Punctuation Restoration Improves Structure Understanding without Supervision

Junghyun Min and **Minho Lee** and **Woochul Lee** and **Yeonsoo Lee**
NC Research / Seongnam, Gyeonggi, Korea
{hyun1, minolee, darkgeo, yeonsoo}@ncsoft.com

Abstract

Unsupervised learning objectives like language modeling and de-noising constitute a significant part in producing pre-trained models 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 transfer of linguistic structure knowledge to computational systems with currently popular pre-training objectives. We show that punctuation restoration as a learning objective improves in- and out-of-distribution performance on structure-related tasks like named entity recognition, open information extraction, chunking, and part-of-speech tagging. Punctuation restoration is an effective learning objective that can improve structure understanding and yield a more robust structure-aware representations of natural language.

1 Introduction

The current framework of natural language processing systems, described by Linzen (2020) as the PAID paradigm, consists of two production stages: unsupervised representation learning and task-specific engineering. Modern transformer-based systems that follow the framework (Devlin et al., 2019; Raffel et al., 2019; Radford et al., 2018; Peters et al., 2018) report high performance in various natural language understanding tasks, often matching or exceeding human performance baselines (Wang et al., 2018, 2019). However, there is ample evidence that current unsupervised representation learning yields weak structure understanding and brittle generalization abilities. In classification systems, we observe unstable outcome despite consistent input and reliance on shallow heuristics while processing unfamiliar input (McCoy et al., 2020b; Zhou et al., 2020). In generative and conversational systems, we observe stagnant natural language understanding performance despite drastic increase in conversational performance (Zhong et al., 2023), and failure to generalize sentences like "A equals B" to "B equals A" (Berglund et al., 2023).

These are examples of weak structure understanding in language model based NLP systems. While it is difficult to pinpoint the exact source of these weaknesses or even disentangle between the effects of unsupervised pre-training and task specific engineering, the pre-training stage is at least partially attributable for these behaviors, and there exists room for improvement (Zhou et al., 2020; Min et al., 2020). We believe word prediction tasks like auto-regressive (Radford et al., 2018), masked (Devlin et al., 2019), and perturbed (Raffel et al., 2019) language modeling may be insufficient to acquire robust representations that contain strong understanding of syntactic and semantic structure. We hypothesize that an additional unsupervised learning objective that focuses on capturing structure within natural language will improve structure understanding, measured by in-distribution (test set from the same source as training set) and out-of-distribution (test set from a different source than training set) performance in structure-related NLP tasks like chunking, information extraction, semantic role labeling, named entity recognition, sentence boundary detection, and part-of-speech tagging.

This paper aims to test this hypothesis, using an unsupervised learning objective that reinforces structure understanding in language models. One nontrivial signal for syntactic and semantic structure in natural language is punctuation (Briscoe, 1996; Nunberg, 1990; Dale, 1991), which can also be an effective parsing constraint that aids grammar induction in web mark-up text (Spitkovsky et al., 2010). During human speech processing, syntactic disambiguation and grammar induction are facilitated by prosody (Kahn et al., 2005; Price et al., 1991), which is analogous to punctuation in written text. Previously, punctuation has been used for grammar induction to improve unsupervised dependency parsing (Spitkovsky et al., 2011). Punctuation restoration is itself also a popular downstream task, especially for polishing output text from automatic speech recognition systems (Gravano et al., 2009; Alam et al., 2020, *inter alia*) but has not been studied as a transferable language modeling objective.

Here, we propose punctuation restoration as the structure-oriented learning objective, which we describe in detail in Section 3. Our results show additional pre-training with the punctuation restoration objective leads to improvements in various structure-related NLP task performance in both discriminative and generative approaches, supporting our hypothesis. Furthermore, this finding suggests that there is room for improvement in the unsupervised pre-training stage in the current paradigm of producing natural language processing systems.

Our contribution is twofold:

1. We suggest a novel research direction in unsupervised transfer learning beyond word prediction

2. We propose an unsupervised learning objective that yields robust structure understanding

2 Structure understanding

Understanding of language structure is vital in both human and machine language processing. While human language acquisition and modern machine representation learning take a similar approach–acquisition of structure via implicit structural signals in an unsupervised setting, their outcomes are different, highlighted by poor generalization abilities and high computational costs of machine language processing systems.

2.1 Human acquisition of structure understanding

Despite the unsupervised and sparse nature of their linguistic stimuli, human learners are able to obtain robust representations that generalize to unfamiliar inputs reliably and with remarkable efficiency. Braine et al. (1971) provide a plethora of examples where children do not respond to negative reinforcement in their corpus. However, even without explicit supervision, humans are able to generalize their linguistics knowledge to novel structures and utterances (Sprouse et al., 2013). Moreover, this is accomplished with remarkable efficiency–Roy et al. (2015) analyze that children hear or produce approximately 8 million words over a 15-month period, which amounts to around 13-14 million tokens. Linzen (2020) acknowledges that NLP tasks or languages with a similar range of available data are often dubbed "low-resource."

2.2 Pre-training to acquire structure understanding

Computational systems struggle to obtain reliable representations of structure, given their lack of human-like inductive bias (Linzen et al., 2016; McCoy et al., 2020a). Modern language models propose word prediction objectives for representation learning–they acquire natural language representation by predicting words that are most likely to appear in the masked, perturbed, or next-in-sequence slot (Devlin et al., 2019; Raffel et al., 2019; Radford et al., 2018; Yang et al., 2020). BERT (Devlin et al., 2019) employs masked language modeling, where random words in a sentence are masked, and the model is tasked with predicting those masked tokens. The Text-to-Text Transfer Transformer (T5; Raffel et al., 2019) utilizes a de-noising objective, where a portion of input sentences is corrupted, and the model is trained to reconstruct the original text. ELECTRA (Clark et al., 2020) introduces a novel approach through corruption classification, where a subset of tokens is replaced with incorrect ones, and the model distinguishes between genuine and corrupted tokens. The Generative Pre-trained Transformer (GPT; Radford et al., 2018) employs an auto-regressive language modeling objective, predicting the next word in a sequence given the preceding context.

2.3 Other methods for structure understanding

In addition to structure learning during the pre-training stage, various work suggest methods applicable after it. Approaches related to dataset adjustment account for a significant portion. Gunasekar et al. (2023) observe that training on textbook quality data reduces the need for scaling while maintaining performance. Yaghoobzadeh et al. (2021) proposes recursion on forgettable examples to curb system reliance on spurious correlations and focus on syntactic and semantic signal. Min et al. (2020)

introduce a simple yet effective human-in-the-loop adversarial augmentation framework that improves general syntactic structure understanding. Clark et al. (2019) suggest performing "additional pre-training" on the supervised Multi-Genre Natural Language Inference dataset (Williams et al., 2018) transfers cross-sentence structure understanding and thus improves downstream performance on their Boolean QA dataset. Other efforts include augmenting input explicitly with syntax by providing constituency or dependency parsing information (Pradhan et al., 2005; Zhang et al., 2019; Lepori et al., 2020) or via joint inference (Punyakanok et al., 2008), detecting a domain-specialized subspan of input text to process them separately (Park et al., 2023), and increasing retrieval relevancy by applying additional constraints to encourage learning and prediction of task-specific input-output structure (Lee et al., 2024). We also note methods that facilitate distillation of structural knowledge from a more robust teacher model, such as attention alignment during distillation (Jin et al., 2024).

3 Objective design and experimental setup

3.1 Objective design

The punctuation restoration objective predicts cleared punctuation marks and capitalization. In our implementation, we predict the following set of punctuation marks: the comma **,**, the period **.**, the single-quotation mark **’**, and the double-quotation mark **"**, along with capitalization, as shown below. Boldface indicates an addition 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**.**

We do not introduce mask tokens that trigger predictions, because we want the learners to be able to infer punctuation marks and capitalization (and hence language structure) from raw text only. We also acknowledge our selection of punctuation marks to restore is arbitrary, and it is possible that a different selection yield better results.

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. Sources include major news outlets like CNN and Reuters. One thousand excerpts each are used as the development and test sets, while the remaining 435,031 excerpts are used for training.

The raw excerpts serve as target text. To create source text, we first normalize punctuation marks, then remove our four selected punctuation marks, then apply the `.lowercase()` transformation. While we intend to produce a training dataset entirely in English, we did not check for this, and it is possible the training data include non-English words, phrases, or articles.

3.2 Experimental setup

We treat punctuation restoration as additional training before fine-tuning on the target datasets. We experiment with three approaches–a single-task generative approach with the conditional language modeling head, a joint multi-task generative approach, and a discriminative approach with a classification head. For all approaches, we use the publicly available `t5-base` model architecture and checkpoint from Hugging Face Transformers’ `T5ForConditionalGeneration` module.

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 RAM, with half precision and gradient accumulation enabled at 16. Our arbitrary 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.

We follow Raffel et al. (2019)’s framework of transfer learning in text-to-text tasks for the generative approach and Radford et al. (2018)’s framework of generative pre-training followed by discriminative fine-tuning for the discriminative approach. Unlike the generative approach, the discriminative approach requires some modification from the original T5 implementation, illustrated in Figure 1.

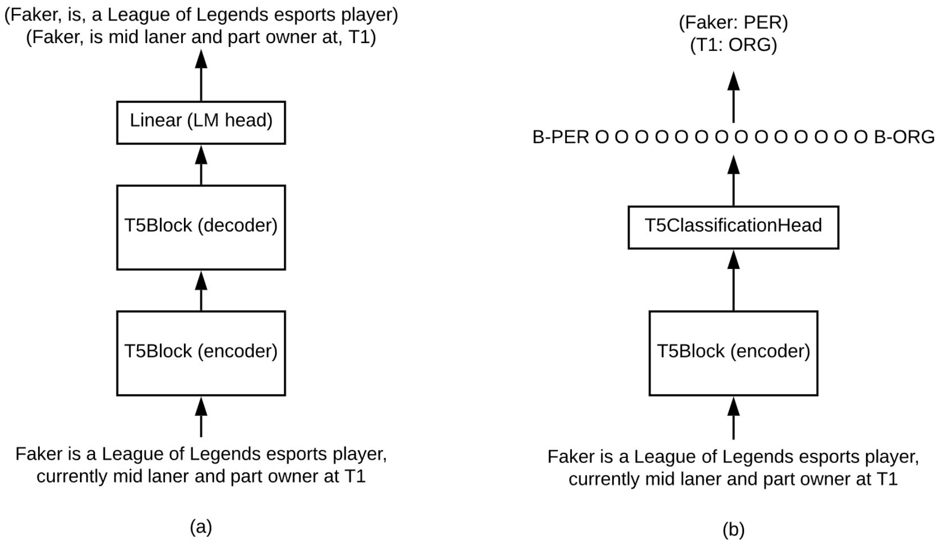
(Faker, is, a League of Legends esports player)
(Faker, is mid laner and part owner at, T1)
Linear (LM head)
T5Block (decoder)
T5Block (encoder)
Faker is a League of Legends esports player, currently mid laner and part owner at T1
(a)
(Faker: PER)
(T1: ORG)
B-PER O O O O O O O O O O O O O O B-ORG
T5ClassificationHead
T5Block (encoder)
Faker is a League of Legends esports player, currently mid laner and part owner at T1
(b)

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.

3.2.1 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.

3.3 Evaluation datasets

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. 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 1.

4 Results

We measure the effects of punctuation restoration as an addition pre-training objective on structure understanding abilities across in- and out-of-distribution performance in datasets described in Table 1. We report the results in various settings, including the generative approach following Raffel et al. (2019) in Section 4.2, the joint multitask approach in Section 4.3, and the discriminative approach following Devlin et al. (2019) in Section 4.4.

4.1 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–small, base, and large to help determine our experimental setup. Table 2 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, we use the `t5-base` model for our experiments.

4.2 Generative approach

Table 3 contains an overview of model performance on various structure related tasks with and without additional training on punctuation restoration. Each task performance represents an average over 5 runs.

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

Table 1: We use a total of 14 datasets across 8 tasks, including punctuation restoration. Four are internal datasets, while the rest are publicly available.

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 2: 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.

We observe increases in in-distribution and out-of-distribution generalization performances across the board. In particular, we note that sentence boundary detection, arguably task closest to punctuation restoration, achieves a near perfect score. Other notable takeaways from the results include out-of-distribution performance jump in open information extraction, even when in-distribution generalization improves little.

The results support that punctuation restoration is an effective and efficient addition to the current framework of natural language understanding. We interpret this as evidence for our hypothesis that an additional unsupervised learning objective that focuses on capturing structure within natural language will improves structure understanding. In addition to the generative approach taken in this section, we discuss whether this supportive behavior persists in other setting like joint multitask learning (Section 4.3) and discriminative learning (Section 4.4).

4.3 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 4.2, but the output sequence is a concatenation of output sequences from the two datasets, as illustrated in Table 6. Similarly to the generative approach, we observe that additional unsupervised structure learning via punctuation restoration results in downstream task performance improvement.

4.4 Discriminative approach

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

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

Table 3: Results from generative NER, OpenIE, and multitask models, where we compare vanilla `t5-base` model to `t5-base` with additional pre-training on punctuation restoration (+PR). ID is short for in-distribution evaluation, denoting evaluation on a dataset from the same source as the training set. Each measurement is an average over a set of 5 seeds, while training data order is not controlled.

	`t5-base` (joint)			+ PR		
	P	R	F1	P	R	F1
Econ-mNER	.86	.84	.85	.87	.86	.87
EconIE-PRO	.57	.60	.58	.60	.62	.61

Table 4: NER (Econ-mNER) and OpenIE (EconIE-PRO) performance after joint training, where we compare vanilla `t5-base` model to `t5-base` with additional pre-training on punctuation restoration (+PR). Punctuation restoration improves performance in both NER and OpenIE. Each measurement is an average over a set of 5 seeds, while training data order is not controlled.

	`t5-base` (EO)			+ PR		
	P	R	F1	P	R	F1
Econ-mNER	.78	.93	.85	.83	.92	.88

Table 5: Discriminative (Encoder-Only) Econ-mNER performance with (+PR) and without (`t5-base`) punctuation restoration as additional pre-training. Each measurement is an average over a set of 15 seeds, while training data order is not controlled.

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 6: Example output from generative NER, OpenIE, and multitask models for illustration purposes

block is removed from the model, as illustrated 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.

5 Discussion

Results from Section 4 support our hypothesis that complementing the de-noising pre-training objective with a structure-reinforcing task improves structure understanding. In particular, we use a punctuation restoration objective, described in Section 3 and evaluate with various structure-related tasks listed in Table 1. While it is difficult to investigate the exact mechanism on how additional training on punctuation restoration improves learned representations, we attempt to provide an explanation.

Providing additional syntactic or semantic information in the form of parses have proven to be effective in improving natural language understanding (Pradhan et al., 2005; Zhang et al., 2019; Lepori et al., 2020). That is, the current methods for representation learning during the pre-training stage lacks sufficient syntactic signal, and effective distillation of implicit syntactic sensitivity via additional training should improve structure understanding. Much like how prosody helps disambiguate syntactic structure in human speech processing (Kahn et al., 2005; Price et al., 1991), punctuation can be a useful guide in syntactic structure disambiguation (Spitkovsky et al., 2010), and eventually in structure understanding and forming a robust representation of text. Because punctuation often indicates syntactic or semantic boundaries, training a computational system to predict punctuation from stripped text also can train the system to predict syntactic and semantic structure within said text, even when there are no punctuation marks to be restored in the original, fully punctuated text. Sufficient training in punctuation restoration or with other markers of syntactic and semantic structure can have similar effects of explicitly providing a syntactic or semantic parse, facilitating natural language understanding via a stronger understanding of sentence structure.

Such improvements are not limited to specific domains or datasets and represent an overall increase in representation robustness, as we observe out-of-distribution performance jump in NER, OpenIE, and chunking. Improvements also 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. Because of the wide range of settings in which improvement is observed, We interpret this to a general improvement of structure understanding rather than fortunate task-specific artifacts from the additional training.

We claim that our methods are democratic in that we employ a non-intrusive unsupervised learning objective that is orthogonal to other architectural or methodological modifications. Punctuation restoration can be applied to reinforce structure understanding and improve robustness of learned representations regardless of model choice, or task-specific engineering policy. The objective requires no supervision, and one can construct a training corpus with little computational or manual resources.

6 Limitations

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

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.

Despite many unanswered questions, however, we conclude that punctuation restoration is an effective learning objective that improves structure understanding without supervision.

Responsible research statement

Our work primarily builds on T5 (Raffel et al., 2019), with the Apache License 2.0. Our model and framework is thus a Derivative Work, and also adopts the same license. We also use ChatGPT (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

We are grateful to members of the Natural Language Understanding Division and Chunghee Lee for their helpful feedback on this project and to Minji Kang for her valuable insight on prosody and human language processing. We dedicate this work to our fond memories of Junghwa Lee (1994 - 2023).

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.

Lukas Berglund, Meg Tong, Max Kaufmann, Mikita Balesni, Asa Cooper Stickland, Tomasz Korbak, and Owain Evans. 2023. The reversal curse: Llms trained on "a is b" fail to learn "b is a".

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.

Martin DS Braine et al. 1971. On two types of models of the internalization of grammars. *The ontogenesis of grammar*, 1971:153–186.

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.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. 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 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators.

Robert Dale. 1991. Exploring the role of punctuation in the signalling of discourse structure. In *Proceedings of a workshop on text representation and domain modelling: ideas from linguistics and AI*, pages 110–120. Technical University of Berlin Berlin, Germany.

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

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.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Harkirat Singh Behl, Xin Wang, Sébastien Bubeck, Ronen Eldan, Adam Tauman Kalai, Yin Tat Lee, and Yuanzhi Li. 2023. Textbooks are all you need.

Heegon Jin, Seonil Son, Jemin Park, Youngseok Kim, Hyungjong Noh, and Yeonsoo Lee. 2024. Align-to-distill: Trainable attention alignment for knowledge distillation in neural machine translation. *arXiv preprint arXiv:2403.01479*.

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.

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

Michael Lepori, Tal Linzen, and R. Thomas McCoy. 2020. Representations of syntax [MASK] useful: Effects of constituency and dependency structure in recursive LSTMs. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3306–3316, Online. Association for Computational Linguistics.

Tal Linzen. 2020. How can we accelerate progress towards human-like linguistic generalization? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5210–5217, Online. Association for Computational Linguistics.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.

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.

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, Robert Frank, and Tal Linzen. 2020a. Does syntax need to grow on trees? sources of hierarchical inductive bias in sequence-to-sequence networks. *Transactions of the Association for Computational Linguistics*, 8:125–140.

R. Thomas McCoy, Junghyun Min, and Tal Linzen. 2020b. 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.

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.

Geoffrey Nunberg. 1990. *The linguistics of punctuation*. 18. Center for the Study of Language (CSLI).

Geon Woo Park, Junghwa Lee, Meiying Ren, Allison Shindell, and Yeonsoo Lee. 2023. VARCO-MT: NCSOFT's WMT'23 terminology shared task submission. In *Proceedings of the Eighth Conference on Machine Translation*, pages 919–925, Singapore. Association for Computational Linguistics.

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.

Sameer Pradhan, Wayne Ward, Kadri Hacioglu, James Martin, and Daniel Jurafsky. 2005. Semantic role labeling using different syntactic views. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 581–588, Ann Arbor, Michigan. 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.

Vasin Punyakanok, Dan Roth, and Wen-tau Yih. 2008. The importance of syntactic parsing and inference in semantic role labeling. *Computational Linguistics*, 34(2):257–287.

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.

Brandon C Roy, Michael C Frank, Philip DeCamp, Matthew Miller, and Deb Roy. 2015. Predicting the birth of a spoken word. *Proceedings of the National Academy of Sciences*, 112(41):12663–12668.

Valentin I. Spitkovsky, Hiyan Alshawi, and Daniel Jurafsky. 2011. Punctuation: Making a point in unsupervised dependency parsing. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, pages 19–28, Portland, Oregon, USA. Association for Computational Linguistics.

Valentin I. Spitkovsky, Daniel Jurafsky, and Hiyan Alshawi. 2010. Profiting from mark-up: Hyper-text annotations for guided parsing. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1278–1287, Uppsala, Sweden. Association for Computational Linguistics.

Jon Sprouse, Carson T. Schütze, and Diogo Almeida. 2013. A comparison of informal and formal acceptability judgments using a random sample from linguistic inquiry 2001–2010. *Lingua*, 134:219–248.

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.

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.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. *arXiv preprint 1905.00537*.

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, Brussels, Belgium. 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.

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.

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.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2020. Xlnet: Generalized autoregressive pretraining for language understanding.

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.

Qihuang Zhong, Liang Ding, Juhua Liu, Bo Du, and Dacheng Tao. 2023. Can chatgpt understand too? a comparative study on chatgpt and fine-tuned bert.

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.