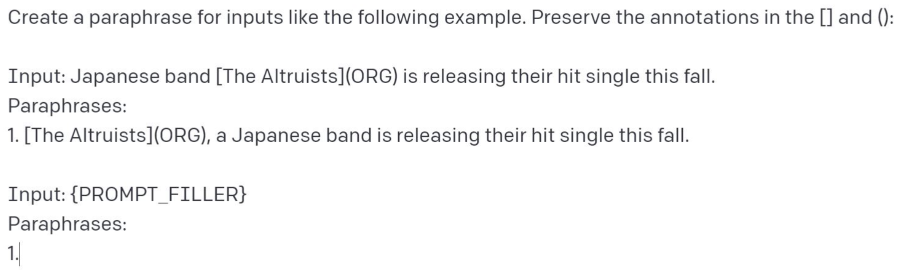
6.1 Prompt design

The following prompts are used in the experiments:



```
Create a paraphrase for inputs like the following example:

Input: Japanese band The Altruists is releasing their hit single this fall.
Paraphrases:
1. The Altruists, a Japanese band is releasing their hit single this fall.

Input: {PROMPT_FILLER}
Paraphrases:
1.
```

Figure 4: GPT-3 is instructed to generate paraphrases similar to an example without any specific instruction to retain inline annotations

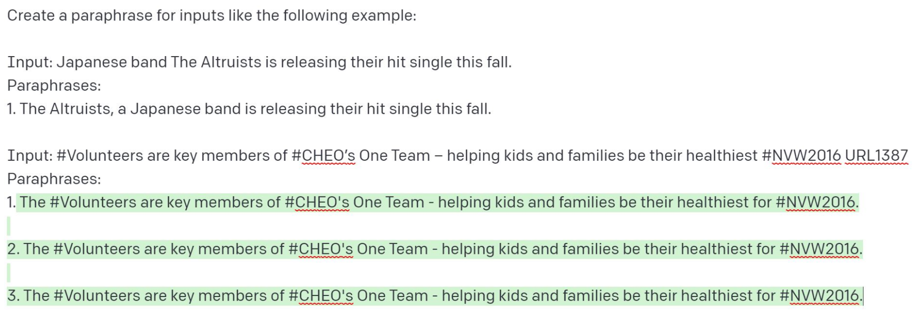


```
Create a paraphrase for inputs like the following example. Preserve the annotations in the [] and ():

Input: Japanese band [The Altruists](ORG) is releasing their hit single this fall.
Paraphrases:
1. [The Altruists](ORG), a Japanese band is releasing their hit single this fall.

Input: {PROMPT_FILLER}
Paraphrases:
1.
```

Figure 5: GPT-3 is instructed to generate paraphrases similar to an example, asking it to retain inline annotations

6.2 Multiline generation

LLM paraphrasers can be triggered to generate multi-line outputs. This behavior is more common in Ada variants over DaVinci, showing the DaVinci is better at following prompt instructions.

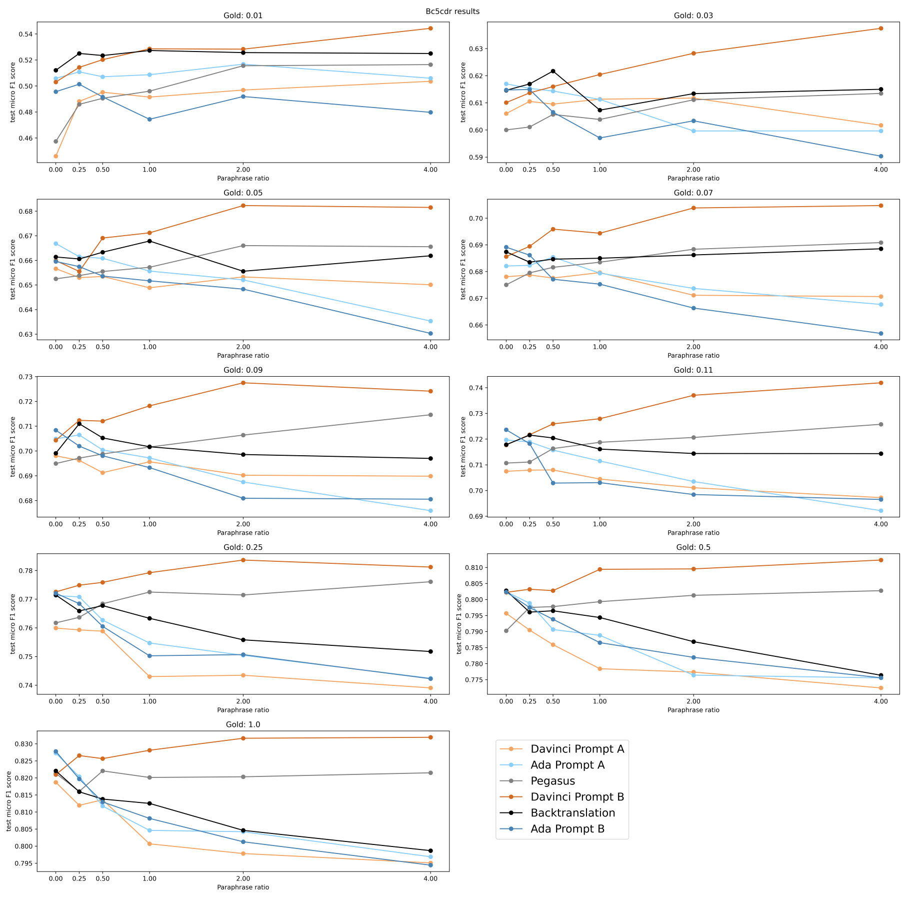


```
Create a paraphrase for inputs like the following example:

Input: Japanese band The Altruists is releasing their hit single this fall.
Paraphrases:
1. The Altruists, a Japanese band is releasing their hit single this fall.

Input: #Volunteers are key members of #CHEO’s One Team – helping kids and families be their healthiest #NVW2016 URL1387
Paraphrases:
1. The #Volunteers are key members of #CHEO's One Team - helping kids and families be their healthiest for #NVW2016.

2. The #Volunteers are key members of #CHEO's One Team - helping kids and families be their healthiest for #NVW2016.

3. The #Volunteers are key members of #CHEO's One Team - helping kids and families be their healthiest for #NVW2016.
```

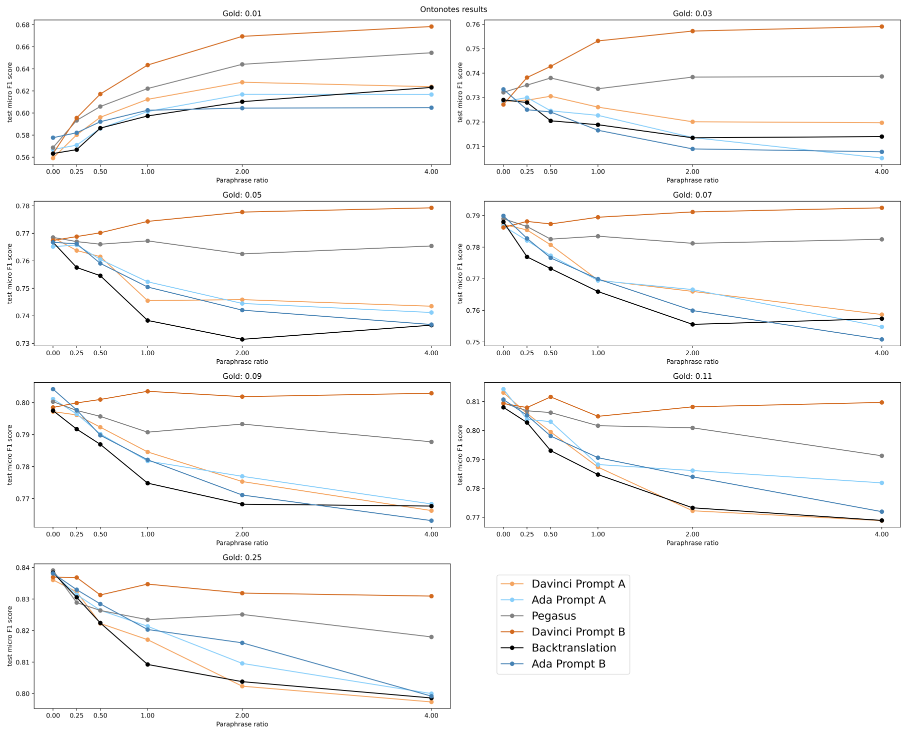
Figure 6: GPT-3 variants sometimes generate multiple numbered paraphrases. We choose to retain only the first paraphrase in these cases

6.3 Detailed results across different gold data sizes for all datasets

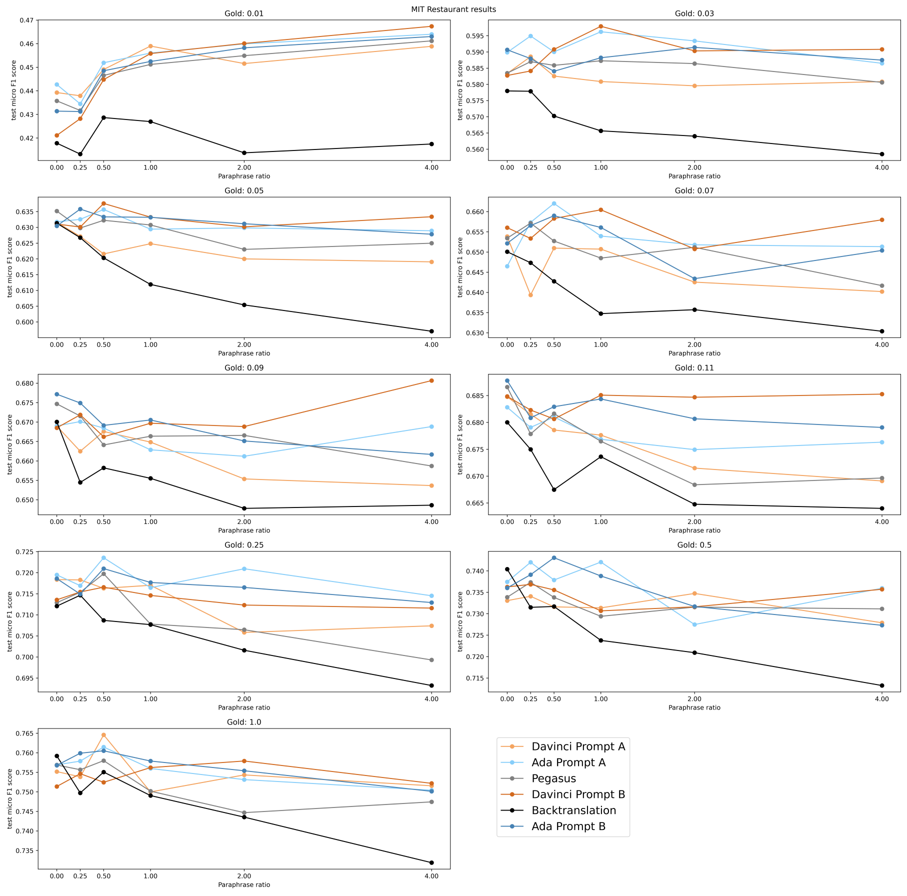
6.3.1 BC5CDR



6.3.2 Ontonotes

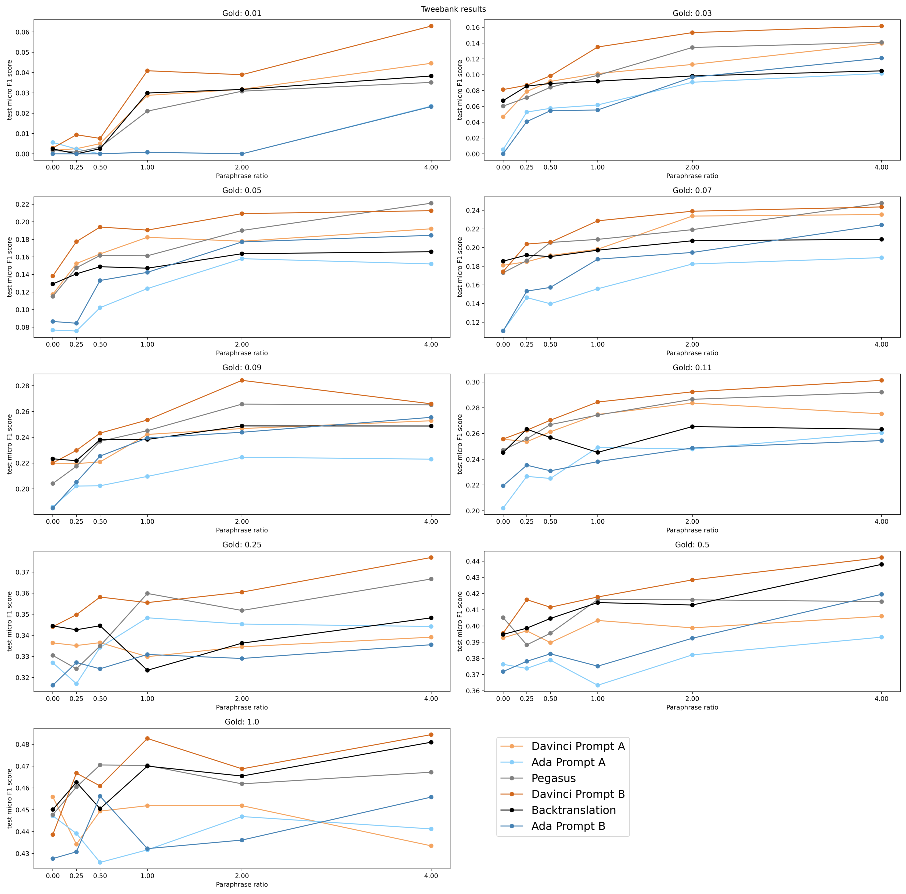


6.3.3 MIT-R

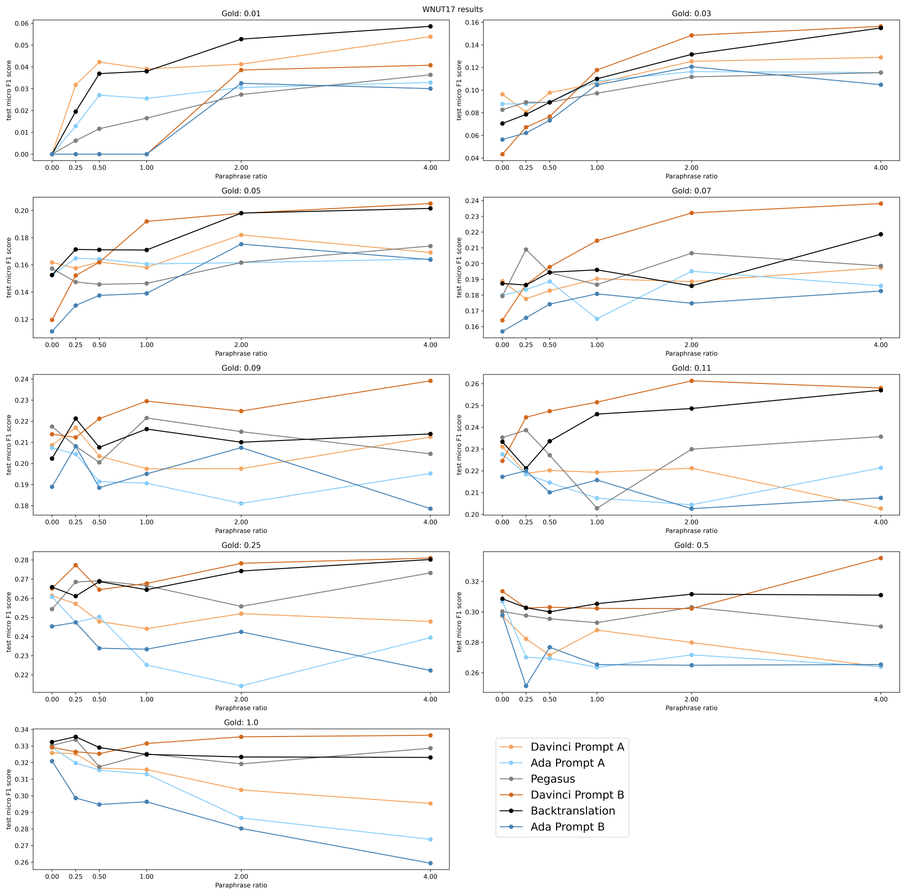


MIT Restaurant results
Gold: 0.01
Gold: 0.03
Gold: 0.05
Gold: 0.07
Gold: 0.09
Gold: 0.11
Gold: 0.25
Gold: 0.5
Gold: 1.0
test micro F1 score
Paraphrase ratio
Davinci Prompt A
Ada Prompt A
Pegasus
Davinci Prompt B
Backtranslation
Ada Prompt B

6.3.4 Tweebank

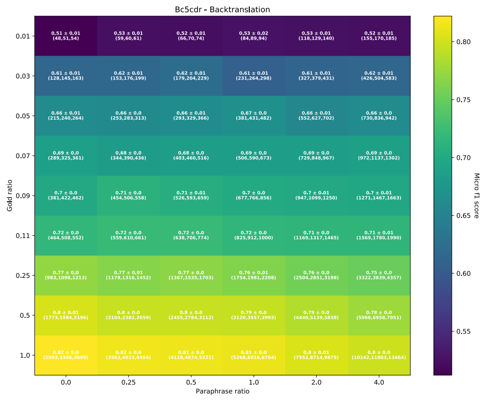


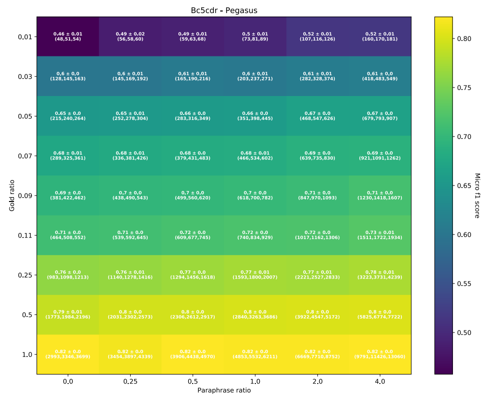
6.3.5 WNUT-17

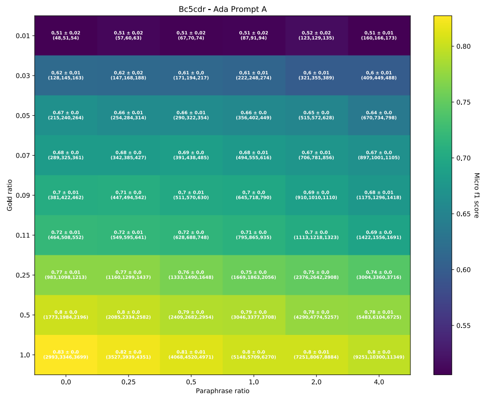

WNUT17 results
Gold: 0.01
Gold: 0.03
Gold: 0.05
Gold: 0.07
Gold: 0.09
Gold: 0.11
Gold: 0.25
Gold: 0.5
Gold: 1.0
test micro F1 score
Paraphrase ratio
Davinci Prompt A
Ada Prompt A
Pegasus
Davinci Prompt B
Backtranslation
Ada Prompt B

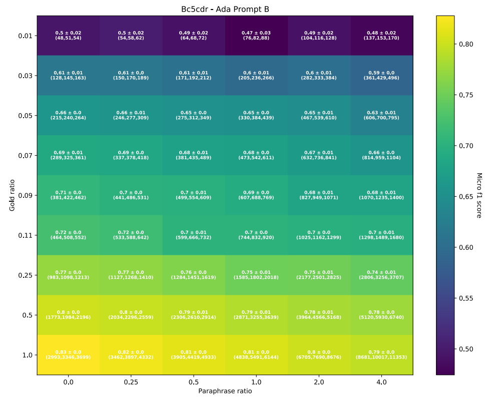
6.4 Heatmap of micro-f1 scores across all datasets & paraphrasers

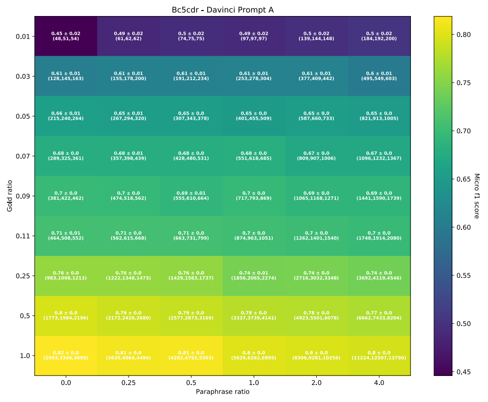
6.4.1 BC5CDR

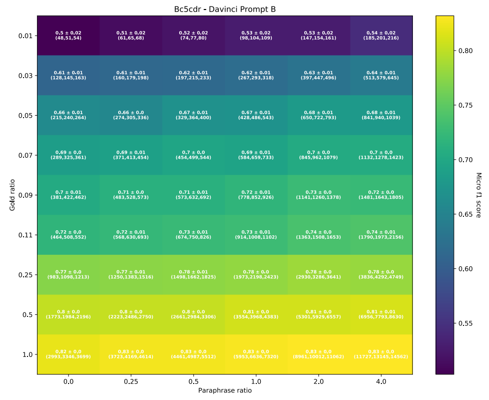




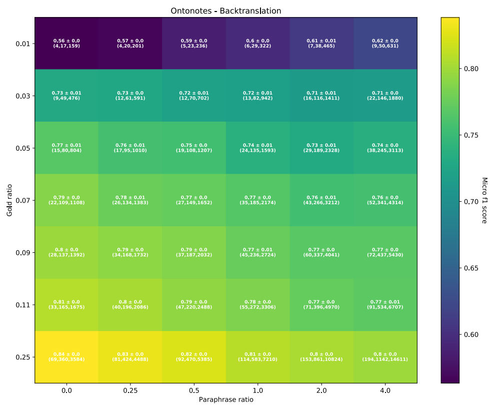


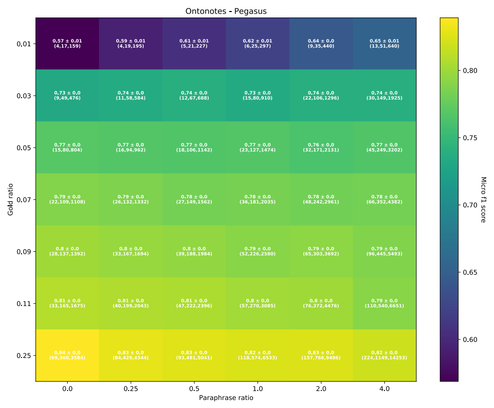


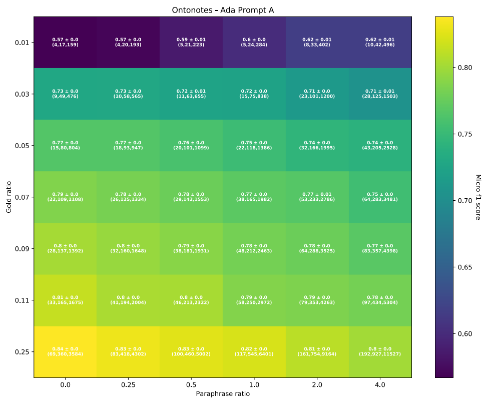


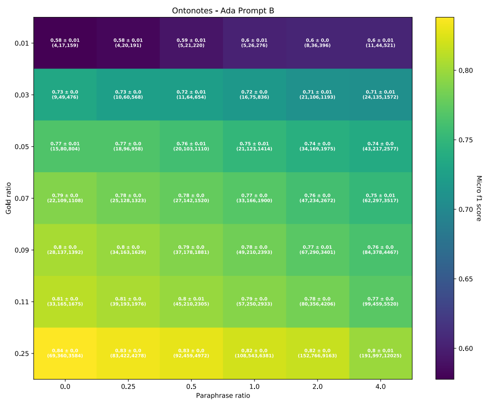


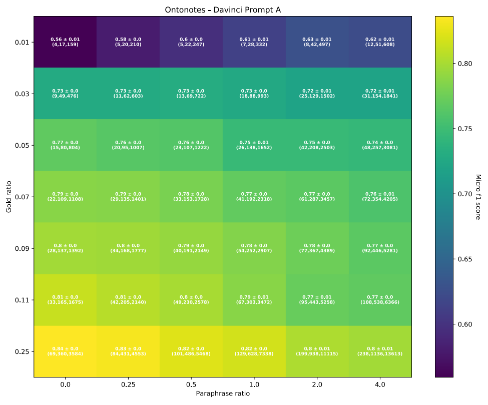
6.4.2 Ontonotes

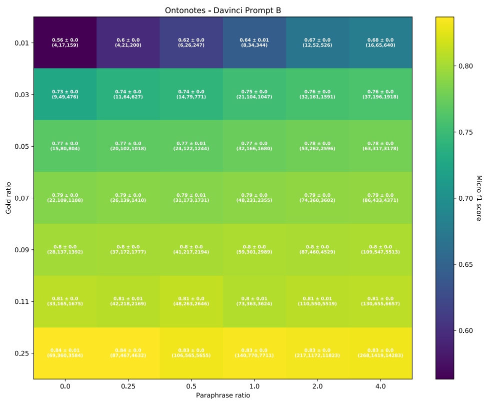




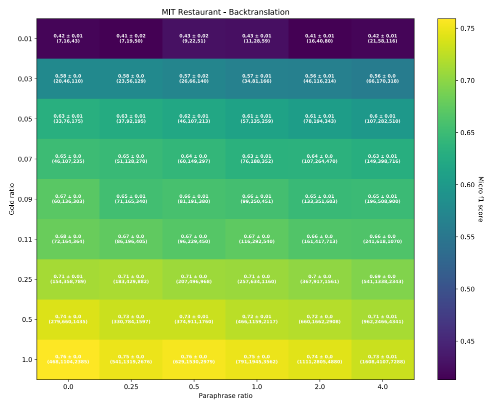


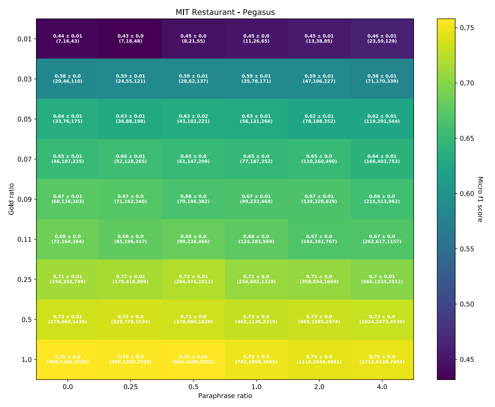


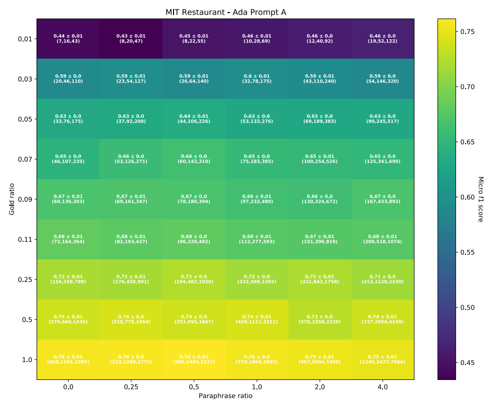


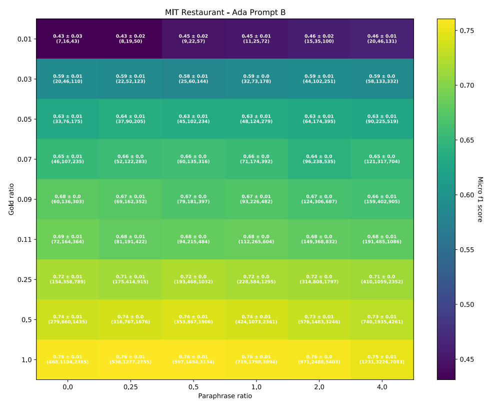


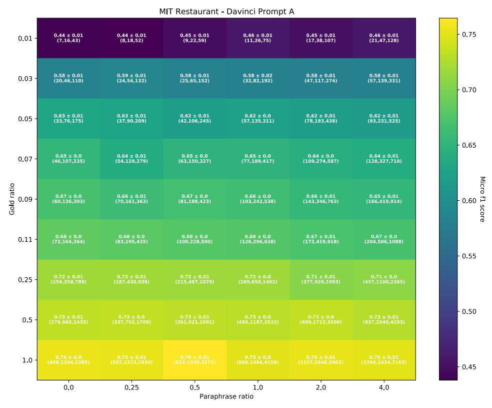
6.4.3 MIT-R

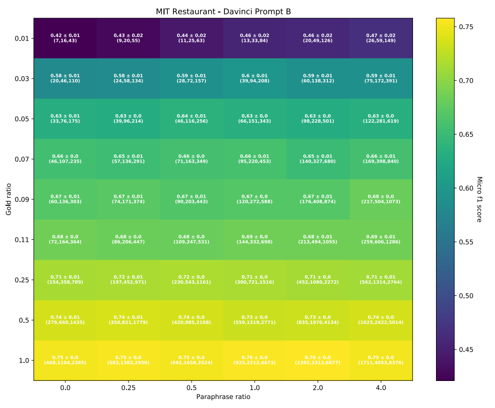

MIT Restaurant - Backtranslation
Gold ratio
Paraphrase ratio
Micro f1 score
0.01
0.03
0.05
0.07
0.09
0.11
0.25
0.5
1.0
0.0
0.25
0.5
1.0
2.0
4.0


MIT Restaurant - Pegasus
Gold ratio
Paraphrase ratio
Micro f1 score

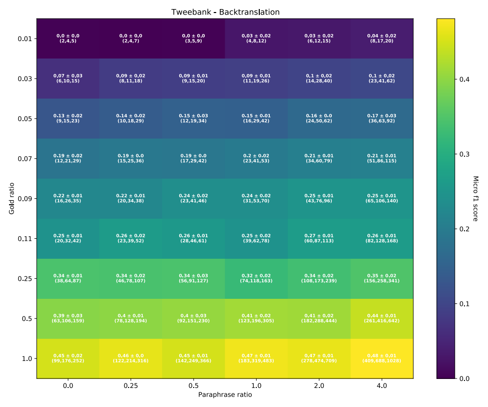

MIT Restaurant - Ada Prompt A
Gold ratio
Paraphrase ratio
Micro f1 score

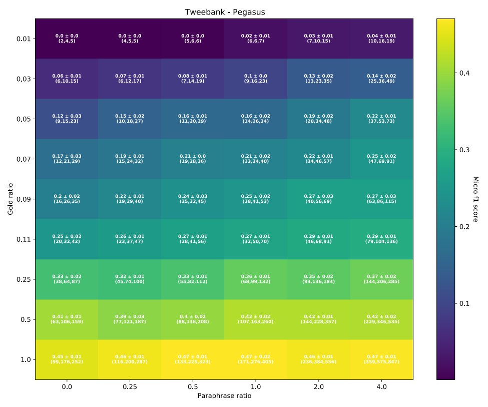

MIT Restaurant - Ada Prompt B
Gold ratio
Paraphrase ratio
Micro f1 score

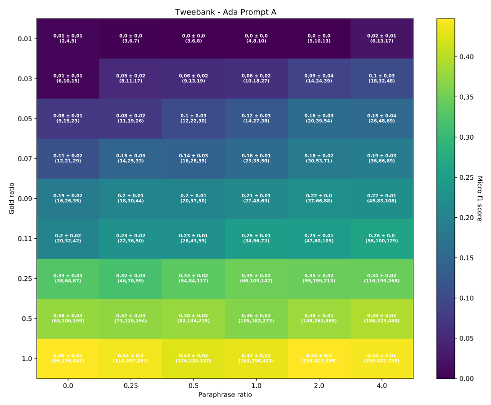

MIT Restaurant - Davinci Prompt A
Gold ratio
Paraphrase ratio
Micro f1 score

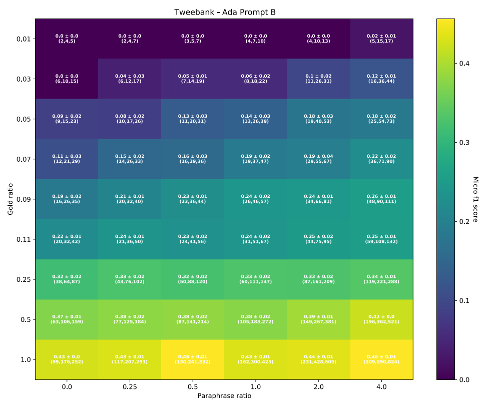

MIT Restaurant - Davinci Prompt B
Gold ratio
Paraphrase ratio
Micro f1 score

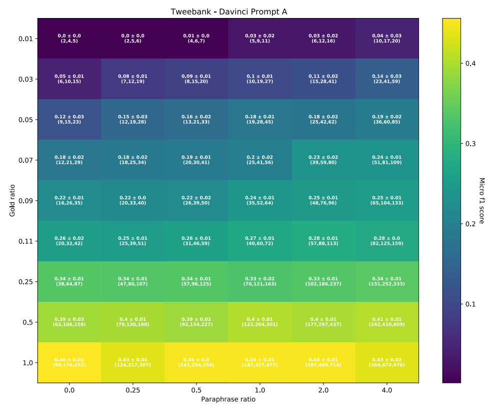
6.4.4 Tweebank

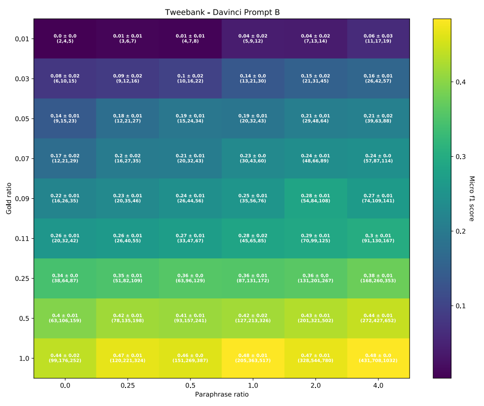




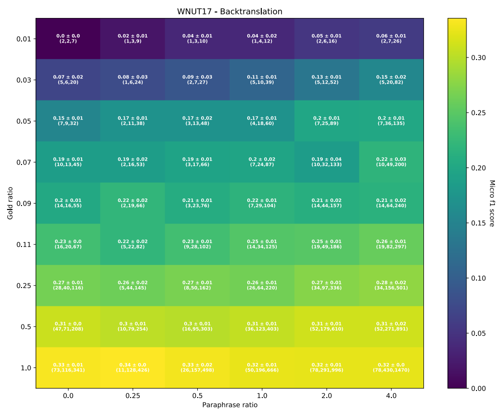


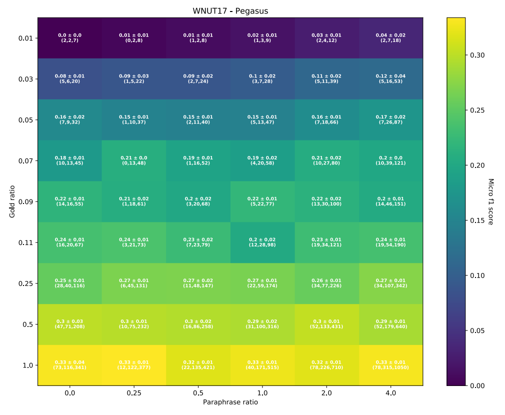


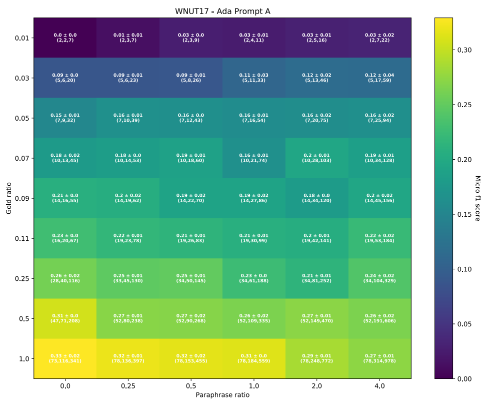


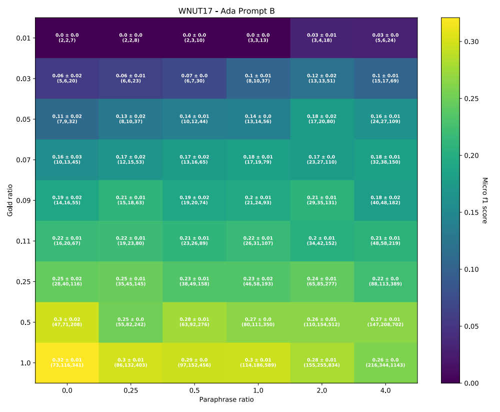


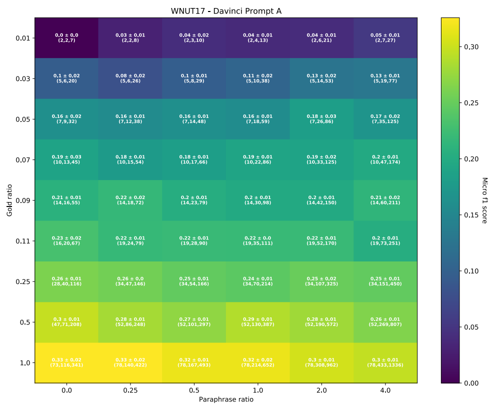
6.4.5 WNUT-17

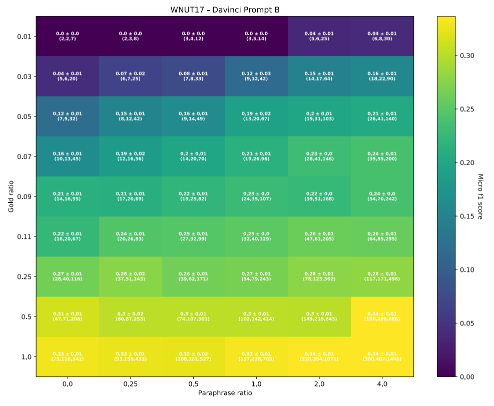

WNUT17 - Backtranslation
Gold ratio
Paraphrase ratio
Micro f1 score


WNUT17 - Pegasus
Gold ratio
Paraphrase ratio
Micro f1 score

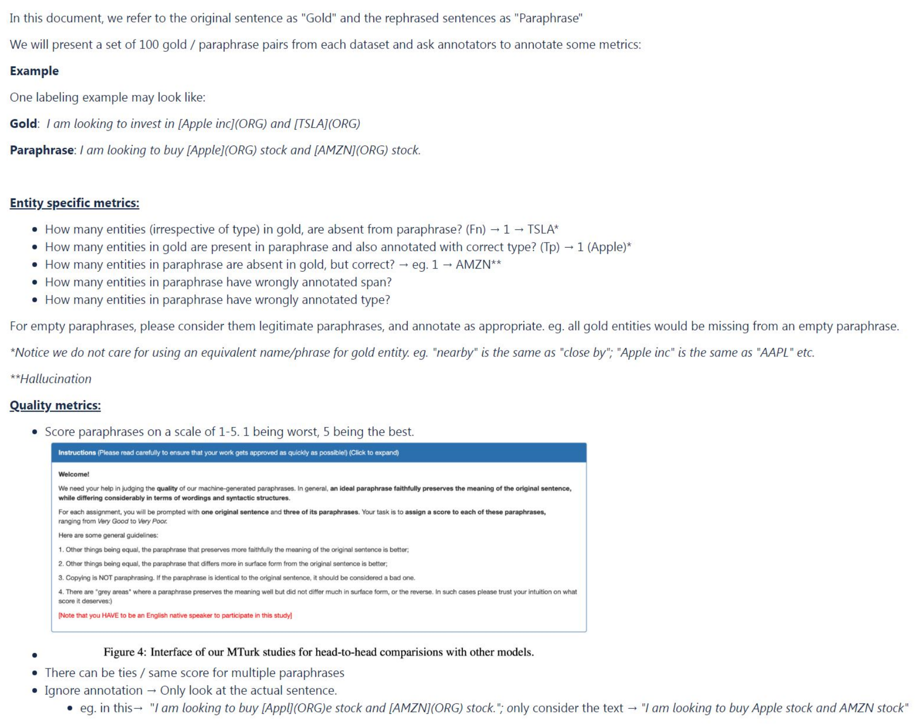

WNUT17 - Ada Prompt A
Gold ratio
Paraphrase ratio
Micro f1 score


WNUT17 - Ada Prompt B
Gold ratio
Paraphrase ratio
Micro f1 score


WNUT17 - Davinci Prompt A
Gold ratio
Paraphrase ratio
Micro f1 score


WNUT17 - Davinci Prompt B
Gold ratio
Paraphrase ratio
Micro f1 score

6.5 Human evaluation guidelines



In this document, we refer to the original sentence as "Gold" and the rephrased sentences as "Paraphrase"

We will present a set of 100 gold / paraphrase pairs from each dataset and ask annotators to annotate some metrics:

Example

One labeling example may look like:

Gold: *I am looking to invest in [Apple inc](ORG) and [TSLA](ORG)*

Paraphrase: *I am looking to buy [Apple](ORG) stock and [AMZN](ORG) stock.*

Entity specific metrics:

- How many entities (irrespective of type) in gold, are absent from paraphrase? (Fn) → 1 → TSLA*
- How many entities in gold are present in paraphrase and also annotated with correct type? (Tp) → 1 (Apple)*
- How many entities in paraphrase are absent in gold, but correct? → eg. 1 → AMZN**
- How many entities in paraphrase have wrongly annotated span?
- How many entities in paraphrase have wrongly annotated type?

For empty paraphrases, please consider them legitimate paraphrases, and annotate as appropriate. eg. all gold entities would be missing from an empty paraphrase.

**Notice we do not care for using an equivalent name/phrase for gold entity. eg. "nearby" is the same as "close by"; "Apple inc" is the same as "AAPL" etc.*

***Hallucination*

Quality metrics:

- Score paraphrases on a scale of 1-5. 1 being worst, 5 being the best.

Instructions (Please read carefully to ensure that your work gets approved as quickly as possible!) (Click to expand)

Welcome!

We need your help in judging the **quality** of our machine-generated paraphrases. In general, **an ideal paraphrase faithfully preserves the meaning of the original sentence, while differing considerably in terms of wordings and syntactic structures.**

For each assignment, you will be prompted with **one original sentence** and **three of its paraphrases**. Your task is to **assign a score to each of these paraphrases,** ranging from *Very Good* to *Very Poor.*

Here are some general guidelines:

1. Other things being equal, the paraphrase that preserves more faithfully the meaning of the original sentence is better;

2. Other things being equal, the paraphrase that differs more in surface form from the original sentence is better;

3. Copying is NOT paraphrasing. If the paraphrase is identical to the original sentence, it should be considered a bad one.

4. There are "grey areas" where a paraphrase preserves the meaning well but did not differ much in surface form, or the reverse. In such cases please trust your intuition on what score it deserves:)

[Note that you HAVE to be an English native speaker to participate in this study]

- Figure 4: Interface of our MTurk studies for head-to-head comparisions with other models.
- There can be ties / same score for multiple paraphrases
- Ignore annotation → Only look at the actual sentence.
  - eg. in this→ "*I am looking to buy [Appl](ORG)e stock and [AMZN](ORG) stock.*"; only consider the text → "*I am looking to buy Apple stock and AMZN stock*"

6.6 Software Acknowledgements

This work would be much harder without the use of several software packages including, but not limited to Pytorch (Paszke et al., 2019), Huggingface transformers (Wolf et al., 2020), Scipy (Virtanen et al., 2020), Pandas (McKinney et al., 2011), Numpy (Harris et al., 2020), Scikit-learn (Pedregosa et al., 2011), and OpenAI models and Python library.