GenerativeRE: Incorporating a Novel Copy Mechanism and Pretrained Model for Joint Entity and Relation Extraction

Jiarun Cao, Sophia Ananiadou*
National Centre for Text Mining
University of Manchester
the United Kingdom
{jiarun.cao,sophia.ananiadou}@manchester.ac.uk

Abstract

Previous neural seq2seq models have shown the effectiveness for jointly extracting relation triplets. However, most of these models suffer from incompletion and disorder problems when they extract multi-token entities from input sentences. To tackle these problems, we propose a generative, multi-task learning framework, named GenerativeRE. We firstly propose a special entity labelling method on both input and output sequences. During the training stage, GenerativeRE fine-tunes the pretrained generative model and learns the special entity labels simultaneously. During the inference stage, we propose a novel copy mechanism equipped with three mask strategies, to generate the most probable tokens by diminishing the scope of the model decoder. Experimental results show that our model achieves 4.6% and 0.9% F1 score improvements over the current state-of-the-art methods in the NYT24 and NYT29 benchmark datasets respectively.

1 Introduction

The seq2seq based models have attracted much attention in recent years (Zeng et al., 2018; Nayak and Tou Ng, 2019; Chen et al., 2019; Zeng et al., 2019) to jointly extract entities and relations. These models can transform the joint entity and relation extraction task into a sequence generation task, in which the relation triplets are generated in a sequence manner.

Early attempts (Gupta et al., 2016; Adel and Schütze, 2017; Zheng et al., 2017; Paterson and Dancík, 1994; Devlin et al., 2019; Takanobu et al., 2018) are limited due to the out of vocabulary and overlapping issues (Zeng et al., 2018; Riedel et al., 2010; Gardent et al., 2017). To overcome these problems, a copying (Gu et al., 2016) or pointing mechanism (Vinyals et al., 2015) has been used. However, two key problems remain: firstly, the model only considers single tokens when copying from input sentences and generates tokens in a token-by-token manner, thus, losing tokens while copying multi-token entities (Zeng et al., 2018; Nayak and Tou Ng, 2019). This results in incompletion errors(See appendix A.5). Secondly, some previous attempts also (Chen et al., 2019; Zeng et al., 2019) suffer from word disorder whilst extracting multi-token relation triplets from a long input sentence as shown in Appendix A.5. These issues worsen when more fine-grained tokenization methods are applied, such as WorkPiece (Wu et al., 2016), which splits the whole sequence into subwords and logically deteriorates these issues (Dong et al., 2019). According to our experimental results on NYT24 and NYT29 , 80.3% examples contain multi-token triplets in NYT24 dataset (Zeng et al., 2018), and 80.9% in NYT29 datasets (Takanobu et al., 2018). Thus, although word incompletion and disorder problem are very common in our task, it has not been fully explored.

To address the issues aforementioned, we propose a multi-task learning framework, called GenerativeRE, which incorporates a novel copy mechanism with a generative pretrained model (Dong et al., 2019; Su, 2021)for joint entity and relation extraction. Specifically, we first design a BIO labelling method by calculating the longest common subsequence between input sentence and output triplets sentence, which enables the BIO labels to locate the boundaries for the complete multi-token entities.

During the training stage, we adopt a generative pretrained model as our backbone model network, and propose a multi-task learning framework to learn the masked tokens and their corresponding BIO labels simultaneously. During the inference stage, at each time step, we first predict the BIO label of each token. BIO labels are aligned with three masking strategies on the probability distribution over the entire vocabulary list, and the model

[*] Corresponding author

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 2119–2126
November 7–11, 2021. ©2021 Association for Computational Linguistics

is guided by different mask strategies to extract the correct token in a correct order. Experimental results show the effectiveness of our model in alleviating the incompletion issue and disorder issue whilst copying multi-token entities.

2 Approach

2.1 BIO Label Construction

In our task, the model input is a sequence of tokens and the output is a set of relation triplets. Following (Nayak and Tou Ng, 2019), we represent the output as a sentence pattern as `entity ; entity ; relation | entity ; entity ; relation` as presented in Figure 1, where `;` is used to separate entities and `|` is used to separate triplets. Multiple relation tuples with overlapping entities and nested entities (Ju et al., 2018) can be represented in a simple way using these special tokens `;` and `|`.

The input and output sentences are then tokenized into subwords by WordPiece (Wu et al., 2016). Then, we adopt the longest common subsequence (LCS) algorithm (Paterson and Dancík, 1994) to generate the BIO labels for each subword. LCS collects the entire longest common subsequences between input and output sentence. For example, as shown in Figure 1, the longest common subsequences are "Evan Z ##ip ##ory ##n", "Massachusetts Institute of Technology", "K ##ya ##w K ##ya ##w Na ##ing" and "New York".

Last, we assign a BIO label on each token in the input and triplets sequence to locate the boundary index of multi-token entities, concretely, "B" is the beginning position of the entity, "I" denotes the middle(inside) position of the entity, and "O" denotes not belonging to any entities.

2.2 Input Representation

We treat triple extraction as a sequence generation task as shown in Figure 1.The input representation follows that of BERT (Devlin et al., 2019). In our task, `[EOS]` token is not only used as an end-of-sequence symbol, but it is also used as a special token to terminate the triplet generation. We denote the input sentence as S_1 and the triplet sentence as S_2. Thus, the model input $\{x_i\}_{i=1}^{|x|}$ is a concatenation of each part into `[SOS] input sentence [EOS] triplet sentence [SOS]` . By conducting the BIO construction approach in section 2.1, each subword in the model input is also assigned to a corresponding BIO label y_i^{BIO} as shown in Figure 1.

We utilize a multi-layer Transformer as the backbone network to encode contextual features which are constituted by stacked self attention layers. Given the input vectors $\{x_i\}_{i=1}^{|x|}$, we first pack them into $H^0 = [x_1, ..., x_{|x|}]$, then we use a L-layer Transformer to encode the input into contextual representation:

$$H^l = \mathrm{Transformer}_l(H^{l-1}) \tag{1}$$

where $l \in [1, L]$. In each Transformer block, multiple self-attention heads are applied to aggregate the output vectors of the previous layer. We compute the output of a self-attention head A_l in the l-th Transformer layer as follows:

$$Q_l = H^{l-1}W_l^Q, K_l = H^{l-1}W_l^K \tag{2}$$

$$M_{ij} = \begin{cases} 0 & \text{allow to attend} \\ -\infty & \text{prevent from attending} \end{cases} \tag{3}$$

$$A_l = \mathrm{softmax}(\frac{Q_l K^\top}{\sqrt{d_k}} + M)(H^{l-1}V_l) \tag{4}$$

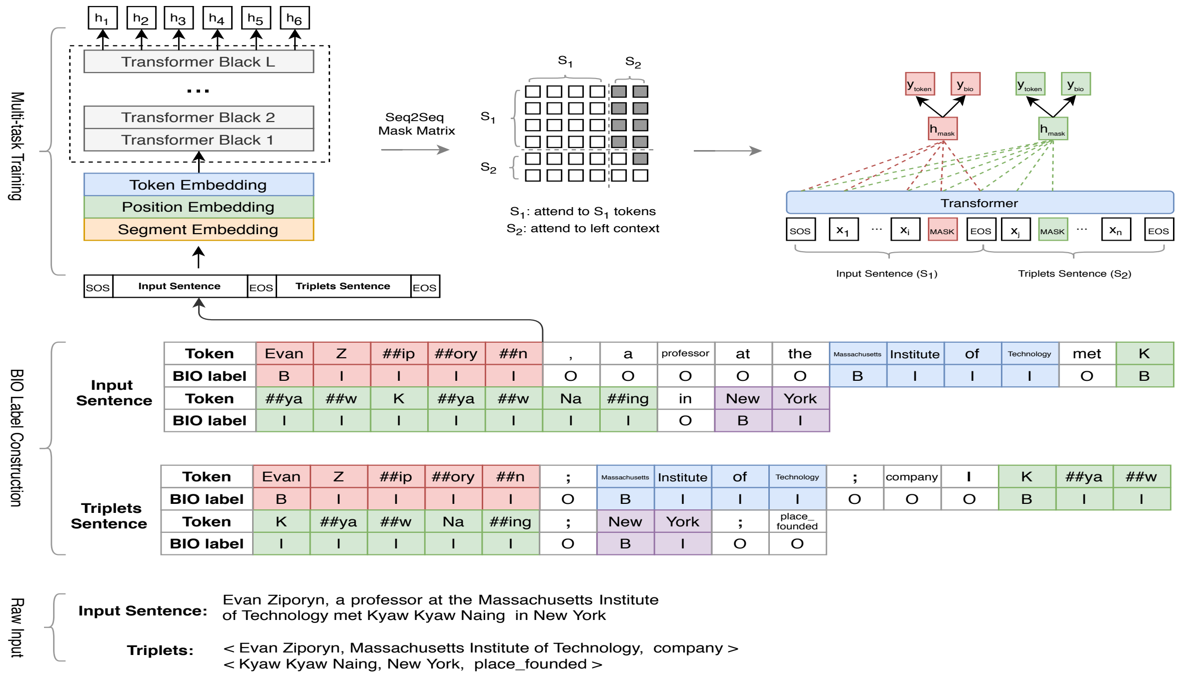
where Q_l, K_l, $V_l^v \in \mathbb{R}^{d_h \times d_k}$ denotes parameter matrices of queries, keys and values for the projection of the previous layer output H^{l-1} respectively, and the mask matrix $M \in \mathbb{R}^{|x| \times |x|}$ determines the context that can be attended by the token. The setting of seq2seq mask matrix follows Unilm (Dong et al., 2019).

2.3 Training

In the training stage, different from Unilm (Dong et al., 2019), we randomly mask not only the tokens as `[MASK]` with a certain probability, but also their corresponding BIO labels from both segments, and then compel the model to learn to recover the masked tokens and BIO labels jointly. The training objective is to maximize the likelihood of masked tokens and their BIO labels given the context. Formally, given the masked tokens x_i, we obtain its contextualized representations h_i, then we add two separate fully-connected layers to obtain their token distribution and BIO label distribution respectively:

$$h_i = \mathrm{Transformer}(x_i) \tag{5}$$

$$y_i^{\mathrm{token}} = \mathrm{softmax}(W_1 h_i + b_1) \tag{6}$$



Input Sentence	Token	Evan	Z	##ip	##ory	##n	,	a	professor	at	the	Massachusetts	Institute	of	Technology	met	K
	BIO label	B	I	I	I	I	O	O	O	O	O	B	I	I	I	O	B
	Token	##ya	##w	K	##ya	##w	Na	##ing	in	New	York						
	BIO label	I	I	I	I	I	I	I	O	B	I						

Triplets Sentence	Token	Evan	Z	##ip	##ory	##n	;	Massachusetts	Institute	of	Technology	;	company	I	K	##ya	##w
	BIO label	B	I	I	I	I	O	B	I	I	I	O	O	O	B	I	I
	Token	K	##ya	##w	Na	##ing	;	New	York	;	place_ founded						
	BIO label	I	I	I	I	I	O	B	I	O	O						

Input Sentence: Evan Ziporyn, a professor at the Massachusetts Institute of Technology met Kyaw Kyaw Naing in New York

Triplets: < Evan Ziporyn, Massachusetts Institute of Technology, company >
< Kyaw Kyaw Naing, New York, place_founded >

Figure 1: Overall model structure during training. The model goes in a bottom-to way: the raw input triplets are firstly concatenated with special separator tokens and then tokenized by WordPiece (Wu et al., 2016). Then, the model constructs the BIO labels for both input and triplet sentence, and feeds them into the Transformers encoder for multi-task learning. The constructed BIO labels are used as the ground truth labels for training.

$$\hat{y}_i^{\text{BIO}} = \text{softmax}(W_2 h_i + b_2) \tag{7}$$

where $W_1 \in \mathbb{R}^{|V| \times d_h}$ is the weight matrix and bias vector $b_1 \in \mathbb{R}^{|V|}$, $|V|$ denotes the number of vocabularies, while $W_2 \in \mathbb{R}^{|L| \times d_h}$ and bias vector $b_2 \in R^{|L|}$, here $|L|$ denotes the number of BIO label types which equals to 3 in our case.

Then we define the cross entropy loss function as the weighted summation of token loss and BIO loss:

$$\mathcal{L} = \sum \sum_{i=0}^{m} \alpha(x_i \log(\hat{y}_i^{\text{token}})) + (1-\alpha)(y_i^{\text{BIO}} \log(\hat{y}_i^{\text{BIO}})) \tag{8}$$

where α is a weight hyper-parameter to balance different objectives.

2.4 Inference

During inference, instead of generating tokens in a straightforward manner, at each time step, our model decoder firstly predicts a BIO label . Then, we conduct one of three mask strategies based on the predicted BIO label to narrow down the scope of token distribution, in which it enforces the model to generate the multi-token entities completely and in a correct order. The detailed mask strategies are as follows:

If BIO = O, it indicates the token to be predicted does not belong to any entities from input sentence, so we retain the original token distribution.

If BIO = B, it indicates the token to be predicted belongs to either a single-token entity or the first token of a multi-token entity from the input sentence, so we only retain the distributions of tokens from the input sentence, otherwise being masked as 0.

If BIO = I, it indicates the token to be predicted belongs to a multi-token entity from the input sentence. Therefore, we look back to the previous predicted tokens and collect all these tokens until we find the nearest token of which its BIO = B. Then, we use these collected tokens as a sequence to match the same sequence from the input sentence. If it can be matched successfully, we will pick all the tokens which are next to this sequence in the input sentence as our candidates, and mask all the distributions of tokens except candidates. If not, we will retain the original token distribution as BIO = O does. A example is shown in Appendix A.4.

3 Experiments

Datasets and experiment settings In this work, we use NYT24 (Zeng et al., 2018) and

NYT29 (Takanobu et al., 2018) as our experimental datasets. Further information can be found in Appendix A.2. We utilize *Unilm-base-cased*[1] as our pretrained model. The model structure of *Unilm* follows that of BERT-Large (Devlin et al., 2019). The experimental parameters are aligned with those in baselines, full details are listed in Appendix A.3.

3.1 Experimental Results

Comparison to previous baselines. Since we extract entity and relation extraction in a seq2seq framework, we compare the performance of GenerativeRE with the state-of-the-art generative models(see Appendix A.1). Table 1 shows the result of different models. The proposed model GenerativeRE substantially outperforms the state-of-the-art models by 4.6% and 0.9 % F1 score in NYT24 and NYT29 respectively. These results verify the effectiveness of our proposed model. Moreover, our GenerativeRE also achieves the best score in terms of 86.3% Recall in NYT24 and 63.6% Recall in NYT29, since GenerativeRE returns the relevant multi-token entities most.

Ablation study. We examine the contributions of our primary model components. As shown in Table 2, `LSTM` represents Bi-LSTM are used as our model encoder and decoder, which is as same model structure as WordDec in baseline (Nayak and Tou Ng, 2019). `Pretrained` represents we use generative pretrained model and generate the tokens in the same way as Unilm (Dong et al., 2019). By comparing the performance between `LSTM` and `Pretrained`, we observbe that the model gains improvement of 7.5% and 5.2% F1 score in NYT24 and NYT29 respectively.+ `Copy` adds the copy mechanism to GenerativeRE, which includes all the steps in Section 2. For both `LSTM` and `Pretrained`, it can be seen that they all gain better result by adding our copy mechanism +`Copy`.

	NYT24			NYT29		
Model	Prec	Rec	F1	Prec	Rec	F1
CopyRE	0.610	0.566	0.587	0.569	0.452	0.504
CopyMTL	0.757	0.687	0.720	0.701	0.623	0.660
MrMep	0.779	0.766	0.771	-	-	-
PtDec	0.806	0.773	0.789	0.732	0.624	0.673
WordDec	**0.881**	0.761	0.817	**0.777**	0.608	0.682
GenerativeRE	0.880	**0.847**	**0.863**	0.756	**0.636**	**0.691**

Table 1: Results of different baseline models in NYT datasets

	NYT24			NYT29		
Model	Prec	Rec	F1	Prec	Rec	F1
LSTM	0.762	0.647	0.700	0.665	0.551	0.603
+ Copy	0.877	0.777	0.824	**0.757**	0.612	0.677
Pretrained	0.890	0.687	0.775	0.739	0.588	0.655
+ Copy	0.880	**0.847**	**0.863**	0.756	**0.636**	**0.691**

Table 2: Ablation study of GenerativeRE with different settings

Effectiveness Analysis. The proposed copy mechanism boosts the performance of joint entity and relation extraction by addressing the incompletion and disorder errors. To prove the effectiveness, we test the number of both incompletion and disorder errors, In Table 3, we can observed that the number of incompletion and disorder cases drop to a large extent by adding our copy mechanism to the raw model. Furthermore, since both incompletion and disorder problem occur in multi-token triplets, we conduct an extra experiment that compare GenerativeRE with state-of-the-art baseline models in terms of tackling the triplets that contain multi-token entities, as we can see from Table 4, our GenerativeRE consistently outperforms the baseline models by 5.2%, 6.1%, and 1.4% F1 score in terms of 2-token triplets, 3-token triplets, and more than 3-token triplets, respectively.

	NYT24		**NYT29**	
	Inc.	*Dis.*	*Inc.*	*Dis.*
Raw model	539	179	425	47
+ Copy	135	127	249	37

Table 3: Number of incompletion and disorder errors with different settings.

Models	**2-token**	**3-token**	**3+ tokens**
WordDec	0.765	0.643	0.642
PtDec	0.731	0.674	0.700
GenerativeRE	**0.817**	**0.735**	**0.714**

Table 4: F1 scores of different multi-token triplets in NYT24.

4 Conclusion

In this paper, we propose GenerativeRE which incorporates a novel copy mechanism to extract the entity and relation autoregressively. GenerativeRE achieves state-of-the-art result on two benchmark datasets, whiche improves the model effectiveness.

[1] `https://github.com/microsoft/unilm/tree/master/unilm`

References

Heike Adel and Hinrich Schütze. 2017. Global normalization of convolutional neural networks for joint entity and relation classification. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1723–1729, Copenhagen, Denmark. Association for Computational Linguistics.

Jiayu Chen, Caixia Yuan, Xiaojie Wang, and Ziwei Bai. 2019. MrMep: Joint extraction of multiple relations and multiple entity pairs based on triplet attention. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 593–602, Hong Kong, China. Association for Computational Linguistics.

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.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 13042–13054.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. Creating training corpora for NLG micro-planners. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 179–188, Vancouver, Canada. Association for Computational Linguistics.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1640, Berlin, Germany. Association for Computational Linguistics.

Pankaj Gupta, Hinrich Schütze, and Bernt Andrassy. 2016. Table filling multi-task recurrent neural network for joint entity and relation extraction. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2537–2547, Osaka, Japan. The COLING 2016 Organizing Committee.

Meizhi Ju, Makoto Miwa, and Sophia Ananiadou. 2018. A neural layered model for nested named entity recognition. 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 1446–1459, New Orleans, Louisiana. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Tapas Nayak and Hwee Tou Ng. 2019. Effective Modeling of Encoder-Decoder Architecture for Joint Entity and Relation Extraction. *arXiv e-prints*, page arXiv:1911.09886.

Mike Paterson and Vlado Dancík. 1994. Longest common subsequences. In *Proceedings of the 19th International Symposium on Mathematical Foundations of Computer Science 1994*, MFCS '94, page 127–142, Berlin, Heidelberg. Springer-Verlag.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Machine Learning and Knowledge Discovery in Databases, European Conference, ECML PKDD 2010, Barcelona, Spain, September 20-24, 2010, Proceedings, Part III*, volume 6323 of *Lecture Notes in Computer Science*, pages 148–163. Springer.

Jianlin Su. 2021. Extraction-generation of long text abstract. `https://spaces.ac.cn/archives/8046` Accessed January 01, 2021.

Ryuichi Takanobu, Tianyang Zhang, Jiexi Liu, and Minlie Huang. 2018. A hierarchical framework for relation extraction with reinforcement learning. *CoRR*, abs/1811.03925.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 2692–2700.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

Daojian Zeng, Ranran Haoran Zhang, and Qianying Liu. 2019. CopyMTL: Copy Mechanism for Joint Extraction of Entities and Relations with Multi-Task Learning. *arXiv e-prints*, page arXiv:1911.10438.

Xiangrong Zeng, Daojian Zeng, Shizhu He, Kang Liu, and Jun Zhao. 2018. Extracting relational facts by an end-to-end neural model with copy mechanism. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 506–514, Melbourne, Australia. Association for Computational Linguistics.

Suncong Zheng, Feng Wang, Hongyun Bao, Yuexing Hao, Peng Zhou, and Bo Xu. 2017. Joint extraction of entities and relations based on a novel tagging scheme. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1227–1236, Vancouver, Canada. Association for Computational Linguistics.

A Appendix

A.1 Baselines and Evaluation Metrics

Since we extract entity and relation extraction in a Seq2Seq framework, we compare the performance of GenerativeRE with the following state-of-the-art Seq2Seq models:

CopyRE (Zeng et al., 2018) firstly uses an encoder-decoder framework to jointly extract entities and relations. It copies only the last token of an entity from the input sentence.

CopyMTL (Zeng et al., 2019) construct their models based on CopyRE and propose a multitask learning framework used to extract complete entities.

MrMep (Chen et al., 2019) proposes a novel architecture that augments the encoder and decoder in two elegant ways. First, they apply a binary CNN classifier for each relation. Second, they perform a multi-head attention over the text and a triplet attention with the target relation interacting with every token.

WordDec (Nayak and Tou Ng, 2019) utilizes a word-level decoder and copy mechanism to generate target sequence token-by-token

PtDec (Nayak and Tou Ng, 2019) is originated from the same paper as WordDec, it uses a pointer network-based decoder to generate the target sequence.

We follow Takanobu et al. (2018) for evaluation, where each extracted triplet is recognized as correct only if the full entity names and the corresponding relations are all correct. The performance is calculated in terms of precision, recall, and F1 score.

A.2 Detailed Dataset Statistics

	NYT29		NYT24	
	Train	Test	Train	Test
relations	29	29	24	24
examples	63,306	4,006	56,196	5,000
triplets	78,973	5,859	88,366	8,120
2 token	42,920	2,718	37,352	3,335
3 token	6,410	406	6,362	566
3+ tokens	1,833	116	1,259	112

Table 5: Statistics of train/test split of the two datasets. n-token denotes the number of examples that contain n-token entities

A.3 Experimental Settings

The model structure of *Unilm* uses a 24-layer Transformer with 1024 hidden size and 16 self-attention heads. The model has been pretrained on English Wikipedia[2] and BookCorpus[3], as well as preprocessed in the same way as Devlin et al. (2019). In the fine-tuning stage, we optimize network parameters by Adam (Kingma and Ba, 2015) with a $3e-5$ learning rate. The dropout rate is 0.1 and weight decay is 0.01. We also set up the maximum length of input sentence is 512, the vocabulary size is 28996. The tokens and their corresponding BIO labels are masked with 15 % possibility. The trade-off parameter α is set to 0.1.

[2] https://www.english-corpora.org/wiki/

[3] https://github.com/soskek/bookcorpus

A.4 Mask Strategy Demonstration

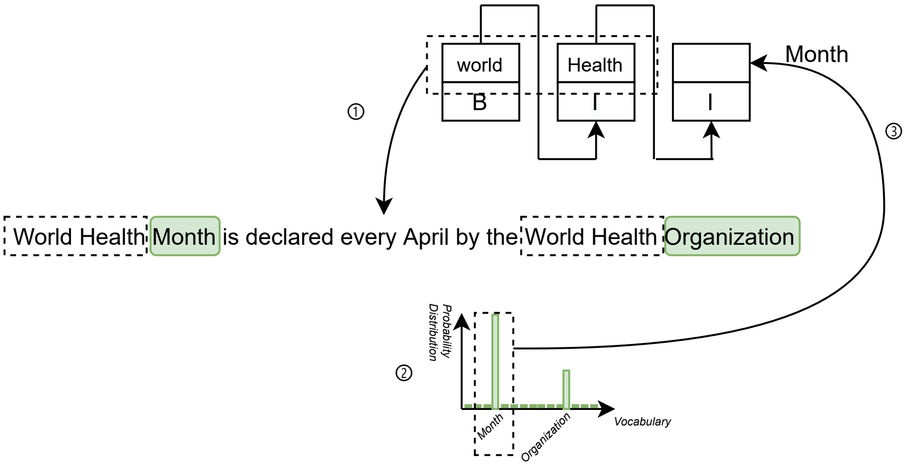
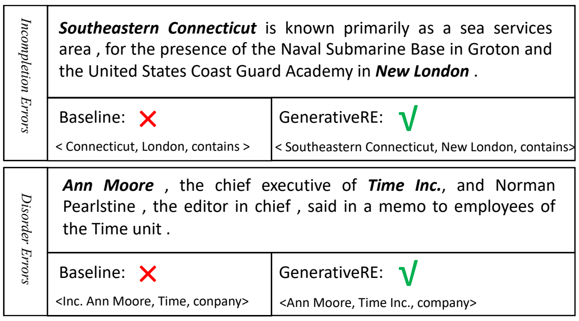

world
Health
B
I
I
Month
①
②
③
World Health Month is declared every April by the World Health Organization
Probability Distribution
Vocabulary
Month
Organization

Figure 2: **Mask strategy workflow when BIO = I**: To predict the current token, there are three steps: (1) we collect the previously predicted token until we meet the nearest BIO = B, which are "world", "health" in this case. (2) We use "world", "health" to match the same sequence in the input text and collect all the next tokens "Month", "Organization" as our candidates. (3) We mask all the token distribution except candidate token "Month", "Organization", so that our model decoder will have a much higher possibility to gain the correct prediction and avoid incompletion and disorder problem, accordingly.

A.5 Case Study



Incompletion Errors	***Southeastern Connecticut*** is known primarily as a sea services area , for the presence of the Naval Submarine Base in Groton and the United States Coast Guard Academy in ***New London*** .	
	Baseline: ✗ < Connecticut, London, contains >	GenerativeRE: √ < Southeastern Connecticut, New London, contains>
Disorder Errors	***Ann Moore*** , the chief executive of ***Time Inc.***, and Norman Pearlstine , the editor in chief , said in a memo to employees of the Time unit .	
	Baseline: ✗ <Inc. Ann Moore, Time, conpany>	GenerativeRE: √ <Ann Moore, Time Inc., company>

Figure 3: The extracted samples are error cases in baseline models while being predicted correctly in GenerativeRE.