BioCopy: A Plug-And-Play Span Copy Mechanism in Seq2Seq Models

Yi Liu [1], Guoan Zhang [2], Puning Yu [3], Jianlin Su [4], Shengfeng Pan [4]

[1] RMIT University, Melbourne, Australia
[2] King's College London, the United Kingdom
[3] South China University of Technology, China
[4] Shenzhen Zhuiyi Technology Co., Ltd., China
yiliumelai@gmail.com, guoan.zhang@kcl.ac.uk,
ypn20010116@163.com, {bojonesu,nickpan}@wezhuiyi.com

Abstract

Copy mechanisms explicitly obtain unchanged tokens from the source (input) sequence to generate the target (output) sequence under the neural seq2seq framework. However, most of the existing copy mechanisms only consider single word copying from the source sentences, which results in losing essential tokens while copying long spans. In this work, we propose a plug-and-play architecture, namely BioCopy, to alleviate the problem aforementioned. Specifically, in the training stage, we construct a BIO tag for each token and train the original model with BIO tags jointly. In the inference stage, the model will firstly predict the BIO tag at each time step, then conduct different mask strategies based on the predicted BIO label to diminish the scope of the probability distributions over the vocabulary list. Experimental results on two separate generative tasks show that they all outperform the baseline models by adding our BioCopy to the original model structure.

1 Introduction

Recent neural seq2seq systems have been successful in various NLP tasks which utilize an encoder to convert a source sentence into a fixed vector, and a decoder to generate a target sentence by using the semantic information amongst the fixed vectors. However, seq2seq suffers from the out-of-vocabulary and rare word problem(Luong et al., 2015; Gulcehre et al., 2016) that words in source sentence are not able to be obtained to generate the target sentence. Due to this problem, (He et al., 2016; Srivastava et al., 2015), propose the so-called 'copy mechanism' where it locates certain words in the input sentence and put these words into the target sequence. In their work, every output word can be generated either by predicting from the vocabulary or copying from the source sequence. Unfortunately, most of them copy the words from the source sentence to the target sentence in a word-by-word manner. However, in many cases, the copied words are generated consecutively from a span in the source sequence.

In this paper, we propose a novel and portable copy mechanism to solve the problem of low accuracy when copying spans in the seq2seq framework. To implement this new copy mechanism, we propose a BIO-tagged strategy that annotates the target sequences with BIO tags by matching the longest common subsequence (LCS) with the source sequence. Therefore, the BIO tags are able to exactly locate the start and end position of every single span in target sequences. In the inference stage, we design a span extractor to determine copying spans from the source sequence. Specifically, the BIO tag is firstly predicted at each time step to indicate the position of the copied span, then we decide a copy algorithm to guide the span extractor by generating the eligible n-gram set.

Since our method does not change the structure of the model itself, this BioCopy can be seen as a plug-and-play component, which is simple and effective enough to transfer and apply to any generative seq2seq framework. The experimental results in generative relation extraction and abstractive summarization tasks indicate the effectiveness of our proposed copy mechanism.

2 Approach

The conventional seq2seq framework generates one token at each time step. Therefore, given the predicted token sequence, the mode decoder predicts the current token by computing the probability distribution over the entire vocabulary list:

$$y_t = p(y_t|y_{<t}, x) \qquad (1)$$

In our case, we add an extra sequence prediction task to the model decoder. Specifically, the former decoder just models the distribution on each token, but the current decoder predicts an additional label

distribution:

$$y_t, z_t = p(y_t, z_t | y_{<t}, x) = p(y_t | y_{<t}, x) p(z_t | y_{<t}, x) \tag{2}$$

where $z \in \{B, I, O\}$. The meaning is as follows:

- **B** indicates the current token is copied from the source sentence.

- **I** indicates the current token is copied and forms a continuous fragment with the previous token from source sentence.

- **O** indicates the current token is not copied, but generated from the vocabulary list.

BIO-tag Building   As we mentioned above, we utilize the supervised method to train the BIO tag z. In order to acquire the labels, we compute the longest common subsequence (Paterson and Dancík, 1994) between the source sentence and target sentence. The token will be considered as copied from source sentence as long as it appears in the longest common subsequence, and different tags are assigned according to the specific meaning of BIO. In summary, in the training phase, besides the original token sequence prediction task, we also introduce one more sequence prediction task, of which the tags are all given. It is easy to implement and does not add any extra computational cost. The loss function $\mathcal{L}$ is defined as the following:

$$\mathcal{L} = \frac{1}{NT} \sum^{N} \sum_i^{T} y_i' \log y_i + z_i' \log z_i \tag{3}$$

where y' and z' denotes gold label of token sequence and BIO-tag sequence, T is the sequence length and N is the batch size. y and z denotes predicted label.

Inference   In the inference stage, at each time step, we first predict the BIO-tag z_t, the result will be one of three circumstances: if $z_t = O$, we do not need to change anything. If $z_t = B$, we mask all the token probability distributions of which they are not in the source sentence. If $z_t = I$, it indicates the token sequence from current token to its nearest token with $z_t = B$ constitutes a consecutive n-gram from source sentence. Therefore, we mask all the tokens in the token probability distribution that cannot constitute the corresponding n-gram in the source sentence. In this way, the model decoder still generates tokens in a step-by-step manner, rather than generating a segment at

one time step. According to utilise the mask operation, the tokens where their $z_t = B$ or $z_t = O$ are selected from a segment in the input sentence. A detailed example has been shown in Figure 1 for illustration.

It should be pointed out that the proposed copy mechanism can not only improve the model performance in terms of long-span extraction, but also ensures the consistency between the generated text and the original text, thereby avoiding professional errors, which is quite necessary for practical use.

3 Experiment

3.1 Generative Relation Extraction

Generative relation extraction tackles the conventional relation extraction problem by utilizing the seq2seq framework. At each time step, the model decoder either predicts the relation or copy a token from the input sentence. We focus on the task of extracting multiple tuples from sentences. We choose the New York Times (NYT) corpus(Zeng et al., 2018) for our experiments. The detailed statistics are listed in Table 1. Intuitively, the model capability of extracting long-span entities can be boosted by adding the proposed BioCopy.

	Train	Test
examples	56,000	5,000
triplets	88,366	8,120
2 token	37,352	3,335
3 token	6,362	566
3+ tokens	1,259	112

Table 1: Statistics of train/test split of the NYT. n-token denotes the number of examples that contain n-token entities.

Baselines   We compare our model performance with the following state-of-the-art relation extraction models. **Tagging**(Zheng et al., 2017) is a neural sequence labeling model which jointly extracts the entities and relations using an LSTM encoder and an LSTM decoder. **CopyR**(Zeng et al., 2018) uses an encoder-decoder framework to extract entities and relations jointly. **GraphR**(Fu et al., 2019) models each token in a sentence as a node, and edges connecting the nodes as relations between them, they adopt the graph neural network to predict the relation triplets. **Ngram-att** uses an encoder-decoder framework with a n-gram attention layer. The encoder takes source sequence as input and decoder produces entity and relation IDs from Wikidata(Vrandečić and Krötzsch,
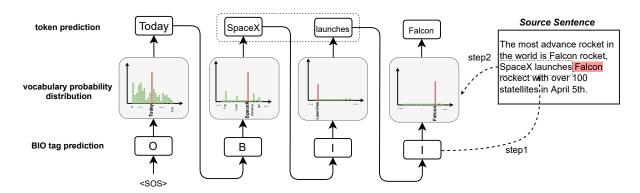
Figure 1: The workflow of model decoder in the inference stage: in order to predict the current token 'Falcon', the model firstly predicts its BIO = I, which indicates the current token constitutes a 3-gram with the predicted tokens 'SpaceX', 'launches'. Then, we search the potential 3-grams from the source sentence, and the 'Falcon' is the only valid candidate. Finally, the token probability distributions except 'Falcon' are all masked as zero.

2014). **WordDec** and **PointerDec** are both originated from (Nayak and Ng, 2020), where they use both word decoder and pointerNet as model decoder.

Model Details Since WordDec (Nayak and Ng, 2020) is the state-of-the-art model on the NYT dataset, thus we select WordDec as our backbone model and all the model details are aligned with the original settings. Concretely, we initialize the word vectors by using Word2Vec(Mikolov et al., 2013). We set the word embedding dimension d_w = 300 and relation embedding dimension d_r = 300, the hidden dimension d_h of the LSTM cell is set at 300. The model is trained with the mini-batch size of 32 and the network parameters are optimized using Adam(Kingma and Ba, 2015). Dropout layers with a dropout rate fixed at 0.3 are used in our network to avoid overfitting.

Models	Precision	Recall	F1 score
Tagging	62.4%	31.7%	42.0%
CopyR	61.0%	56.6%	58.7%
GraphR	63.9%	60.0%	61.9%
Ngram-att	78.3%	69.8%	73.8%
PointerDec	80.6%	77.3%	78.9%
WordDec	**88.1**%	76.1%	81.7%
Our model	87.7%	**77.7**%	**82.4**%

Table 2: Performance comparison on NYT24 dataset

Result We run the model with the above experimental settings, and we get the result as shown in Table 2. Our model outperforms the state-of-the-art methods 0.4% by recall and 0.7% by F1 score respectively.

To further verify the effectiveness of the proposed BioCopy, We conduct the ablation experiment, and the results are presented in Table 3. To be more specific, we firstly use a raw seq2seq model without any involvement of copy mechanism. Then, we add the *Attention Copy* that has been conducted by WordDec (Nayak and Ng, 2020), i.e., if the predicted token is unknown, we will select the token with the highest attention score from the input sentence . As we can see from Table 3, *raw seq2seq* unsurprisingly turns out to be the lowest performance due to the massive prediction of unknown. Adding *Attention Copy* can alleviate the unknown problem, but it is still a lack of capability of extracting long-span entities. By adding our *BioCopy*, the model performance exceeds the other two since our model is able to not only deal with the unknown problem, but extracting the long-span entities properly.

	Precision	Recall	F1 score
Raw Seq2seq	71.4%	59.6%	65.0%
+ Attention Copy	**88.1**%	76.1%	81.7%
+ BioCopy	87.7%	**77.7**%	**82.4**%

Table 3: Model performance with different settings

Table 3 suggests our approach boosts the model performance to some extent. However, it is still not clear how effectively could our method act on multi-token entity extraction. Since the multi-token entities can be regarded as a long span, we speculate that our method can improve the model capability in terms of tackling the multi-token entities, We conduct an auxiliary experiment to verify this. As shown in Table 4, we can notice that the percentage of error cases, which are caused by incorrectly extracting long-span entities, drop from 49.1% to 18.6% by adding our BioCopy where it

strongly verifies that the improvement of the model performance in this task is due to the reduction of long-span copy errors.

	Error Percentage
Raw Seq2seq	49.1%
+ Attention Copy	23.7%
+ BioCopy	18.6%

Table 4: Percentage of error cases of extracting long-span entities.

3.2 Auxiliary Experiment

To verify the robustness of our proposed method, we also conduct an auxiliary experiment on abstractive summarization task. Abstractive summarization task aims to generate a new shorter text that conveys the most critical information from the original long text, where it usually requires to generate much longer text spans from the source sentenc, which can be leveraged to further evaluate our Bio-Copy.

We conducted our experiment on a Chinese legal summarization dataset (CAIL2020) [1], which contains a large number of legal terms. CAIL2020 dataset has 9,484 sample pairs. Each source text contains an average of 2569 words and each summary text contains 283 words.

Baseline and Metrics Recently, pre-training models have achieved promising results when fine-tuned on several text summarization tasks (Dong et al., 2019a; Lewis et al., 2020). We choose NEZHA (Dong et al., 2019a) as our backbone model, which is a large-scale Chinese pre-trained model and is able to encode the input text with any length. For a fair comparison, we convert the original encoder mask of NEZHA to the seq2seq mask as same as the BERT-UniLM (Dong et al., 2019b) used in Table 5. We use ROUGE scores for evaluation, in which Total is calculated as a weighted average of the above scores.

Result In Table 5, we first compare the model performance with BERT-based models. We can notice that BERT model can achieve a better result by utilizing our BioCopy than by just simply using UniLM (Dong et al., 2019b). Since the dataset is in Chinese, we select NEZHA (Dong et al., 2019a) as our backbone model as it is pre-trained on massive Chinese corpus. The results in Table 5 show that

---

[1] https://github.com/china-ai-law-challenge/CAIL2020/tree/master/sfzy

Models	Rouge-1	Rouge-2	Rouge-L	Total
LSTM-Seq2seq	46.48	30.48	41.80	38.21
BERT-UniLM	63.83	51.29	59.76	57.19
BERT-BioCopy	64.98	53.92	66.54	61.18
NEZHA	70.93	54.38	69.89	63.89
NEZHA-BioCopy	**71.31**	**54.72**	**70.29**	**64.29**

Table 5: Performance comparison on CAIL2020 dataset

the model can gain 71.31 Rouge-1, 54.72 Rouge-2, 70.29 Rouge-L and 64.29 Total, respectively, where it surpasses all the baseline models.

4 Related Work

Generative relation extraction. (Zeng et al., 2018) proposed CopyRE, a joint model based on a copy mechanism, which transforms the joint extraction task into a generation task.(Nayak and Ng, 2020) propose two different encoders. The word decoder generates multiple triplets in a token-by-token manner, and each triplets is distinguished by the special token, while the pointer decoder simply utilizes pointerNet (Vinyals et al., 2015) to generate start and end indexes for each entity. Copy-MTL (Zeng et al., 2019) introduced a multi-task framework with a sequence labeling layer in the encoder, to alleviate the problem that CopyRE can only extract the last token in a multi-token entity.

Abstractive summarization. The previous work mainly leverage an encoder-decoder framework by choosing different model structures. (Zhong et al., 2019) utilize Transformers or graph neural network (Wang et al., 2020) for model encoder. Despite most of the previous work conducting the seq2seq model, some works (Wang et al., 2019) deploy a non-autoregressive model decoder to tackle this task, which also shows great effectiveness.

5 Conclusion

In this paper, we propose BioCopy, a plug-and-play copy mechanism to alleviate the long-span copying problem in generative tasks. By adding an extra sequence prediction layer in the training stage, our proposed approach is able to diminish the scope of probability distribution on each token. Experiments in generative relation extraction and abstractive summarization verifies the model's effectiveness.

References

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019a. 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.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019b. Unified language model pre-training for natural language understanding and generation. *arXiv preprint arXiv:1905.03197*.

Tsu-Jui Fu, Peng-Hsuan Li, and Wei-Yun Ma. 2019. Graphrel: Modeling text as relational graphs for joint entity and relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1409–1418.

Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. 2016. Pointing the unknown words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 140–149, Berlin, Germany. Association for Computational Linguistics.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society.

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*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Thang Luong, Ilya Sutskever, Quoc Le, Oriol Vinyals, and Wojciech Zaremba. 2015. Addressing the rare word problem in neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 11–19, Beijing, China. Association for Computational Linguistics.

Tomás Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 3111–3119.

Tapas Nayak and Hwee Tou Ng. 2020. Effective modeling of encoder-decoder architecture for joint entity and relation extraction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8528–8535.

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.

Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015. Highway networks. *CoRR*, abs/1505.00387.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer Networks. *arXiv e-prints*, page arXiv:1506.03134.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

Danqing Wang, Pengfei Liu, Yining Zheng, Xipeng Qiu, and Xuanjing Huang. 2020. Heterogeneous graph neural networks for extractive document summarization. *arXiv preprint arXiv:2004.12393*.

Danqing Wang, Pengfei Liu, Ming Zhong, Jie Fu, Xipeng Qiu, and Xuanjing Huang. 2019. Exploring domain shift in extractive text summarization. *arXiv preprint arXiv:1908.11664*.

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. *arXiv preprint arXiv:1706.05075*.

Ming Zhong, Danqing Wang, Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2019. A closer look at data bias in neural extractive summarization models. *arXiv preprint arXiv:1909.13705*.