Acquiring Bidirectionality via Large and Small Language Models

Takumi Goto[1,2], **Hiroyoshi Nagao**[1], **Yuta Koreeda**[1],
[1]Research & Development Group, Hitachi, Ltd., Tokyo, Japan,
[2]NARA Institute of Science and Technology, Nara, Japan,
Correspondence: yuta.koreeda.pb@hitachi.com

Abstract

Using token representation from bidirectional language models (LMs) such as BERT is still a widely used approach for token-classification tasks. Even though there exist much larger unidirectional LMs such as Llama-2, they are rarely used to replace the token representation of bidirectional LMs. In this work, we hypothesize that their lack of bidirectionality is what is keeping unidirectional LMs behind. To that end, we propose to newly train a small backward LM and concatenate its representations to those of an existing LM for downstream tasks. Through experiments in token-classification tasks, we demonstrate that introducing a backward model can improve the benchmark performance by more than 10 points. Furthermore, we show that the proposed method is especially effective for rare domains and in few-shot learning settings.

1 Introduction

In recent years, pretrained large unidirectional language models (UniLMs), such as Llama-2 (Touvron et al., 2023) and OpenAI GPT (OpenAI, 2024), have become widely used. Large UniLMs have demonstrated that various tasks can be solved by means of generation. On the other hand, bidirectional language models (BiLMs), most well-known by BERT (Devlin et al., 2019), equipped with a classification layer are still widely used in many NLP tasks. In particular, BiLMs are still dominant for token-level classification tasks. For example, as of 2024, top three models in two popular token-level classification tasks, CoNLL 2003 named entity recognition (NER) (Tjong Kim Sang and De Meulder, 2003) and DocRED relationship extraction (Yao et al., 2019), are all based on BiLMs[1].

[1]From Papers With Code, as of May, 2024. Top three models are (Wang et al., 2021; Yamada et al., 2020; Zhou and Chen, 2021) for CoNLL 2003 and (Ma et al., 2023; Tan et al., 2022; Xu et al., 2021) for DocRED.

The reason why the application of large UniLMs in token-level classification tasks has not progressed can be attributed to their lack of bidirectionality. In a UniLM, the representation of a token is computed solely based on the preceding context, as we elaborate in Section 2.1. To overcome this problem, BehnamGhader et al. (2024) introduced LLM2Vec, where UniLMs are fine-tuned with masked token prediction after removing their causal attention masks. This allows the model to attend to both the beginning and the end of a sentence thus acquiring bidirectionality. This allows utilizing existing UniLMs not only for generation tasks but also for highly accurate solutions to token-level classification tasks.

LLM2Vec, however, has a downside that it requires training for each UniLM, which is costly if we are to try out various UniLMs to find a good fit for a downstream task. Given the already large and rapidly evolving zoo of UniLMs, it would be beneficial if there is a one-for-many solution for equipping bidirectionality to UniLMs.

In this work, we propose a new way to acquire bidirectionality without tuning large UniLMs themselves. Specifically, we newly train a small UniLM for generating text from the end (referred to the "backward LM") and concatenate its token representations to those of the pretrained UniLM (referred to the "forward LM") to obtain pseudo bidirectionality. After that, we train only the classification layer for the downstream tasks as a drop-in replacement of BiLMs. The backward LM is independent on which forward LM it is used with, thus it can be combined with various size of UniLMs even if it ended up in a heterogeneous configuration.

In the experiments, we focus on three kinds of token-classification tasks, i.e., chunking, part-of-speech (POS) tagging and NER, and compare the performances of UniLMs with and without backward LM. We observe that adding backward LMs consistently improves the performance by up to

Proceedings of the 31st International Conference on Computational Linguistics, pages 1711–1717
January 19–24, 2025.

more than 10 points in CoNLL2003-NER. Additionally, we demonstrate that the proposed method consistently improves performance in few-shot settings or when targeting rare domains.

The contributions of this study are as follows:

1. We empirically show that unidirectionality is a problem when adopting UniLMs to token-level classification tasks.
2. We proposed a novel method to newly train a small-scale backward LM and concatenate its representations to those of existing LM to achieve pseudo bidirectionality in UniLMs.
3. We open-sourced backward LM and its training code to foster future research[2].

2 Proposed Method

2.1 Prerequisite

In this section, we review two types of LMs: UniLMs and BiLMs. Given an input sequence $\boldsymbol{x} = (x_1, x_2, \ldots, x_N)$ with N tokens, the difference between the two models lies in how they compute representations for each word x_i $(1 \leq i \leq N)$.

In BiLMs, the representation $\mathbf{h}_i^{bi}$ is computed based on the context from both the beginning and the end of the text:

$$\mathbf{h}_i^{bi} = BiLM_\phi(x_i|\boldsymbol{x}_{<i}, \boldsymbol{x}_{>i}), \quad (1)$$

where $\boldsymbol{x}_{<i} = (x_1, x_2, \ldots, x_{i-1})$ and $\boldsymbol{x}_{>i} = (x_{i+1}, x_{i+2}, \ldots, x_N)$. To solve token-classification tasks, we can input $\mathbf{h}_i^{bi}$ to the newly added classification layers.

On the other hand, a forward LM computes the representation $\overrightarrow{\mathbf{h}}_i$ solely based on earlier context:

$$\overrightarrow{\mathbf{h}}_i = \overrightarrow{UniLM_\theta}(x_i|\boldsymbol{x}_{<i}). \quad (2)$$

As can be seen from equation (2), UniLMs need to compute the representation for the i-th word without using the subsequent context $\boldsymbol{x}_{>i}$.

2.2 Utilization of the Bidirectional Language Model

The proposed method leverages the concatenated representations of both the forward LM $\overrightarrow{UniLM}(\cdot)$ and the backward LM $\overleftarrow{UniLM}(\cdot)$ for the downstream task. In contrast to the forward LM, a backward LM computes $\overleftarrow{\mathbf{h}}_i$ given the context from the end:

$$\overleftarrow{\mathbf{h}}_i = \overleftarrow{UniLM_{\theta'}}(x_i|\boldsymbol{x}_{>i}). \quad (3)$$

The final representation for the i-th token considers both the forward and backward contexts by concatenating $\overrightarrow{\mathbf{h}}_i$ and $\overleftarrow{\mathbf{h}}_i$, denoted as $\mathbf{h}_i = \mathrm{Concat}[\overrightarrow{\mathbf{h}}_i, \overleftarrow{\mathbf{h}}_i]$. Therefore, the dimensions of $\mathbf{h}_i$ is the sum of the hidden vector dimensions of the forward and backward LMs.

To compute the concatenated representations as described above, it is necessary to share the vocabulary between the forward and backward LMs. Nevertheless, it is possible to use arbitrary architectures and parameter sizes for both models. For instance, we can employ a heterogeneous configuration, such as $|\theta| \gg |\theta'|$. This means that we can utilize the existing assets of $|\theta|$ with just a small compute of training $|\theta'|$. In this study, a part of experiments is conducted with such heterogeneous configuration.

3 Experiments

We verify whether UniLMs can acquire bidirectionality by adding backward UniLMs through evaluation on four token-classification tasks. We train a backward LM (124M parameters) and apply it to GPT-2 (base, 124M) (Radford et al., 2019) and Llama2-7b (Touvron et al., 2023), to verify whether the proposed method can be applied to UniLMs of different sizes.

3.1 Training Backward LM

We train a backward LM for each of Llama2 and GPT-2, as the backward LM should have the same vocabulary as the forward LM. The architecture follows that of GPT-2 (base), but we resize the input dimension of the embedding layer to match the vocabulary size of the forward LM. We initialize the models with random parameters and train it on BookCorpus (Zhu et al., 2015) and Wikitext (Merity et al., 2017) (`wikitext-103-raw-v1`) datasets, with next token prediction objective. During the preprocessing step, we concatenate the training data from both datasets and shuffle them on a document level[3]. Next, we perform subword tokenization with the forward LM's tokenizer. We extract training data by cutting it into segments of 1,024 tokens, starting the beginning of the dataset, and then reversing them. For training, we set the batch size to 512 and the learning rate to 2e-5 with a cosine scheduler.

[2] https://github.com/hitachi-nlp/backward-llm

[3] For Wikitext, we removed empty lines and strings corresponding to headings beforehand.

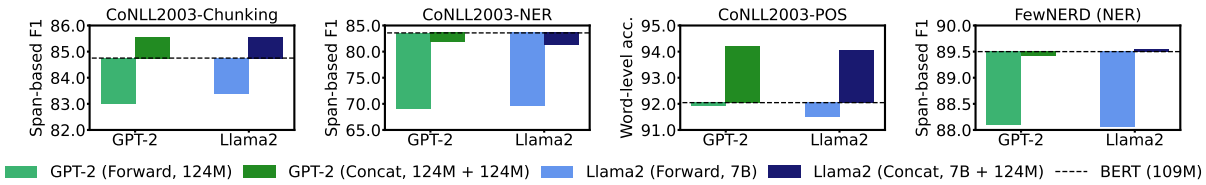

Figure 1: The performance of UniLMs ("Forward") and the proposed concatenated LMs ("Concat") against a BiLM (BERT)

3.2 Downstream Tasks

To evaluate the effectiveness of the proposed method in a token-level classification task, we employ chunking, POS tagging, NER from CoNLL-2003 dataset (Tjong Kim Sang and De Meulder, 2003). In addition, we utilize Few-NERD dataset (supervised setting) (Ding et al., 2021) to verify the NER performance on rare domains. We hypothesize that larger forward LMs are effective in this more challenging setting because of their extensive knowledge.

In the evaluation, we use span-based F_1 score for chunking and NER, and word-level accuracy for POS.

During downstream task training, we input the representations $\mathbf{h}_i$ from each model into the classification layer and optimize the classification layer while keeping using cross-entropy loss. The classification layer consists of two linear layers. Let d be the dimensionality of $\mathbf{h}_i$ and c be the number of classes. We use $\mathbf{W}_1 \in \mathbb{R}^{d \times d}$ and $\mathbf{W}_2 \in \mathbb{R}^{c \times d}$ to estimate the distribution $\mathbf{p} = \mathrm{softmax}(\mathbf{W}_2 \mathrm{ReLU}(\mathbf{W}_1 \mathbf{h}_i)) \in \mathbb{R}^c$.

We set the batch size to 32 and employ AdamW (Loshchilov and Hutter, 2019). We linearly decay the learning rate to zero from 1e-3. We only train the classification layer while keeping the other layers fixed.

We report the average scores of three different seeds using the checkpoint that has maximum F_1 scores on the validation set. We also train a model using BERT (`bert-base-uncased`) with the same setting to compare the proposed method to a BiLM.

3.3 Few-shot Setting

One of the potential benefits of large LMs is their ability to make generalized predictions even with a small number of training examples, leveraging the knowledge embedded in their parameters. We also analyze NER performance on CoNLL-2003 with K-shot setting to examine the impact of limited training examples.

In our K-shot setting, the training data consists of $4K$ samples since we draw K samples from each entity type: `PER`, `LOC`, `ORG`, and `MISC`. Note that we only extract instances from the training data that contain a single specific named entity type. During training, we set the batch size to 4 and randomly sampled the following hyperparameters; a learning rate from $\{9e-3, 8e-3, \ldots, 2e-4, 1e-4\}$, a seed from $\{10, 11, \ldots, 19\}$, a dropout probability from $\{0, 0.1, 0.2, 0.3\}$. We determined top-3 hyperparameter settings in terms of F_1 score on the CoNLL-2003 validation set, and report the average F_1 on the test set of those settings.

3.4 Results

3.4.1 Full Dataset Setting

We show the results in Figure 1. The results show the difference of the performance between BERT and each of the settings. We can see the effectiveness of the proposed method by comparing "Forward" and "Concat" settings, which indicates the proposed method improves the performance on all tasks. In particular, F_1 scores on CoNLL2003-NER have been improved by more than 10 points for both models.

These results indicate that considering backward context improves token-classification performance and that the proposed method can provide the backward context to UniLMs. Moreover, the comparison of BERT and the proposed method indicates that our approach has ability to bring the performance of existing UniLMs comparable or better to BERT performance. This implies that UniLMs can acquire bidirectionality post hoc.

In the case of Few-NERD, Llama-2 outperformed GPT-2 with the proposed method. It can be inferred that larger forward LMs are more effective when targeting a rarer domain.

3.4.2 Few-shot Setting

The experimental results using BERT (`bert-base-cased`) and GPT-2 (base and xl) are shown in Figure 2. The x-axis represents

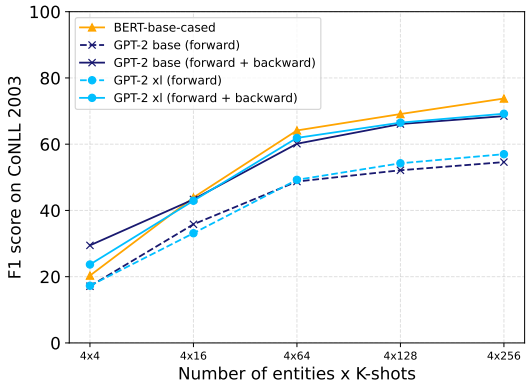

Figure 2: Few-shot setting results on CoNLL-2003 test set. In the x-axis, the number of training examples is represented by multiplication of the number of entities ($= 4$) and K.

the number of training examples $4K$, and the y-axis represents the F_1 score on CoNLL2003 test set. The dashed lines indicate only forward LM while the solid lines represent the bidirectional and the proposal setting. As observed from Figure 2, the proposed method consistently outperforms the forward LM-only setting. Particularly, when the training data is limited (less than 16 shots for each entity) the proposed method is more effective than BERT. This has a significant value in practice as there is generally a *valley* of performance between few- to many-shots settings; zero- to few-shots settings are more effectively addressed by in-context learning and many-shots settings are covered by BiLMs, but neither covers the middle. Addressing this *valley* is important, because annotating dozens of data for each label might be justified but annotating hundreds sounds overwhelming to many practitioners.

3.5 Case Study

We conduct a case study to understand how adding backward context improves the token-level classification performance. As shown in Table 1, we found that the proposed method is particularly effective when there is an entity at the beginning of a sentence. Specifically, the NER results for the sentence "Jones Medical completes acquisition ." are [B-PER, I-ORG, –, –, –] for the forward UniLM setting only, and [B-ORG, I-ORG, –, –, –] for the proposed method and the reference. The forward UniLM could not capture any context because the entity appears at the beginning of the sentence. In contrast, the proposed method was able to predict

Input	Jones	Medical	completes	acquisition	.
UniLM	B-PER	I-ORG	-	-	-
Proposal	B-ORG	I-ORG	-	-	-
Reference	B-ORG	I-ORG	-	-	-

Table 1: An example of NER with GPT-2 (base) in a case that an entity at the beginning of the sentence.

the entity using the context from the end. We also found that the proposed method could accurately estimate the leading entity in phrases where entities are conjoined by "and."[4] These results suggest that the UniLM representations are not suitable for token-level classification tasks, even when the LM is of large size, but the proposed method is able to overcome this weakness by the simple idea.

4 Related Work

A traditional approach to combining UniLMs, such as BiLSTM (Schuster and Paliwal, 1997) and ELMo (Peters et al., 2018), is similar to the proposal method. Our study revisits this idea in the era of large LMs, and systematically shows its effectiveness through compute-demanding experiments. Our method is simple, but it in returns shows that backward context matters even in the era of large LM and that the traditional approach of acquiring bidirectionality is still valid. In the era of large language models, *meet-in-the-middle* approach (Nguyen et al., 2023; Li et al., 2023b) consider bidirectionality during generation by incorporating backward generation probability. While there is some relevancy in the concept, these studies work with generation whereas we aim to improve the quality of token-level representations.

For the study to improve the quality of the representation, Li et al. (2023a), BehnamGhader et al. (2024) and Dukić and Snajder (2024) fine-tuned UniLMs after removing the causal attention mask to incorporate the context from the end of the sentence. Although these methods require fine-tuning for each LM, the proposed method can reuse the backward LM as long as the vocabulary and tokenizer are the same. Moreover, another benefit is that our approach can be applied to a black-box model, i.e., when the parameters of a model are not accessible but its final representations can be obtained via API.

[4]The actual example can be found in Appendix A.

5 Conclusion

In this study, we proposed to concatenate the representations of forward and backward LM, to overcome the lack of bidirectionality problem of UniLMs. From the results in token-classification tasks, we could confirm the effectiveness of the proposed method. The proposed method provides more use cases to UniLM, not only for a generation model but also as an encoder model.

Acknowledgments

We would like to thank Dr. Masaaki Shimizu for arranging the computational environment. We would like to thank the COLING reviewers and program chairs for their suggestive comments and feedbacks.

Limitations

Limited Scope of Tasks and Conditions In this work, we evaluated our proposed method with two backbone UniLMs under four different token-level classification tasks in English, and we observed consistent performance improvement from the original UniLMs. We focused on token-level classification tasks in this study because we believe they are the tasks where the effects of considering bidirectionality are best demonstrated. Nevertheless, we would like to extend the experiments in the future to investigate varying effects of the proposed method under different conditions. For example, we can apply the proposed method to text classification tasks by pooling token-level representations. Also, we would like to see if significantly larger UniLMs can yield better results, as we did not observe any performance gain within the relatively small scope of model sizes (124M to 7B parameters) that we experimented with in this paper.

Training Strictly Comparable Models from Scratch We compared BERT and GPT-2 because(a) they were release roughly the same time and together represent the state-of-the-arts of unidirectional and bidirectional LM of the time, and (b) they are mostly comparable in terms of magnitude of parameter sizes and training data. In reality, there *might* be auxiliary differences because the dataset for training or detail training settings, e.g., the number of epochs, to train GPT-2 are not disclosed (Radford et al., 2019). Though our main finding that unidirectionality is a problem in UniLMs holds from the GPT-2 results alone, strict comparison can be performed by training BERT and GPT-2-like models from scratch with the same trainig data, which we leave for the future work.

Additionally, we can also consider using BiLMs instead of a backward language model, as a provider of backward context.

Comparison against Other Methods We did not quantitatively compare the proposed method against LLM2vec (BehnamGhader et al., 2024) in the experiments. It was due to the fact their reported results employed non-standard word-level scores for NER instead of more standard span-level scores that we employed and hence direct comparison was not possible. We note that our method has unique benefits that LLM2vec does not have: (a) our backward model can be reused for different backbone UniLMs as long as the tokenizer is the same, and (b) we only need the output representations of UniLMs and do not require access to the internal representations nor gradients of the UniLMs. Furthermore, our work has contribution to the community by validating through a different approach that bidirectionality is the key ingredient of the success of BiLMs.

Computation Cost The proposed method requires a backward LM, thus the inference computation cost is higher than when using only a forward LM. Nevertheless, this is generally not a serious problem because the backward language model works even with modest sized models such as GPT-2. In practice, the proposed method can be executed by reusing the large UniLMs deployed for other purpose, e.g., generation, so there should be cases where it is sufficient to deploy only the backward UniLM. In this case, unless the UniLM's GPU utilization is 100%, we can run the proposed method with only the cost of running backward UniLM.

Ethical Consideration

Our research involves training moderately large LMs and its carbon footprint can have negative impact to the environment. That being said, we train reusable LMs that allow us to utilize existing assets (large UniLMs) to where they were previously weak at. This can make the whole ecosystem of LLMs more efficient and might be able to reduce carbon footprint in the long run.

From the social justice and language preservation perspective, the downside of the proposed

method is that it mainly benefits resource-rich languages with existing assets of large UniLMs. Nevertheless, more and more of recent UniLMs support multilinguality (Team, 2024; Llama Team, AI@Meta, 2024). We would like to explore our proposed method in low-resource, cross-lingual settings in the future.

References

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. LLM2vec: Large language models are secretly powerful text encoders. In *First Conference on Language Modeling*.

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*.

Ning Ding, Guangwei Xu, Yulin Chen, Xiaobin Wang, Xu Han, Pengjun Xie, Haitao Zheng, and Zhiyuan Liu. 2021. Few-NERD: A few-shot named entity recognition dataset. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*.

David Dukić and Jan Snajder. 2024. Looking right is sometimes right: Investigating the capabilities of decoder-only LLMs for sequence labeling. In *Findings of the Association for Computational Linguistics ACL 2024*.

Zongxi Li, Xianming Li, Yuzhang Liu, Haoran Xie, Jing Li, Fu lee Wang, Qing Li, and Xiaoqin Zhong. 2023a. Label supervised LLaMA finetuning. *Preprint*, arXiv:2310.01208.

Zuchao Li, Shitou Zhang, Hai Zhao, Yifei Yang, and Dongjie Yang. 2023b. BatGPT: A bidirectional autoregessive talker from generative pre-trained transformer. *arXiv preprint arXiv:2307.00360*. Accessed v2.

Llama Team, AI@Meta. 2024. The Llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Youmi Ma, An Wang, and Naoaki Okazaki. 2023. DREEAM: Guiding attention with evidence for improving document-level relation extraction. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models. In *International Conference on Learning Representations*.

Anh Nguyen, Nikos Karampatziakis, and Weizhu Chen. 2023. Meet in the middle: A new pre-training paradigm. In *Advances in Neural Information Processing Systems*, volume 36, pages 5079–5091. Curran Associates, Inc.

OpenAI. 2024. GPT-4 technical report. *Preprint*, arXiv:2303.08774v6.

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*.

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

Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681.

Qingyu Tan, Ruidan He, Lidong Bing, and Hwee Tou Ng. 2022. Document-level relation extraction with adaptive focal loss and knowledge distillation. In *Findings of the Association for Computational Linguistics: ACL 2022*.

Gemma Team. 2024. Gemma 2: Improving open language models at a practical size. *Preprint*, arXiv:2408.00118.

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*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov,

Input	Note	-	Lotte	and	Hyundai	,	Haitai	and	Samsung	played	two	games	.
Forward-LM only	-	-	B-PER	-	B-ORG	-	-		B-ORG	-	-	-	-
Proposed	-	-	B-ORG	-	B-ORG	-	B-MISC		B-ORG	-	-	-	-
Reference	-	-	B-ORG	-	B-ORG	-	B-ORG		B-ORG	-	-	-	-

Table 2: An example of GPT-2 (base) in phrases where entities are conjoined by "and." For the prediction corresponding to `Lotte`, the forward-LM struggles to infer the entity type correctly, but proposal can estimate correctly. This can be explained by the difference between considering only the context from the beginning: "`Note - Lotte`" or entire context, cause the improvement.

Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

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*.

Benfeng Xu, Quan Wang, Yajuan Lyu, Yong Zhu, and Zhendong Mao. 2021. Entity structure within and throughout: Modeling mention dependencies for document-level relation extraction. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. LUKE: Deep contextualized entity representations with entity-aware self-attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

Yuan Yao, Deming Ye, Peng Li, Xu Han, Yankai Lin, Zhenghao Liu, Zhiyuan Liu, Lixin Huang, Jie Zhou, and Maosong Sun. 2019. DocRED: A large-scale document-level relation extraction dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Wenxuan Zhou and Muhao Chen. 2021. Learning from noisy labels for entity-centric information extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *The IEEE International Conference on Computer Vision*.

Appendices

A Another Example in Case Study

Table 2 shows an example where entities are concatenated with "and." The forward-LM only setting struggles with identifying the type of first entity, for "Lotte." In contrast, the proposed method correctly identified the entity type of it by using other organization names such as "Hyundai."