PSG: Prompt-based Sequence Generation for Acronym Extraction

Bin Li (Co-first author)[1], Fei Xia (Co-first author)[2,3], Yixuan Weng[2], Xiusheng Huang[2,3], Bin Sun[1] and Shutao Li (Corresponding author)[1]

[1]*College of Electrical and Information Engineering, Hunan University*

[2]*National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy Sciences*

[3]*School of Artificial Intelligence, University of Chinese Academy of Sciences*

Abstract

Acronym extraction aims to find acronyms (i.e., short-forms) and their meanings (i.e., long-forms) from the documents, which is important for scientific document understanding (SDU@AAAI-22) tasks. Previous works are devoted to modeling this task as a paragraph-level sequence labeling problem. However, it lacks the effective use of the external knowledge, especially when the datasets are in a low-resource setting. Recently, the prompt-based method with the vast pre-trained language model can significantly enhance the performance of the low-resourced downstream tasks. In this paper, we propose a **P**rompt-based **S**equence **G**eneration (**PSG**) method for the acronym extraction task. Specifically, we design a template for prompting the extracted acronym texts with auto-regression. A position extraction algorithm is designed for extracting the position of the generated answers. The results on the acronym extraction of Vietnamese and Persian in a low-resource setting show that the proposed method outperforms all other competitive state-of-the-art (SOTA) methods.

Keywords

Acronym Extraction, Document Understanding, Prompt Learning, Language Model

1. Introduction

With the development of technology and global informatization, the number of acronyms is increasing rapidly. The forms of the acronyms are complicated and changeable because of various languages [1]. Therefore, understanding the acronyms of long technical phrases is very important for scientific document understanding (SDU@AAAI-22) [2].

In addition, the carefully designed document reading system should be able to recognize the correct meaning of acronyms and their long forms so that these documents can be processed correctly. This is fairly critical for a variety of downstream tasks, such as question and answer [3], reading comprehension [4], translation [5], medical consultant [6], etc.

The acronym extraction task is mainly used to extract acronyms (i.e., short forms) and their meanings (i.e., long forms) in the science document [7]. To take the English language as an example, which is shown in Figure 1, the output label is the position of the input text string. In the earliest attempts, the rules or features [8] are adopted to capture the acronyms. However, these methods often

Figure 1: Examples of English acronym extraction.

require a lot of manual design, making it hard to process documents with complex grammatical structures.

Recent works tend to model this task as a sequence labeling task [9, 10, 11], which helps the model to capture the local definition of acronyms in the document. However, the form of acronyms not only appears in the English scene but also in other language scenes (multilingual). Traditional sequence labeling methods are weak in utilizing external knowledge [12]. As a result, once the system runs on low-resource downstream tasks, its performance is relatively poor. Furthermore, the traditional sequence labeling method requires a lot of manual rules for labeling, which is unrealistic in low-resource scenarios. Inspired by the prompt-based method [13], the prompt tends to extract the knowledge from the large-scale pre-trained model [14], which is helpful to improve the generalization ability in low-resource scenarios.

In this paper, we propose the **P**rompt-base **S**equence **G**eneration (**PSG**) method for acronym extraction. Specifically, we design a prompt for sequence generation with the large-scale pre-trained model. With the prompt being used as a template, the acronym form and the long-form can be generated via auto-regression. After obtaining the answer, we designed a position extraction algorithm to locate the labels. The proposed method ranks 1-st un-

der the low-resource language setting (i.e., Vietnamese and Persian) in the shared task 1 of the SDU@AAAI-22, which outperforms all other competitive methods. The main contributions are summarized as follows:

- As far as we know, this is the first attempt to adopt prompt-based sequence generation for the acronym extraction task.
- We propose a novel acronym extraction method, including prompt-based sequence generation method and position extraction algorithm for obtaining the final labels.
- Extensive experiments are conducted on the low-resource datasets (i.e., Vietnamese and Persian). The results demonstrate the effectiveness of our proposed method compared with other competitive baselines.

2. Task introduction

2.1. Problem definition

We treat the acronym task as a sequence generation problem. Given a series of tokens in the text $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$, this task aims at finding the corresponding position from the original text. The label indicates the short-form S (i.e., acronym) and the long-form L (i.e., phrase). We formulate the above process as follows:

$$L, S = h(x_1, x_2, \ldots, x_n) \tag{1}$$

where h is the model which extracts the answers.

2.2. Evaluation metric

The submitted results will be evaluated based on the macro-averaged precision, recall, and F1 scores on the online test set. The final scores represent the prediction correctness of short-form (i.e., acronym) and long-form (i.e., phrase) boundaries in the sentences. The short-form or long-form boundary prediction is counted as correct if the beginning and the end of the predicted short-form or long-form boundaries are equal to the ground-truth beginning and end of the short-form or long-form boundary, respectively. The official score is the macro average of short-form and long-form F1 scores.

2.3. Dataset

This acronym extraction task consists of various multi-lingual datasets composed of document sentences in science fields [15]. Among them, the Vietnamese dataset and the Persian dataset are set in a low-resource scenario compared to other languages. As shown in Table 1, the Vietnamese dataset is divided into training (1274), development (159), and testing according to the data set (160).

Table 1
Statistical Information of Vietnamese Dataset.

Data	Sample Number	Ratio
Training Set	1274	79.98%
Development Set	159	9.98%
Test Set	160	10.04%
Total	1593	100%

Table 2
Statistical Information of Persian Dataset.

Data	Sample Number	Ratio
Training Set	1336	80.34%
Development Set	167	10.04%
Test Set	160	9.62%
Total	1663	100%

As shown in Table 2, the Persian dataset is divided into training (1336), development (167), and testing according to the data set (160). The training and validation sets of the above two datasets have been manually labeled, where the label is a list of position boundaries.

3. Method

In this section, we will introduce our method in detail, including the model architecture, prompt design, sequence generation and position extraction algorithm.
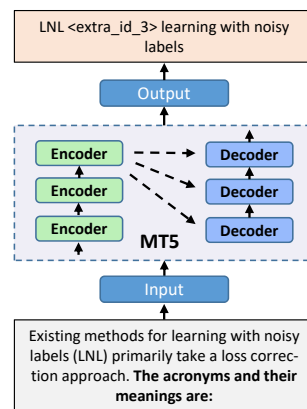

Figure 2: Overview of model architecture, where the sentence in bold is the prompt and the $<$extra_id_3$>$ is designed to separate the acronym of short-form and long-form.

3.0.1. Model architecture

The overall architecture of our me-thod is shown in Figure 2, the MT5 model [16] is adopted as our backbone for sequence generation. We first input the text with a manually designed prompt to be tokenized with MT5 tokenizer,

then the input shall be encoded with the encoder through the self-attention [17] mechanism. Finally, the output is produced by the decoder via auto-regression. Notice that the output contains the unused token, which is designed as the placeholder for prompt tuning, thus further utilizing the external knowledge from the pre-trained model.

3.0.2. Prompt design

We manually design the prompt to extract relevant knowledge from the pre-trained model for sequence generation, which is presented as the fixed tokens, i.e., "The acronyms and their meanings are:". In addition, the unused tokens are adopted as a placeholder to control the outputs. Specifically, the unused tokens are used as the placeholder to form a template for prompt tuning, where the <extra_id_1> represents the separator of the short forms, the <extra_id_2> represents the separator of the long forms, and the <extra_id_3> represents the separator between the acronym of the long-form and short-form. The <extra_id_4> indicates that no acronym of short-form appears, while the <extra_id_5> indicates that no acronym of long-form appears.

3.0.3. Sequence generation

The sequence generation task is designed to generate extracted possible acronyms. The text $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ is encoded through the encoder to obtain the context encoding E_C. At the decoding stage, the loss function of the sequence generation task can be performed as auto-regression, which is shown as equation (2):

$$L_D(\varphi) = -\sum_i \log P_\varphi \left(y_i \mid y_0, \ldots, y_{i-1}, E_C \right) \quad (2)$$

where the φ is the parameters of the model, the y_i represents the i-th word generated by the decoder, and $y_0, \ldots, y_{i-1}$ is a sequence of previously generated tokens.

3.0.4. Position extraction algorithm

After the possible acron-yms of short-form and long-form are generated with sequence generation, the next step is to consider how to extract their positions. A good extraction algorithm determines the final quality of the generated result. We use a greedy traversal search method, adopting the regular method from left to right to find the corresponding location boundary. At the same time, we need to ensure there is no overlap in the extracted outputs by detecting the boundary margins so that the extracted positions are independent of each other. To take the acronym of short-form as an example, the algorithm is represented in Algorithm 1.

Algorithm 1 Position extraction algorithm
Input: Output text $\mathbf{y}$ tokenized into list L; Original input text $\mathbf{x}$
Output: Short-form acronym position list D
1: Let D = []
2: Let truncated text $\mathbf{R} = \emptyset$
3: **if** '<extra_id_4>' in L **then**
4:     D = []
5: **else**
6:     i = $\mathbf{y}$.find('<extra_id_3>')
7:     $\mathbf{R}$ = $\mathbf{y}[:i]$.strip()
8:     **for** j in $\mathbf{R}$.split(' <extra_id_1> ') **do**
9:         **for** m in [k.start() for k in re.finditer(j, $\mathbf{x}$)] **do**
10:             D.append([m, m + len(j)])
11:         **end for**
12:     **end for**
13: **end if**
14: Making the position intervals in list D independent of each other
15: **return** solution D

---

4. Experiment setup

4.1. Baseline models

- **Rule-based method** The baseline method proposed by Schwartz is a rule-based method [8]. In this baseline, the words that more than 60% of their characters are uppercased are selected as acronym. To select long-forms, if the initial characters of the preceeding words before an acronym can form the acronym they are selected as long-form. The related codes can be found on the website[1].

- **BERT-CRF model** The BERT-CRF [9] architecture is composed of a BERT model [18] concatenated with a token-level perception layer with a conditional random field (CRF) on top. For the input tokens, the BERT model produced encoded tokens and the classification model projects encodings to the label space. The classification output scores are then sent to the CRF layer, whose parameters are the tag transition matrix, where the elements represent the tag transition score. The matrix contains two states: begin (B) and end (E). We only consider the cross-entropy loss of the first sub-token of each token.

- **BERT-Span model** The task is also considered as the boundary of phrase spans modeled by BERT-Span model [10], including acronyms of short-form and long-form. Two binary classifiers are

---

[1]https://github.com/amirveyseh/AAAI-22-SDU-shared-task-1-AE

Table 3
F1 Performace in Vietnamese.

Method	Val F1	Test F1
Rule-based	0.5337	0.5646
BERT-CRF	0.7712	0.7613
BERT-Span	0.8118	0.7844
MT5-base	0.8222	0.8012
PSG-base	0.8313	0.8195
PSG-large	0.8523	0.8344
PSG-xlarge	**0.8611**	**0.8416**

Table 4
F1 Performace in Persian.

Method	Val F1	Test F1
Rule-based	0.5811	0.5636
BERT-CRF	0.6126	0.5782
BERT-Span	0.6334	0.5916
MT5-base	0.6578	0.6613
PSG-base	0.6901	0.7077
PSG-large	0.7363	0.7437
PSG-xlarge	**0.7745**	**0.7993**

adopted to output the multiple start and end indexes. The prediction represents whether each token is the start or end tag. Given each token representation from the BERT model, the probabilities of each token are predicted as the start or end position. The weighted cross-entropy loss is implemented to train the model with parameters shared at the BERT encoder layers.

- **MT5 model** The MT5 [16] is a multilingual Transformer model pre-trained on a dataset (mC4) containing text from 101 different languages include the language of Vietnamese and Persian. The architecture of the MT5 model, which is based on T5 [19], is designed to support any Natural Language Processing task by reframing the required task as a sequence-to-sequence task. Also, the MT5 model has different variants to perform the sequence generation task, where we final adopt the model size of the base, the large and the x-large. The above models can be found and downloaded on the website[2].

4.2. Training strategies

For the sequence generation part, we design a curriculum learning method to train the sequence generation model, where the model is fine-tuned with a corpus of different difficulties. We mix up all the training datasets in the acronym tasks, including 4,000 English, 1,000 Persian, and 800 Vietnamese paragraphs in the scientific domain and 4,000 English, 8,000 French, 6,400 Spanish, and 3,000 Danish paragraphs in the legal domain. The training steps are as follows:

1. The trained MT5 model is utilized to initialize the parameters of the encoder and decoder, and fine-tune with the multi-lingual data. Finally, we train the model with 8 epochs.
2. We fine-tune the MT5 model with a single language for Vietnamese and Persian respectively with 8 epochs being trained.

[2]https://huggingface.co/models

Notice that the Persian is a right-to-left language, it is different from other languages. However, we deem that the whole training process is an auto-regressive task, where the pre-trained model can learn the features of multi-languages via self-supervised pre-training and fine-tuning. In short, the model can well produce the results with the proposed position extraction algorithm.

4.3. Implementation

All models are implemented based on the open-source transformers library of huggingface [20], where thousands of pretrained models are provided to perform different tasks on texts such as sequence labeling and sequence generation. The huggingface toolkit provides multiple APIs to quickly download and use those pre-trained models, thus fine-tuning them on the downstream tasks. We adopt Pytorch deep learning framework to finish this task. Specifically, we use four GPUs of NVIDIA 3090 with 24 cores to complete these experiments.

In BERT-CRF method, we initialize the model with mbert [21], and initial learning rates are 5e-5 and 5e-2 for BERT and CRF respectively. We utilize the AdamW optimizer [22] with a batch size of 32.

In BERT-Span method, we initialize the model with mbert, and initial learning rates are 5e-5. We utilize the AdamW optimizer with a batch size of 32.

Our models are implemented with various varients, where the base model is implemented with the batch size of 32, the large model is implemented with the batch size of 4 and the xlarge model is implemented with the batch size of 2.

As for multi-lingual fine-tune and single language fine-tuning, we used the AdamW optimizer with an initial learning rate of 1e-4 and annealed it gradually after a warm-up epoch until it reached 1e-5.

5. Results

The main results of our model and baselines are shown in Table 3 and Table 4, where the F1 performance in Vietnamese and Persian are presented respectively. It

can be found that the pre-trained model has more advantages than the rule-based method, since the rule-based method has limited generalization capabilities in validation and test datasets. The BERT-CRF and BERT-Span methods have similar performance. This may be because the sequence labeling method with pre-trained model has limited ability to capture external information in the unseen test dataset, especially in the low-resource setting. It is worth noting that the F1 scores of the method based on the generation method are higher than the sequence labeling method, which indicates that the generation method is more conducive to capturing the relationship between the acronym of the short-form and the long-form. What's more, our method is higher than other baselines in both the validation set and the test set. More precisely, on the Vietnamese test set, the proposed method reaches an F1 score of 0.8416, and on the Persian test set, the proposed method reaches a score of 0.7993. Further conclusion can be found that: 1) compared with the traditional generation method (i.e., MT5), we have designed the prompt method, which is effective to make full use of the knowledge from the pre-trained model. As a result, the proposed method gets a good performance on the test set as its stronger generalization ability. 2) As the scale of the initial pre-trained model increases, the prompt can enhance the generalization of downstream tasks.

6. Conclusion

We describe the sequence generation system used for submission in the acronym extraction task of SDU@AAAI-22. Three types of methods have been tried to complete this task, including the rule-base, sequence labeling, and sequence generation. The results show that the proposed method outperforms the other baselines, achieving the best performance (Top-1) in the acronym extraction of Vietnamese and Persian. It can be concluded that the sequence generation method has more ability for generalization than the sequence labeling. The prompt can further enhance the generalization ability. Furthermore, there are some promising works to be done in the future. As for the training method, it will be a wise way to merge the sequence labeling into the sequence generation as the multi-task learning. As for the prompt design, the soft prompt can be a better way to bridge the gap between the upstream and the downstream task than the fixed prompt templates.

Acknowledgement

References

[1] A. P. B. Veyseh, F. Dernoncourt, T. H. Nguyen, W. Chang, L. A. Celi, Acronym identification and disambiguation shared tasks for scientific document understanding, arXiv preprint arXiv:2012.11760 (2020a).

[2] A. P. B. Veyseh, F. Dernoncourt, Q. H. Tran, T. H. Nguyen, What does this acronym mean? introducing a new dataset for acronym identification and disambiguation, in: Proceedings of the 28th International Conference on Computational Linguistics, 2020b, pp. 3285–3301.

[3] W. Dijkstra, Y. Ongena, Question-answer sequences in survey-interviews, Quality and Quantity 40 (2006) 983–1011.

[4] B. Qiu, X. Chen, J. Xu, Y. Sun, A survey on neural machine reading comprehension, arXiv preprint arXiv:1906.03824 (2019).

[5] S. Yang, Y. Wang, X. Chu, A survey of deep learning techniques for neural machine translation, arXiv preprint arXiv:2002.07526 (2020).

[6] B. Li, E. Chen, H. Liu, Y. Weng, B. Sun, S. Li, Y. Bai, M. Hu, More but correct: Generating diversified and entity-revised medical response, arXiv preprint arXiv:2108.01266 (2021).

[7] S. Y. R. J. F. D. T. H. N. Amir Pouran Ben Veyseh, Nicole Meister, Multilingual Acronym Extraction and Disambiguation Shared Tasks at SDU 2022, in: Proceedings of SDU@AAAI-22, 2022.

[8] A. S. Schwartz, M. A. Hearst, A simple algorithm for identifying abbreviation definitions in biomedical text, in: Biocomputing 2003, World Scientific, 2002, pp. 451–462.

[9] L. Luo, Z. Yang, P. Yang, Y. Zhang, L. Wang, H. Lin, J. Wang, An attention-based bilstm-crf approach to document-level chemical named entity recognition, Bioinformatics 34 (2018) 1381–1388.

[10] H. Zhao, L. Huang, R. Zhang, Q. Lu, H. Xue, Spanmlt: A span-based multi-task learning framework for pair-wise aspect and opinion terms extraction, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 3239–3248.

[11] X. Huang, Y. Chen, S. Wu, J. Zhao, Y. Xie, W. Sun, Named entity recognition via noise aware training mechanism with data filter, in: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, Association for Computational Linguistics, Online, 2021, pp. 4791–4803. URL: https:

//aclanthology.org/2021.findings-acl.423. doi:10.18653/v1/2021.findings-acl.423.

[12] L. Liu, J. Shang, X. Ren, F. Xu, H. Gui, J. Peng, J. Han, Empower sequence labeling with task-aware neural language model, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 32, 2018.

[13] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, G. Neubig, Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing, arXiv preprint arXiv:2107.13586 (2021).

[14] C. Zheng, M. Huang, Exploring prompt-based few-shot learning for grounded dialog generation, arXiv preprint arXiv:2109.06513 (2021).

[15] S. Y. R. J. F. D. T. H. N. Amir Pouran Ben Veyseh, Nicole Meister, MACRONYM: A Large-Scale Dataset for Multilingual and Multi-Domain Acronym Extraction, in: arXiv, 2022.

[16] L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant, A. Barua, C. Raffel, mt5: A massively multilingual pre-trained text-to-text transformer, in: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2021, pp. 483–498.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in neural information processing systems, 2017, pp. 5998–6008.

[18] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, 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), 2019, pp. 4171–4186.

[19] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, arXiv preprint arXiv:1910.10683 (2019).

[20] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, et al., Huggingface's transformers: State-of-the-art natural language processing, arXiv preprint arXiv:1910.03771 (2019).

[21] J. Libovickỳ, R. Rosa, A. Fraser, How language-neutral is multilingual bert?, arXiv preprint arXiv:1911.03310 (2019).

[22] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, arXiv preprint arXiv:1711.05101 (2017).