Punctuation Restoration Improves Structure Understanding without Supervision

Junghyun Min[1], Minho Lee[2], Woochul Lee, Yeonsoo Lee[3],
[1]Linguistics Department, Georgetown University, Washington, DC, USA
[2]KT Gen AI Lab, Seocho, Seoul, Republic of Korea,
[3]NC AI, Seongnam, Gyeonggi, Republic of Korea
Correspondence: jm3743@georgetown.edu

Abstract

Unsupervised learning objectives like autoregressive and masked language modeling constitute a significant part in producing pre-trained representations that perform various downstream applications from natural language understanding to conversational tasks. However, despite impressive generative capabilities of recent large language models, their abilities to capture syntactic or semantic structure within text lag behind. We hypothesize that the mismatch between linguistic performance and competence in machines is attributable to insufficient learning of linguistic structure knowledge via currently popular pre-training objectives. Working with English, we show that punctuation restoration as a learning objective improves performance on structure-related tasks like named entity recognition, open information extraction, chunking, and part-of-speech tagging. Punctuation restoration results in ▲≥ 2%p improvement in 16 out of 18 experiments, across 6 out of 7 tasks. Our results show that punctuation restoration is an effective learning objective that can improve structure understanding and yield a more robust structure-aware representations of natural language in base-sized models.

1 Introduction

The modern natural language processing paradigm centers around transformer-based pre-trained language models (PLMs; Peters et al. (2018); Radford et al. (2018); Devlin et al. (2019)). They are optimized on masked language modeling (MLM) and autoregressive language modeling, which provide powerful representations to approach various problems in natural language processing. It is no exaggeration that language models have become effective in tasks like named entity recognition (NER), information extraction, semantic role labeling (SRL) that require understanding of syntactic, semantic, and discourse structure (Wang et al., 2021, 2022). However, the following suggests there is still room for improvement in current language models' abilities to understand such structure in natural language to perform downstream tasks reliably and robustly.

1. **The reversal or factorization curse**. Language models fail to infer "B is A" from "A is B" (Berglund et al., 2024), or their representations are highly dependent on the order (factorization) of the input (Kitouni et al., 2024).

2. **The curse of performance instability**. Model checkpoint initialization and training dataset order strongly affects sensitivity to syntactic structure (Zhou et al., 2020; McCoy et al., 2020; Du and Nguyen, 2023).

3. **Poor out-of-distribution generalization**. Systems report close-to-human performance on one dataset yet perform poorly on other datasets representing the same task, due to their picking up **spurious correlations** rather than learning the task (Gururangan et al., 2018; McCoy et al., 2019; Serrano et al., 2023).

4. **Insufficient or underutilized structure information**. While PLMs do encode some structure, they are poor few-shot structure predictors (Zhao et al., 2023; Bai et al., 2023) and perform better when input is reinforced with linguistic structure information (Strubell et al., 2018; He et al., 2020; Sachan et al., 2021; Wu et al., 2021; Fei et al., 2021; Xie et al., 2023; Huang et al., 2024). This indicates their representations are insufficient or underutilized.

These four phenomena illustrate that current representations as a result of autoregressive (Radford et al., 2018) or masked (Devlin et al., 2019; Liu et al., 2019; Raffel et al., 2019) language modeling are insufficient for structure understanding.

Efforts to mitigate such shortcomings include data-oriented approaches like syntactic augmentation to improve robustness to spurious correlations (Min et al., 2020; Yaghoobzadeh et al., 2021) and reversing input to mitigate the reversal curse (Golovneva et al., 2024). Architecture oriented efforts include adding explicit graph network layers to encode structure, resulting in improvement in benchmark scores (Zhang et al., 2019; Sachan et al., 2021) and generalization abilities (He et al., 2020; Sartran et al., 2022).

They are human-in-the-loop methods that require human input or annotation, or a system that requires it. Recent work in distilling linguistic structure knowledge from natural language text to representations without supervision include inside-outside dynamic programming for tree induction (Drozdov et al., 2019), dependency-constrained self-attention (Shen et al., 2021; Momen et al., 2023), and augmenting MLM with sentence-level contrastive learning (CLEAR; Wu et al., 2020). With the exception of CLEAR, these methods require additions to the model architecture. Wang et al. (2021) and Wang et al. (2022) propose structure pre-training but use human-annotated data.

In this paper, we investigate whether it is possible for an unsupervised method to mitigate the four shortcomings of the modern language models without additional parser or tree architecture implementation. In particular, we believe the pre-training stage of current PLMs may be further improved and propose punctuation restoration (PR) as an unsupervised learning objective that improves structure understanding. Punctuation markers, along with capitalization, often serve as boundary markers between different syntactic components of the sentence (Briscoe, 1996; Bayraktar et al., 1998). Thus, the model's ability to predict punctuation from plain text may correlate to its ability to encode syntactic boundaries and thus structure. We hypothesize that additional optimization on punctuation restoration yields representations with increased sensitivity to structure, measured by in-distribution and out-of-distribution generalization performance in structure-related NLP tasks.

Punctuation and capitalization restoration partially overlaps with language modeling. However, the task still remains nontrivial (Păiş and Tufiş, 2022; de Lima et al., 2024), and explicit optimization would allow models to predict them without explicit local context (e.g. beginning of sentence or quotation).

2 Objective and experimental setup

2.1 Objective design

The PR objective predicts the original text from its "cleared-formatting" counterpart. In our implementation, we remove the following set of punctuation marks: the comma **,**, the period **.**, the exclamation point **!**, the question mark **?**, the single-quotation mark **'**, and the double-quotation mark **"**, along with capitalization, as shown below. Boldface indicates an addition to or a modification of source text.

- Source: lee faker sang-hyeok (hangul: 이상혁) is a league of legends esports player currently mid laner and part owner at t1
- Target: **L**ee **"F**aker**"** **S**ang-hyeok (**H**angul: 이상혁) is a **L**eague of **L**egends esports player**,** currently mid laner and part owner at **T**1**.**

While it is possible that a different selection yield better results, our selection reflects frequency (Sun and Wang, 2019) as well as syntactic significance (Bayraktar et al., 1998; Brabanter, 2023).

Similarly to popular pre-training objectives like MLM, autoregressive language modeling, and next-sentence prediction, the objective requires no human input. The objective is also architecture-agnostic and can be easily modified as appropriate.

From an internal database of English news articles, accessed between January 2022 and August 2023, we collected a total of 437,031 article excerpts, which are non-overlapping parts separated by a limiting word count of 150. One thousand excerpts each are used as the development and test sets, while the remaining 435,031 excerpts are used for training.

2.2 Experimental setup

Our experiments involve two stages. In the first stage, we take the pre-trained weights of the T5-base[1] model (Raffel et al., 2019), and perform additional pre-training on the PR objective to produce PR-T5. Then, in the second stage, we fine-tune PR-T5 on downstream tasks and datasets.

In the first stage, the model f is given the "cleared-formatting" token sequence x comprising of tokens x_t and optimized to predict the original,

[1] See Appendix C.1 for selection details and objective performance

fully punctuated and capitalized text y comprising of tokens y_t as described in Section 2.1. However, since there is textual overlap between x and y, assuming trivial copy error rate, we can write the model f as a predictor of capitalization and punctuation information $m_t = y_t - x_t$:

$$m_t = f(x, y_{<t}) = \begin{cases} \phi \\ \text{addPunct}(x_t, \theta) \\ \text{addCap}(x_t, \theta) \end{cases}$$

Thus, the effective loss is as follows:

$$\mathcal{L} \approx -\frac{1}{N} \sum_{t=1}^{N} \log P \left(m_t \mid x, y_{<t} \right).$$

In the second stage, we fine-tune PR-T5 and measure the effects of punctuation restoration in downstream tasks. We measure effects across 13 datasets that represent 7 tasks[2] and across 3 settings: generative, discriminative , and multi-task. In the generative setting, fine-tuned PR-T5 makes entity or tag predictions via autoregressive generation. We conduct 16 experiments in the generative setting, with 13 datasets from 7 tasks. In the multitask setting, fine-tuned PR-T5 is trained to make predictions for two tasks at once, namely NER and Open Information Extraction (OpenIE). We conduct 1 experiment in the multitask setting, with 2 datasets from 2 tasks. Generative and multitask predictions are illustrated in Table 5. In the discriminative setting, PR-T5's decoder block is replaced with a classification head, as described in Appendix A.1 and Figure 1. We conduct 1 experiment in the discriminative setting, with 1 dataset from 1 task. We fine-tune the publicly available pre-trained T5 weights on the same downstream tasks and use their performance as comparison baseline for all three settings. We publicly release our architecture, training, and inference code.

3 Results

We measure the effects of punctuation restoration as an additional pre-training objective on downstream tasks on `t5-base`, with the four behaviors outlined in Section 1 in mind. In this section, we find direct evidence that this method helps mitigate three out of four behaviors we describe in Section 1.

We report our results in Tables 1, 2, 3. Each reported value of precision, recall, and F1 represents an average over the same 5 seed initializations, with the exception of discriminative NER, where we analyze 15 seed initializations.

3.1 Structure information encoding and use

In all 18 experiments across dataset, task, and setting, PR-T5 reports improved performance over T5 baselines. Among them, 16 experiments report improvements ▲$\geq$ .02, and 10 experiments ▲$\geq$ .05 (Tables 1, 2, 3). This is evidence that punctuation restoration makes available a nontrivial amount of structure information that previously may have been unavailable or underutilized, mitigating behavior 4 from Section 1.

3.2 Performance stability and out-of-distribution generalization

An out-of-distribution evaluation measures performance on a dataset that represents the same task but comes from a different source than the training dataset (e.g. evaluating on CaRB (Bhardwaj et al., 2019) after fine-tuning on OIE2016 (Stanovsky and Dagan, 2016)). It is an effective measure of robustness of a representation, as fine-tuned models often learn the dataset, rather than learning the task (Gururangan et al., 2018; McCoy et al., 2019; Serrano et al., 2023). We compare out-of-distribution generalization ability of PR-T5 to that of T5 in 5 experiments across NER, OpenIE, Chunking, and POS tagging, where we observe ▲$\geq$ .05 increase in 4 of them (Table 1). This is evidence that punctuation restoration improves out-of-distribution generalization, mitigating behavior 3 in Section 1.

In addition, we observe that punctuation restoration reduces performance instability. Compared to T5, PR-T5's distribution of NER performance across initialization seeds is narrower. Minimum-maximum range (▼.04) and standard deviation (▼23%) both decrease with additional pre-training in PR, as reported in Table 3. The results support our hypothesis that punctuation restoration increases stability across initialization seed and training dataset order, mitigating behavior 2 discussed in Section 1.

4 Discussion

Results from Section 3 support our hypothesis that complementing MLM with a more structure-related objective improves structure understanding. In particular, we use a PR objective, described in Section 2 and evaluate with various structure-related tasks.

[2] See Appendix B for task and dataset details

Task	Training set	Evaluation set	t5-base			+ PR			Δ
			P	R	F1	P	R	F1	F1
NER	Econ-mNER	ID	.69	.65	.67	.90	.89	.89	▲.22
		Econ-sNER	.67	.76	.71	.74	.81	.77	▲.06
	GENIA	ID	.57	.73	.64	.64	.76	.69	▲.05
	CoNLL03	ID	.89	.90	.89	.92	.92	.92	▲.03
	ontonotes	ID	.87	.88	.88	.91	.91	.91	▲.03
OpenIE	EconIE-PRO	ID	.47	.43	.45	.60	.63	.62	▲.17
		CaRB	.22	.16	.19	.62	.42	.50	▲.31
	OIE2016	ID	.16	.19	.18	.19	.19	.19	●.01
		CaRB	.10	.15	.12	.26	.27	.27	▲.15
Chunking	CoNLL00	ID	.94	.94	.94	.96	.96	.96	▲.02
		CoNLL03	.41	.41	.41	.41	.42	.42	●.01
SRL	CoNLL12	ID	.75	.79	.77	.84	.86	.85	▲.08
SBD	PTB	ID	.97	.72	.81	.98	.98	.98	▲.17
POS	CoNLL00	ID	.96	.96	.96	.98	.98	.98	▲.02
		CoNLL03	.74	.87	.79	.84	.88	.86	▲.07
RE	TACRED	ID			.67			.83	▲.16

Table 1: Our main results where we compare t5-base model to PR-t5-base (+PR). ID denotes in-distribution evaluation on a dataset from the same source as the training set. See Appendix B for dataset details.

	t5-base (joint)			+ PR			Δ
	P	R	F1	P	R	F1	F1
NER	.86	.84	.85	.87	.86	.87	▲.02
OIE	.57	.60	.58	.60	.62	.61	▲.03

Table 2: Multitask (Econ-mNER, EconIE-PRO) performance.

	t5-base (EO)			+ PR			Δ
	P	R	F1	P	R	F1	F1
min	.67	.91	.78	.74	.90	.82	▲.04
max	.88	. 94	.91	.90	.94	.91	●.00
avg	.78	.93	.85	.83	.92	.88	▲.03
sdev	.061	.009	.035	.048	.010	.027	▼.008

Table 3: Discriminative Econ-mNER performance.

While it is difficult to investigate the exact mechanism of how additional training on punctuation restoration improves learned representations, we attempt to provide an explanation.

In Section 1, we analyze that current methods for representation learning during the pre-training stage lack sufficient signal, and hypothesize additional training with a structure-sensitive objective should improve structure understanding. Much like how prosody helps disambiguate syntax in human speech processing (Price et al., 1991; Kahn et al., 2005), punctuation can be a useful guide in syntax disambiguation, and eventually toward forming a robust representation of text. Punctuation marks often indicate syntactic or semantic boundaries (Briscoe, 1996; Bayraktar et al., 1998). Optimizing a computational system to predict punctuation allows it to predict syntactic and semantic boundaries, even in the absence of punctuation in the original text. Sufficient training in restoring punctuation can imitate effects of explicitly providing a parse, facilitating natural language understanding via a stronger understanding of sentence structure.

Performance improvement from PR is not limited to a specific dataset, task, and setting[3]. and represents an overall increase in representation robustness, as we observe out-of-distribution performance jump in NER, OpenIE, and chunking. Because of the wide range of experiments in which improvement is observed, we interpret this to be a general improvement of structure understanding rather than fortunate task-specific artifacts from the additional training.

Our methods yield a more reliable and robust representation that can be easily implemented and do not interfere with architectural additions. PR can be applied to reinforce structure understanding and improve robustness of learned representations regardless of model choice, or task-specific engineering policy. The effective objective requires no supervision, and one can construct a training corpus with little computational or manual resources.

[3]And decoding method, discussed in Appendix B

Limitations

The idea of structure understanding reinforcement via punctuation restoration is still young–many decisions relevant to the learning objective in this paper, including selection of punctuation marks and source of learning corpus warrant additional investigation in future work. Our set of training hyper-parameters also will benefit from additional attention.

Among the 4 behaviors discussed in Section 1, we find direct evidence that punctuation restoration mitigates only three of them. While we predict that unsupervised structure learning via objectives like PR can help mitigate the reversal (factorization) curse, this will need explicit verification.

While our experiments show promise in base-sized NLU models for English, its effects in larger models, implications to generative or conversational systems, and generalization to other languages and thus language-agnostic nature also need to be verified.

It is also likely that punctuation restoration is not the only unsupervised learning objective that can be used to improve the representation learning stage of training NLP systems. Other forms of unsupervised structure learning, possibly simpler and more effective methods than punctuation restoration, as well as optimizations on objective combination (e.g. with word prediction methods) should be studied in future work.

Responsible research statement

We use OpenAI's GPT-3.5 Turbo (Brown et al., 2020) as a punctuation restoration performance baseline, and as a debugging assistant during the project's technical implementation.

The Econ-mNER dataset was annotated by paid, full-time employees who are trained linguists knowledgeable about their work and the dataset's downstream use. They are compensated similarly to the region's 2021 median income level. Their work has been reviewed by an internal board to not contain any personally identifiable information. Other internal datasets did not require manual annotation.

Acknowledgements

This work was performed during the authors' time at NCSOFT. We thank Yerang Kim, Tatsuya Aoyama, and Ethan Wilcox for their helpful comments. Any errors remain our own.

References

Tanvirul Alam, Akib Khan, and Firoj Alam. 2020. Punctuation restoration using transformer models for high- and low-resource languages. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, pages 132–142.

Xuefeng Bai, Jialong Wu, Yulong Chen, Zhongqing Wang, and Yue Zhang. 2023. Constituency parsing using llms. *arXiv preprint arXiv:2310.19462*.

Murat Bayraktar, Bilge Say, and Varol Akman. 1998. An analysis of english punctuation: The special case of comma. *International Journal of Corpus Linguistics*, 3(1):33–57.

Lukas Berglund, Meg Tong, Maximilian Kaufmann, Mikita Balesni, Asa Cooper Stickland, Tomasz Korbak, and Owain Evans. 2024. The reversal curse: LLMs trained on "a is b" fail to learn "b is a". In *The Twelfth International Conference on Learning Representations*.

Sangnie Bhardwaj, Samarth Aggarwal, and Mausam. 2019. CaRB: A crowdsourced benchmark for open IE. 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 6262–6267, Hong Kong, China. Association for Computational Linguistics.

Philippe De Brabanter. 2023. Quotation does not need marks of quotation. *Linguistics*, 61(2):285–316.

Ted Briscoe. 1996. The syntax and semantics of punctuation and its use in interpretation. In *Proceedings of the Association for Computational Linguistics Workshop on Punctuation*, pages 1–7. Citeseer.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Tiago Barbosa de Lima, Vitor Rolim, André C.A. Nascimento, Péricles Miranda, Valmir Macario, Luiz Rodrigues, Elyda Freitas, Dragan Gašević, and Rafael Ferreira Mello. 2024. Towards explainable automatic punctuation restoration for portuguese using transformers. *Expert Systems with Applications*, 257:125097.

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.

Andrew Drozdov, Patrick Verga, Mohit Yadav, Mohit Iyyer, and Andrew McCallum. 2019. Unsupervised latent tree induction with deep inside-outside recursive auto-encoders. 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 1129–1141, Minneapolis, Minnesota. Association for Computational Linguistics.

Yupei Du and Dong Nguyen. 2023. Measuring the instability of fine-tuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6209–6230, Toronto, Canada. Association for Computational Linguistics.

Hao Fei, Shengqiong Wu, Yafeng Ren, Fei Li, and Donghong Ji. 2021. Better combine them together! integrating syntactic constituency and dependency representations for semantic role labeling. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 549–559, Online. Association for Computational Linguistics.

Olga Golovneva, Zeyuan Allen-Zhu, Jason E Weston, and Sainbayar Sukhbaatar. 2024. Reverse training to nurse the reversal curse. In *First Conference on Language Modeling*.

Agustin Gravano, Martin Jansche, and Michiel Bacchiani. 2009. Restoring punctuation and capitalization in transcribed speech. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4741–4744.

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

Keqing He, Shuyu Lei, Yushu Yang, Huixing Jiang, and Zhongyuan Wang. 2020. Syntactic graph convolutional network for spoken language understanding. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2728–2738.

Xiaosai Huang, Jing Li, Jia Wu, Jun Chang, Donghua Liu, and Kai Zhu. 2024. Flexibly utilizing syntactic knowledge in aspect-based sentiment analysis. *Information Processing & Management*, 61(3):103630.

Jeremy G Kahn, Matthew Lease, Eugene Charniak, Mark Johnson, and Mari Ostendorf. 2005. Effective use of prosody in parsing conversational speech. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 233–240.

J.-D. Kim, T. Ohta, Y. Tateisi, and J. Tsujii. 2003. GENIA corpus—a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(1):180–182.

Ouail Kitouni, Niklas Nolte, Diane Bouchacourt, Adina Williams, Mike Rabbat, and Mark Ibrahim. 2024. The factorization curse: Which tokens you predict underlie the reversal curse and more. In *Advances in Neural Information Processing Systems*, volume 37, pages 112329–112355. Curran Associates, Inc.

Minho Lee, Junghyun Min, Woochul Lee, and Yeonsoo Lee. 2024. Structured language generation model for robust structure prediction. *arXiv preprint arXiv:2402.08971*.

Frederick Liu, Terry Huang, Shihang Lyu, Siamak Shakeri, Hongkun Yu, and Jing Li. 2022. Enct5: A framework for fine-tuning t5 as non-autoregressive models.

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.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: The penn treebank. *Comput. Linguist.*, 19(2):313–330.

R. Thomas McCoy, Junghyun Min, and Tal Linzen. 2020. BERTs of a feather do not generalize together: Large variability in generalization across models with similar test set performance. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 217–227, Online. Association for Computational Linguistics.

Tom 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*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Junghyun Min, R. Thomas McCoy, Dipanjan Das, Emily Pitler, and Tal Linzen. 2020. Syntactic data augmentation increases robustness to inference heuristics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2339–2352, Online. Association for Computational Linguistics.

Omar Momen, David Arps, and Laura Kallmeyer. 2023. Increasing the performance of cognitively inspired data-efficient language models via implicit structure building. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural*

Language Learning, pages 327–338, Singapore. Association for Computational Linguistics.

Vasile Păiş and Dan Tufiş. 2022. Capitalization and punctuation restoration: a survey. *Artificial Intelligence Review*, 55(3):1681–1722.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. 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 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *Joint Conference on EMNLP and CoNLL - Shared Task*, pages 1–40, Jeju Island, Korea. Association for Computational Linguistics.

Patti J Price, Mari Ostendorf, Stefanie Shattuck-Hufnagel, and Cynthia Fong. 1991. The use of prosody in syntactic disambiguation. *the Journal of the Acoustical Society of America*, 90(6):2956–2970.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer.

Devendra Sachan, Yuhao Zhang, Peng Qi, and William L Hamilton. 2021. Do syntax trees help pre-trained transformers extract information? In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2647–2661.

Laurent Sartran, Samuel Barrett, Adhiguna Kuncoro, Miloš Stanojević, Phil Blunsom, and Chris Dyer. 2022. Transformer grammars: Augmenting transformer language models with syntactic inductive biases at scale. *Transactions of the Association for Computational Linguistics*, 10:1423–1439.

Sofia Serrano, Jesse Dodge, and Noah A. Smith. 2023. Stubborn lexical bias in data and models. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8131–8146, Toronto, Canada. Association for Computational Linguistics.

Yikang Shen, Yi Tay, Che Zheng, Dara Bahri, Donald Metzler, and Aaron Courville. 2021. StructFormer: Joint unsupervised induction of dependency and constituency structure from masked language modeling. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7196–7209, Online. Association for Computational Linguistics.

Gabriel Stanovsky and Ido Dagan. 2016. Creating a large benchmark for open information extraction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2300–2305, Austin, Texas. Association for Computational Linguistics.

Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. 2018. Linguistically-informed self-attention for semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5027–5038, Brussels, Belgium. Association for Computational Linguistics.

Kun Sun and Rong Wang. 2019. Frequency distributions of punctuation marks in english: Evidence from large-scale corpora. *English Today*, 35(4):23–35.

Erik F. Tjong Kim Sang and Sabine Buchholz. 2000. Introduction to the CoNLL-2000 shared task chunking. In *Fourth Conference on Computational Natural Language Learning and the Second Learning Language in Logic Workshop*.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Chenguang Wang, Xiao Liu, Zui Chen, Haoyun Hong, Jie Tang, and Dawn Song. 2022. DeepStruct: Pretraining of language models for structure prediction. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 803–823, Dublin, Ireland. Association for Computational Linguistics.

Xinyu Wang, Yong Jiang, Nguyen Bach, Tao Wang, Zhongqiang Huang, Fei Huang, and Kewei Tu. 2021. Automated concatenation of embeddings for structured prediction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2643–2660, Online. Association for Computational Linguistics.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, Mohammed El-Bachouti, Robert Belvin, and Ann Houston. 2013. Ontonotes.

Shengqiong Wu, Hao Fei, Yafeng Ren, Donghong Ji, and Jingye Li. 2021. Learn from syntax: Improving pair-wise aspect and opinion terms extraction with rich syntactic knowledge. In *Proceedings of the Thirtieth International Joint Conference on Artificial*

Intelligence, IJCAI-21, pages 3957–3963. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Zhuofeng Wu, Sinong Wang, Jiatao Gu, Madian Khabsa, Fei Sun, and Hao Ma. 2020. Clear: Contrastive learning for sentence representation.

Yifeng Xie, Zhihong Zhu, Xuxin Cheng, Zhiqi Huang, and Dongsheng Chen. 2023. Syntax matters: Towards spoken language understanding via syntax-aware attention. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 11858–11864.

Yadollah Yaghoobzadeh, Soroush Mehri, Remi Tachet des Combes, T. J. Hazen, and Alessandro Sordoni. 2021. Increasing robustness to spurious correlations using forgettable examples. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3319–3332, Online. Association for Computational Linguistics.

Yue Zhang, Rui Wang, and Luo Si. 2019. Syntax-enhanced self-attention-based semantic role labeling. 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 616–626, Hong Kong, China. Association for Computational Linguistics.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*, pages 35–45.

Haoyu Zhao, Abhishek Panigrahi, Rong Ge, and Sanjeev Arora. 2023. Do transformers parse while predicting the masked word? In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16513–16542, Singapore. Association for Computational Linguistics.

Xiang Zhou, Yixin Nie, Hao Tan, and Mohit Bansal. 2020. The curse of performance instability in analysis datasets: Consequences, source, and suggestions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8215–8228, Online. Association for Computational Linguistics.

A Additional details on experimental setup

We train the model on the punctuation restoration objective for 40 epochs, before fine-tuning with supervised datasets for downstream tasks. The experiments are run on a single V100 GPU with 32GB VRAM, with half precision and gradient accumulation enabled at 16. Our choice of hyper-parameters are as follows: batch size 32, maximum sequence length 256, learning rate 3e-4, maximum grad norm 0.5, and Adam epsilon 1e-8. Number of fine-tuning epochs was 10, with the exception of SRL, which is fine-tuned for 1 epoch only. The additional pre-training lasts about 2 weeks, while the length of each epoch of training varies across datasets between 10 minutes and around 2 hours.

A.1 Discriminative approach

While there exist sophisticated attempts to incorporate the decoder layers in producing a discriminative model from a pre-trained encoder-decoder architecture (Liu et al., 2022), we use a simple architecture where we forgo the decoder block and place a `T5ClassificationHead` on top of the encoder block of the T5 model. That is, we take the hidden state output from model's encoder and use it as input to the classification head. An illustration of the model architecture is shown in Figure 1. After additional pre-training on punctuation restoration objective, the decoder block of the `t5-base` model is removed and a newly initialized classification head is placed on top of the encoder block. The architecture is comparable to those of BERT-like encoder-only models. Even by retaining weights from the encoder blocks only, we observe that additional unsupervised structure learning via punctuation restoration results in downstream task performance improvement.

A.2 Joint multitask generative approach

The joint multitask approach, where we focus on open information extraction using the EconIE-PRO dataset and NER using the Econ-mNER dataset, is similar to the generative approach. The input sequence is identical to the experiments from Section 3, but the output sequence is a concatenation of output sequences from the two datasets, as illustrated in Table 5.

Model architecture	P	R	F1
ChatGPT 0-shot*	.75	.71	.73
`t5-small`	.91	.86	.88
`t5-base`	.93	.92	.93
`t5-large`	.94	.93	.93

Table 4: Punctuation restoration performance after 50 epochs (small), 40 epochs (base), and 20 epochs (large) of training respectively. *Measured on a small subset of the punctuation restoration evaluation dataset.

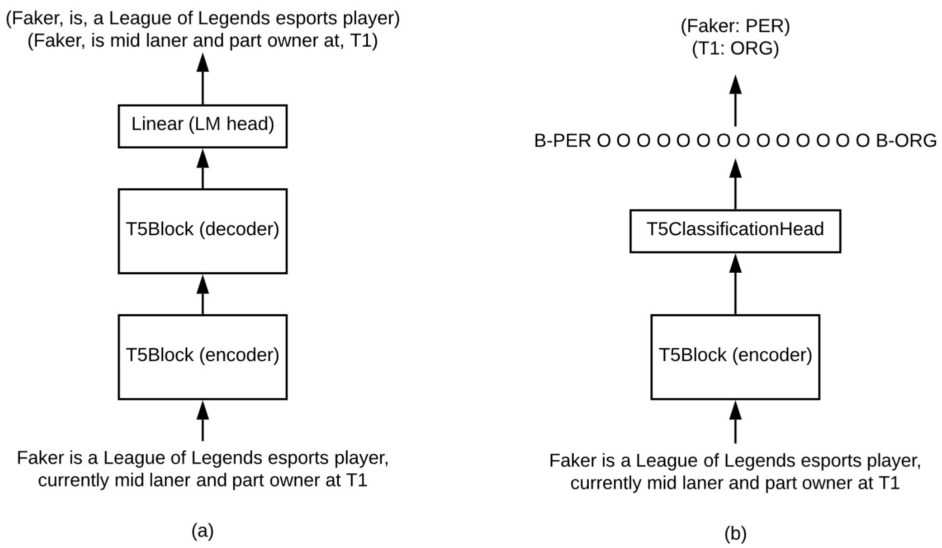

Figure 1: (a) The `t5` architecture for a generative, text-to-text approach to NLP tasks. Here, we illustrate open information extraction. (b) A modification to the `t5` architecture to allow a discriminative approach to NLP tasks. Here, we illustrate named entity recognition.

B Additional details on dataset

We use a suite of structure-related NLP tasks to measure model structure understanding. Relevant tasks include named entity recognition (NER), sentence boundary detection (SBD), open information extraction (OpenIE), chunking, semantic role labeling (SRL), part-of-speech tagging, and relation classification. Our selection mostly follows that from Wang et al. (2021) and Lee et al. (2024). We use both public and internal datasets, and check for in- and out-of-distribution generalization. A full list of datasets for each task is shown in Table 6. In the main body of the paper, we discuss effects of PR across task, dataset, and setting. Here, we discuss another variable across which PR is effective: decoding method.

B.1 Entity generation tasks

NER, OpenIE, SRL, and relation classification are entity generation tasks, where fine-tuned models autoregressively generate entity objects. For example, `(Faker: PER)`, `(Faker, is, a League of Legends esports player)`, `(Faker, employeeAt, T1)` are NER, OpenIE, and relation classification examples, respectively. The order in which entities are generated does not affect evaluation in the case of entity generation tasks.

Source	Faker is a League of Legends esports player, currently mid laner and part owner at T1.
OpenIE	(Faker, is, a League of Legends esports player) (Faker, is mid laner and part owner at, T1)
NER	(Faker: PER) (T1: ORG)
Multitask	(Faker: PER) (Faker, is, a League of Legends esports player) (Faker, is mid laner and part owner at, T1) (T1: ORG)

Table 5: Example output from generative NER, OpenIE, and multitask models.

B.2 Tag sequence generation tasks

Chunking and POS tagging are tag sequence generation tasks, where fine-tuned models auto-regressively generate tag sequences. `"NP VP ADVP PP NP NP NP"` and `"NP VBZ DT NP IN NP"` are example sequences of chunking and POS tagging, respectively.

B.3 Sequence generation tasks

Punctuation restoration and sentence boundary detection are sequence generation tasks. Fine-tuned models auto-regressively generate natural text sequences, with predefined tags to perform the task. For example, a sentence boundary detection model would generate a [<s>] token between sentences, given a passage.

Task	Dataset	Source	Task type
Internal datasets			
PR	finPR	Rule-based tagging on finance news	Seq. gen.
NER	Econ-mNER	Manual tagging on finance news and corporate filings	Ent. gen., Tok. cls.
	Econ-sNER	Semi-supervised tagging on finance news	Ent. gen.
OpenIE	EconIE-PRO	Rule-based tagging on finance news, predicate range optimized	Ent. gen.
Public datasets			
NER	GENIA	Kim et al. (2003)	Ent. gen.
	CoNLL 2003	Tjong Kim Sang and De Meulder (2003)	Ent. gen.
	ontonotes	Weischedel et al. (2013)	Ent. gen.
SBD	PTB	Marcus et al. (1993)	Seq. gen.
OpenIE	OIE2016	Stanovsky and Dagan (2016)	Ent. gen.
	CaRB	Bhardwaj et al. (2019)	Ent. gen.
Chunk, POS	CoNLL 2000	Tjong Kim Sang and Buchholz (2000)	Tag gen.
	CoNLL 2003	Tjong Kim Sang and De Meulder (2003)	Tag gen.
SRL	CoNLL 2012	Pradhan et al. (2012)	Ent. gen.
ORE	TACRED	Zhang et al. (2017)	Ent. gen.

Table 6: We use a total of 14 datasets across 8 tasks, including punctuation restoration. Four are internal datasets, while the rest are publicly available.

B.4 Token classification tasks

NER in the discriminative setting is a token classification task. Given a sentence of length n, the fine-tuned model outputs an array of length n, each element of which represents whether its corresponding token is part of a named entity. For example, one from a tag set such as `[O, B-PER, I-PER, B-LOC, I-LOC, B-ORG, I-ORG]`, as illustrated in Figure 1.

C Additional details on results

In our results, improvements from PR persist across decoding methods–entity generation in NER, OpenIE, SRL, and relation classification; tag sequence generation in chunking and POS tagging; sequence generation in sentence boundary detection; and token classification in discriminative NER.

C.1 Objective results

Punctuation restoration is no trivial task (Gravano et al., 2009; Alam et al., 2020). Should our hypothesis hold, it is likely that syntactic signals from punctuation restoration transfer more effectively in models with stronger punctuation restoration performances. We experiment with three sizes of the T5 architecture. We consider `t5-small`, `t5-base`, and `t5-large`. Table 4 includes their punctuation restoration performance, in addition to ChatGPT's (Brown et al., 2020) zero-shot performance as a reference point, which shows that the objective is nontrivial.

Across the T5 models, there is some correlation between size and punctuation restoration performance. Because the performance gap between `t5-base` and `t5-large` models is small (●.00), while gap between `t5-small` and `t5-base` more significant (▲.05), we use the `t5-base` model for our experiments.

We also note that our selection of the T5 model is due to its ability to perform both generative and discriminative tasks after single pre-training.

C.2 Joint multitask generative setting

Similarly to the generative approach, we observe that additional unsupervised structure learning via punctuation restoration results in downstream task performance improvement (▲.02 NER and ▲.03 OpenIE). While PR-T5 multi-task performance slightly degrades compared to its single-task generative setting (▼.02 NER and ●.01 OpenIE), multitask-T5 outperforms single task-T5 on EconIE-PRO, an open information extraction dataset (▲.13).

C.3 Discriminative setting

Given the results from the single-task generative approach, the transfer from punctuation restoration to multi-task generative approach may be no big surprise, as there is no drastic difference between the generative nature of the two approaches. However, we report that our improved representations from punctuation restoration non-trivially transfers to the discriminative approach as well, where the decoder block is removed from the model, as illustrated in Figure 1. Although the maximum performance for T5 and PR-T5 are similar at .91 (●.00), there is a significant difference in the minimum, at .78 and .82, respectively (▲.04). Punctuation restoration results in not only higher performance, but also more consistent and stable sets across different initializations.