Sequence Generation with Label Augmentation for Relation Extraction

Bo Li[1,2] *, **Dingyao Yu**[1,2] *, **Wei Ye**[1†], **Jinglei Zhang**[1,2], **Shikun Zhang**[1†]

[1]National Engineering Research Center for Software Engineering, Peking University
[2]School of Software and Microelectronics, Peking University
deepblue.lb@stu.pku.edu.cn, yudingyao@pku.edu.cn, wye@pku.edu.cn,
jinglei.zhang@stu.pku.edu.cn, zhangsk@pku.edu.cn

Abstract

Sequence generation demonstrates promising performance in recent information extraction efforts, by incorporating large-scale pre-trained Seq2Seq models. This paper investigates the merits of employing sequence generation in relation extraction, finding that with relation names or synonyms as generation targets, their textual semantics and the correlation (in terms of word sequence pattern) among them affect model performance. We then propose **R**elation **E**xtraction with **L**abel **A**ugmentation (RELA), a Seq2Seq model with automatic label augmentation for RE. By saying label augmentation, we mean prod semantically synonyms for each relation name as the generation target. Besides, we present an in-depth analysis of the Seq2Seq model's behavior when dealing with RE. Experimental results show that RELA achieves competitive results compared with previous methods on four RE datasets.

1 Introduction

Paradigm shift has been observed in an increasing number of tasks in recent years, that is some paradigms have the potential to solve diverse NLP tasks (Sun et al. 2021). For example, although most natural language understanding (NLU) tasks treat the classification-based methods as the default solution, sequence-to-sequence (Seq2Seq) models could achieve on-pair performances (Yan et al. 2021; Lu et al. 2022; Saxena, Kochsiek, and Gemulla 2022; Zhang et al. 2022b; Sun et al. 2022; Mao et al. 2022). With the help of pre-trained Seq2Seq models (Lewis et al. 2020; Raffel et al. 2020) and carefully designed schema, these methods directly generate various objectives, such as sentiment polarity, class name, entity type, etc.

There are also numerous works incorporating Seq2Seq models for relation extraction (RE). Previous efforts mainly focus on the following three aspects. The first one designs novel modules or model architectures, for example, Chen, Zhang, and Huang (2022) proposes a framework of text and graph to learn relational reasoning patterns for relational triple extraction. The second one utilizes various external datasets or resources to enrich the input (Cabot and Navigli 2021; Zhang et al. 2022a; Paolini et al. 2021; Lu et al. 2022).

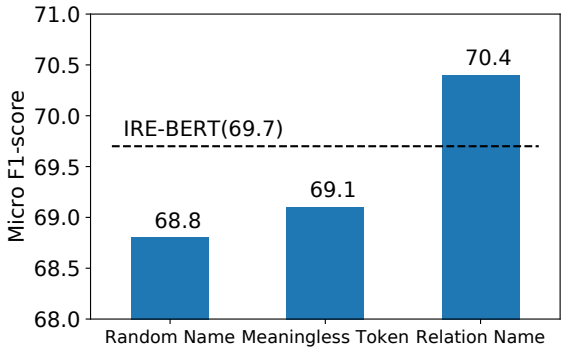

Figure 1: The results of different models on the TACRED test set. We report three kinds of BART-based variants with different generation objectives. The horizontal line is the result from IRE-BERT(Zhou and Chen 2021), which is a strong classification-based model. Note that we use BART-large and BERT-large as backbone networks. The results indicate that the generation target significantly affects the performance of Seq2Seq-based RE model.

The last one aims to design proper generation schema. For example, Josifoski et al. (2022) leveraged constrained beam search (Sutskever, Vinyals, and Le 2014; Cao et al. 2021) to generate reasonable outputs. Lu et al. (2022) proposed a structured extraction language to better guide the Seq2Seq model. By improving the model itself or enriching the input, the above methods could better utilize the knowledge existing in Seq2Seq model. A natural question is raised: does the target side could also active the knowledge and have significant impact on the Seq2seq-based RE model? In this research, we work towards closing the above research gap by enriching the information existing in relation names.

To answer the aforementioned questions, we first conduct a pilot experiment to explore whether Seq2Seq models could learn useful information from label spaces. Specifically, for a given input and two target entities, we force the BART to generate three kinds of labels: 1) **Relation Name**. We use original relation names as generation objectives directly. These labels *contain correct information*. 2) **Meaningless Token**. As its name implies, we replace relation names with some meaningless token, such as ö. These tokens *do not in-*

*These authors contributed equally.
† Corresponding author.
Copyright © 2023, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

clude any useful information. 3) **Random Relation Name**. We randomly map the original relation name to another relation name one by one and select the mapped one as the generation objective. Thus for a given input, the corresponding relation name *provides misleading information*.[1] The results are shown in Figure 1, from which we can see that directly generating correct relation names achieves the highest F1-score, while training with misleading label semantic information largely behind. The above results indicate that label information significantly affects Seq2Seq-based RE models, and these correct relation names contain valuable information.

In fact, relation names existing in most public RE datasets are meaningful phrases. These phrases describe corresponding relations and carry more or less available information. Normally, we find that most relation names contain two types of information: 1) **Semantic Information**. Relation names provide highly abstract semantics since they are mainly composed of nouns and verbs. 2) **Correlation Information**. Some relations have the same patterns or writing styles, e.g., *place of birth* and *place of death* both have the prefix *place of*. These patterns may exhibit some correlations between relations. We argue that Seq2Seq methods could utilize the above two types of information from relation names and benefit for RE (§3.4).

Based on the above observations, we raise another question: can we enhance the information in relation names, and achieve better performance? With the help of the powerful generative model GPT-2(Radford et al. 2019) and off-the-shelf synonym dictionary, in this research, we introduce **R**elation **E**xtraction with **L**abel **A**ugmentation (RELA). Specifically, we extend the relation names via three approaches: 1) **Paraphrase** that modifies a given relation name and some prefix phrases as input, then leverages GPT-2 to generate several related words, 2) **Inquiry** that takes each training text and a query as input and directly outputs the relation name by asking GPT-2, and 3) **Synonym** that retrievals several synonyms by utilizing an off-the-shelf synonym dictionary. Our experiments show that simply using several synonyms for each relation name could gain remarkable improvements(§3.5), especially for the domain-specific RE dataset and the low-resource scenario(§3.6).

Finally, we investigate the BART's behavior when dealing with classical generation tasks (such as summarization) and relation extraction. Our experiments show that BART exhibits distinctly different behaviors among different tasks. Interestingly, we find that when generating relation names, BART highly relies on the $\langle bos \rangle$ token and almost ignores previous decoder outputs, which is different from classical generation tasks. We report more visualization and in-depth analysis in the following part(§4). Below we summarize our main contributions:

1. We explore the advantages of utilizing Seq2Seq models for relation extraction task. We find that Seq2Seq models could obtain good performance by leveraging semantic information and correlation information existing in relation names.
2. We propose **R**elation **E**xtraction with **L**abel **A**ugmentation (RELA), a simply yet effective method to extend the relation names and achieves desirable performances among four RE datasets.
3. We present an in-depth analysis of the BART's behavior when dealing with RE and classical generation task and conduct some instructive conclusions, which may be useful for future work [2].

[1]Details of these label transformations can be found in the Appendix A.1.

2 Approach

In this section, we first describe the Seq2Seq method for relation extraction. Then, we will introduce three augmentation methods to enrich relation names in detail.

2.1 Sequence-to-Sequence Relation Extraction Model

For a given input sentence s contains N words, relation extraction aims at extracting the relationship between h and t, where h and t are two given target entities in s. Standard classification-based RE models transform labels into one-hot vectors with cross-entropy loss (Miwa and Bansal 2016; Guo, Zhang, and Lu 2019; Ye et al. 2019; Soares et al. 2019; Li et al. 2021; Roy and Pan 2021). However, these approaches largely ignore both semantic information and correlation information existing in relation names. While relation names are proven to be beneficial for RE in §1, we introduce the Seq2Seq models to leverage this valuable information via decoding the relation names directly. In fact, some previous works already explored Seq2Seq models for information extraction. They mainly focus on relational triple extraction with large-scale weakly supervised datasets pre-training (Cabot and Navigli 2021; Paolini et al. 2021). In this research, we argue that it is possible to directly generate relation names for the RE task, as shown in Figure 2 (a).

We treat relation extraction as a generation task following the standard Seq2Seq manner. We first highlight the target entities by enclosing entity spans with two special tokens, e.g., '@' for the head entity and '#' for the tail entity. The modified input sentence $\hat{s}$ is then passed into the transformer encoder. Finally, we use an auto-regressive transformer decoder to generate the relation name Y, where the label length is L. Note we generate each token in Y one by one. The training manner is similar to the classical Seq2Seq task, such as the summarization. In this research, we use pre-trained BART as the backbone network. We denote this Seq2Seq-based RE model as BART-RE. By fine-tuning BART on the downstream RE dataset, we minimize the sequential cross-entropy loss between the relation name and the model's output. The training procedure can be written as follows:

$$p(Y|\hat{s}) = \prod_{t=1}^{T} p(y_t|\hat{s}, y_{<t}), \tag{1}$$

[2]Code is available at https://github.com/pkuserc/RELA

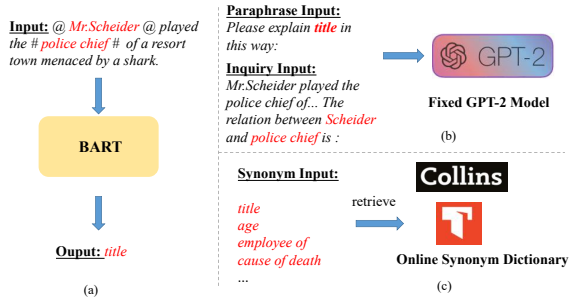

Figure 2: The Sequence-to-Sequence relation extraction model and three label augmentation methods. We take the following instance as an example: *Mr.Scheider played the police chief of a resort town menaced by a shark*. The target entities are *Mr.Scheider* and *police chief*, and the ground truth relation is *'title'*. In (a), we fine-tune BART to generate the relation name directly. In (b) and (c), we use GPT-2 and off-the-shelf online synonym dictionaries to augment label information automatically.

2.2 Relation Extraction with Label Augmentation

From the pilot experiment in (§1) we can see that the Seq2Seq-based RE model can learn valuable information from relation names and achieve impressive results. However, relation names usually contain few words, which may hinder BART to fully understand their meanings. Based on this, we argue that enriching the relation names would provide more supplementary information and benefit the model's performance. In this research, we introduce **R**elation **E**xtraction with **L**abel **A**ugmentation (RELA). By utilizing the powerful generative model GPT-2 [3] and off-the-shelf synonym dictionary, as shown in Figure 2, we propose three automatic label augmentation approaches: **Paraphrase**, **Inquiry** and **Synonym**.

1. **Paraphrase.** For a given relation name r, **Paraphrase** first converts r as a query by adding a prefix and then feed to the GPT-2 to explain the meaning of r. For example, relation name *'title'* will be transformed to *'Please explain title in this way: '*. To reduce the randomness with considering the variety, we use five different prefixes and run ten times with different random seeds for each of them. The above process can generate diverse and complementary explanations for the given relation name. After explaining all relation names, we use TF-IDF to distinguish the top-50 most related phrases for each relation. Finally, we manually select two most relevant phrases for each relation name as the augmented information.
2. **Inquiry.** This method directly uses the input text and a question to guide the GPT-2 output the target relation. For example, for the given input, *'Mr.Scheider played the police chief of a resort town menaced by a shark'*. The two target entities are *'Scheider'* and *'police chief'*. We use the following question to ask the GPT-2: *'Mr.Scheider played the police chief of a resort town menaced by a shark. The relation between Scheider and police chief is :'*.[4] We run fifty times for each input with different random seeds to generate more answers. Note that we only collect answers from the training set. We gather all answers for each relation name and form a long text, then use TF-IDF and the same selection process in **Paraphrase** to choose two most relevant phrases for each relation name.
3. **Synonym.** Although the above two approaches can generate some valuable information for relation names, useless information is unavoidable in the meanwhile. To further improve the quality of the augmented information, we utilize the off-the-shelf online synonym dictionary, e.g., collins dictionary[5] and thesaurus. [6] We directly retrieve several synonyms for each relation name as the augmented information.

The final generation objective is the concatenation of the original relation name and its corresponding augmented information. For example, given a relation name *'place of birth'* and its two synonyms *'birthplace'* and *'born in a place'*, the generation target for RELA is *'place of birth, birthplace, born in a place'*. When decoding, only exactly generating *'place of birth, birthplace, born in a place'* (instead of *'place of birth'*) can yield a correct relation prediction. Any mismatch (including OOV) will be treated as *'no relation'*. RELA can be viewed as an extension of the BART−RE model. Due to the space limitation, we only show the detailed augmented information for the Google RE dataset in Table 1. Others are shown in the Appendix A.5.

3 Experiments

3.1 Datasets

We evaluate our method on four RE datasets, i.e., TACRED (Zhang et al. 2017), SemEval2010 (Hendrickx et al. 2010), Google RE,[7] and sciERC (Luan et al. 2018). Specifically, TACRED, SemEval2010, and Google RE are commonly used in previous works. To verify the generalization ability in the domain-specific scenario, we also consider using sciERC in our experiments, a large RE dataset with scientific entities and their relations. Table 2 shows the detailed datasets information used in this research. Note that we use the modified version released by Beltagy, Lo, and Cohan (2019), because it removes lots of redundant sentences. We use the Micro-F1 score among positive relation labels (excluding 'no relation') as the evaluation metric. Following previous works (Yamada et al. 2020; Cabot and Navigli 2021), we do not include any entity type information, because the annotation of entity type is frequently violated in practice.

[3] We use the large version from https://huggingface.co/gpt2.

[4] The ground truth is 'title'

[5] https://www.collinsdictionary.com/

[6] https://www.thesaurus.com/

[7] https://github.com/google-research-datasets/relation-extraction-corpus

Relation Name	Paraphrase	Inquiry	Synonym
place of birth	born, family	first appearance, first time	birthplace, born in a place
education degree	college, student	law school, from university	study, graduate
institution	group, community	high school, new york	institution, organization,
date of birth	birth date, born	years old, one founders	birthday, time of birth,
place of death	death place, kill	died on, who died	dead in a palce, deathplace,

Table 1: Augmented information for each relation name via different approaches.

Dataset	#Train	#Dev	#Test	#Class
TACRED	68,124	22,613	15,509	42
SemEval	8,000	-	2,712	10
Google RE	38,112	9,648	9,616	5
sciERC	3,219	455	974	7

Table 2: Statistics of different RE datasets used in our experiments. SemEval2010 does not have validation set. #classes contains the 'No Relation' instances.

3.2 Experimental Setup

We use Pytorch (Paszke et al. 2019) and Tesla T4 GPU with a batch size of 8. For RELA and its variants, we use BART-large as the backbone network, and the checkpoint is downloaded here.[8] The maximum input/output sequence length is 256/32 tokens. As an optimiser, we used AdamW (Loshchilov and Hutter 2019) with a 1e-5 learning rate and a 0.2 warmup ratio. The training epoch is 20 for Semeval and 10 for other datasets. We choose the checkpoint that achieves the best result on the development set and evaluates the test set. To reduce the randomness, we run each model five times in the standard scenario and ten times in the low-resource scenario.

3.3 Comparison Models

Three types of models are considered in our comparison: 1) Classification-based models, LUKE (Yamada et al. 2020), MTB (Soares et al. 2019) and IRE (Zhou and Chen 2021), 2) Seq2Seq-based models, REBEL (Cabot and Navigli 2021) and TANL (Paolini et al. 2021), and 3) several variants of our models.

1. **Classification-based Model.** LUKE is a large pre-trained language model with external entity-annotated corpus, which achieves impressive results on several information extraction tasks. MTB leveraged entity linking datasets to generate better entity and relation representations. IRE is a powerful RE model without external corpus usage or additional pre-training.
2. **Seq2Seq-based Model.** REBEL pre-trained a BART-large model with a huge external corpus for relational triplet extraction. We download the pre-trained model from the open-source code,[9] and then fine-tune REBEL to directly generate relation names with the same input format as our method. As for TANL, it utilized T5 as the backbone network and took structured prediction as a translation task.
3. **Variants of Our Model.** We first build a BART model that directly generates relation names, named as BART-RE. To explore which information BART learns from relation names, we then design three kinds of variants: 1) BART-DS, which **D**rops the **S**emantic information in relation names while keeping their label relevance. Specially, every token in the label space will be transformed into a special meaningless token, which is a rare token from BART vocabulary. For example, *place of birth* will be transformed as *)+ cfffcc û*, and *place of death* will be transformed as *)+ cfffcc][*. 2) BART-DC is designed to keep semantic information while **D**rop the **C**orrelation information between relation names. We achieve this by replacing each relation name with its synonym while eliminating the correlation information. 3) BART-DB, this variant **D**rops **B**oth the semantic information and the label relevance within relation names. We achieve this by mapping each relation name into a unique meaningless token. In addition, to justify the effectiveness of the enhancement in relation names, we also test the performances of three different methods introduced in (§2.2), which are BART-Paraphrase, BART-Inquiry, and BART-Synonym. The detail of these training objectives can be found in the Appendix A.5. We choose two augmented phrases for each relation name, and the ablation study on the effectiveness of selection numbers can be found in the Appendix A.2.

3.4 Main Results

Table 3 shows the main results on the test sets. The first block exhibits classification-based models, the second block shows Seq2Seq-based models, and the last block is our method with several model variants. Rather than propose a new state-of-the-art method, we intend to propose a simple Seq2Seq model for RE in this research. We mainly compare our approach with strong RoBERTa-based models without using any external pre-training corpus, thus numbers in parentheses are improvements compared with IRE-RoBERTa. From Table 3 we can see that:

1. We find that BART-RE is a strong model equipped with Seq2Seq architecture. It performs on par or even better with IRE-RoBERTa. The above results indicate that directly generating relation names is an reliable solution for RE. Compared with three types of variants, ignoring semantic information (BART-DS) or correlation information (BART-DC) decreases the model's performance. Semantic information is more crucial than correlation information in terms of performance degradation. BART-DB has the worst performance since it drops the above two

[8] https://huggingface.co/facebook/bart-large
[9] https://github.com/Babelscape/rebel

Method	Semantic Information	Correlation Information	TACRED	SemEval	Google RE	sciERC
IRE-BERT	✗	✗	69.7	89.1	92.2	88.8
IRE-RoBERTa	✗	✗	70.5	89.8	93.1	88.9
MTB	✗	✗	71.5	89.5	92.7	87.4
LUKE	✗	✗	<u>72.7</u>	90.1	<u>94.0</u>	87.7
REBEL	✓	✓	70.7	82.0	93.5	86.3
TANL(T5)	✓	✓	71.9	-	-	-
BART-RE	✓	✓	70.4(-0.1)	89.7(-0.1)	93.3(+0.2)	88.8(-0.1)
BART-DS	✗	✓	69.3(-1.2)	89.4(-0.4)	93.1(+0.0)	88.3(-0.6)
BART-DC	✓	✗	70.1(-0.4)	-	93.0(-0.1)	88.6(-0.3)
BART-DB	✗	✗	69.1(-1.4)	89.4(-0.4)	92.2(-0.9)	88.1(-0.8)
RELA	✓	✓	**71.2(+0.7)**	**90.4(+0.6)**	**93.9(+0.8)**	**90.3(+1.4)**

Table 3: Performance of different methods on four relation extraction datasets. Results are all cited from public papers or re-implemented with official open-source code. RELA here is the implementation of BART−Synonym. Underlined results are previous state-of-the-art, bold results are the best performances RELA achieved. We left blank in BART−DC on the SemEval dataset since relation names in this dataset do not contain any similar writing style. Results of our methods are averaged over five random seeds, and the results are statistically significant with $p < 0.05$. We use the large version of BERT, RoBERTa, and BART in our experiments.

types of information. These results show that the Seq2Seq method learns both semantic information and correlation information existing in relation names, and achieves competitive performances compared with previous strong baseline models.

2. After enriching relation names, RELA obtains consistent improvements and achieve much better results than IRE-RoBERTa. We attribute this to RELA could learn more useful information from enhanced relation names. Interestingly, for the domain-specific dataset sciERC, RELA gains 1.4% F1-score improvements over IRE-RoBERTa, which is much bigger than the rest of the datasets. The lesson here is that RELA is more robust via label augmentation when original relation names are difficult to understand. This conclusion also can be learned from the comparison with LUKE. Although LUKE is pre-trained with a large external entity-annotated corpus, which helps LUKE to be more powerful on the dataset with commonly used relations, RELA outperforms LUKE by a large margin when dealing with domain-specific datasets, e.g., +2.9% F1-score on sciERC. The above results demonstrate that although the original relation name contains valuable information, it may not be enough for Seq2Seq methods. With the help of automatic label augmentation, RELA could strengthen the expressive power of Seq2Seq methods.

3. Compared with previous Seq2Seq-based RE architectures, although REBEL was pre-trained with a large external relational triplet extraction dataset, RELA still surpasses REBEL among all datasets. We guess pre-trained REBEL may not be suitable to generate relation names directly. TANL(T5) achieves better results on TACRED than RELA. However, it needs manually designed prompt and utilizes T5 as the backbone network, and the model size is much larger than RoBERTa-large or BART-large. To sum up, RELA achieves comparable results compared with previous excellent methods and can be trained with an academic budget. While being slightly worse than some techniques that are pre-trained with large external datasets or larger models, RELA's simplicity and effectiveness make it an ideal Seq2Seq baseline for relation extraction.

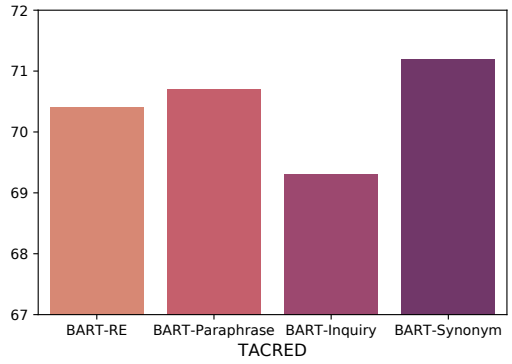

Figure 3: The Effectiveness of different Relation Name Extension Approaches on TACRED. Due to the space limitation, we only report the results on TACRED here. Full results on four datasets can be found in the Appendix A.4

3.5 The Effectiveness of Different Relation Name Extension Approaches

We explore the performances of three different automatic label augmentation approaches in this subsection. Figure 3 shows that: 1) **Synonym** consistently outperforms other approaches and achieves remarkable improvements over

BART−RE. 2) The improvements obtained from **Paraphrase** or **Inquiry** are mixed. **Paraphrase** could provide useful label augmentation information to some extent, while **Inquiry** is not. With detailed augmented information from all approaches,[10] we find that **Synonym** could provide semantically close and correct supplementary information for each relation name. At the same time, **Paraphrase** and **Inquiry** usually generate some irrelevant information, which may not always be helpful for the Seq2Seq-based RE model. Insightfully, we notice that **Paraphrase** and **Inquiry** sometimes imitate associative thinking of human, generating some supplementary information like *'grew up'* from original relation name *'city of birth'* and *'real name'* from original relation name *'person alternate name'*. Unfortunately, this supplementary information may not always useful in the current scenario. To sum up, all augmentation methods could provide some supplementary information, and the **Synonym** is the most stable and effective one.

n	IRE-RoBERTa	BART−RE	RELA
8	60.3	66.7	72.3 (+12.0)
16	69.8	71.1	78.3 (+8.5)
32	73.4	73.8	80.5 (+7.1)
64	82.1	81.7	85.6 (+3.5)

Table 4: Low-resource Setting. Due to the space limitation, we only report results on the Google RE dataset, and similar trends emerge in other datasets. *n* means the sample number for each label. We randomly select training samples ten times and report the mean F1-score on the whole test set to reduce the randomness.

3.6 Low-resource Setting

Since RELA could learn more helpful information from automatic label augmentation, we believe RELA is more suitable for the low-resource setting. To verify our assumption, we conduct low-resource experiments on Google RE. We vary the number of training samples ranging from 8 to 64 for each label, and the comparison results with IRE-RoBERTa are shown in Table 4. We can observe that RELA outperforms IRE-RoBERTa by a large margin in all settings. These results indicate that our method has significant advantages for the low-resource setting. Besides, RELA is especially effective when the training sample size is very tiny, i.e., when each class only has 8 samples, RELA achieves 12.0% F1-score improvements compared with IRE-RoBERTa. We believe the label augmentation methods provide supplementary information for RELA when dealing with low-resource RE.

4 Analysis

In this research, we train our model in a Seq2Seq manner by generating relation names one by one. We are interested in the BART's behavior when dealing with different scenarios, i.e., relation extraction v.s. summarization. We visualize several training details on BART's hidden states and conduct some instructive conclusions. We find that: 1) BART exhibits totally different behaviors when dealing with RE and classical sequence generation tasks (e.g., summarization). Although trained by sequential cross-entropy loss, BART−RE and RELA are more similar to the classification method. 2) RELA could learn more separable relation representations than BART−RE via label augmentation.

4.1 Differences with Classical Sequential Generation Tasks

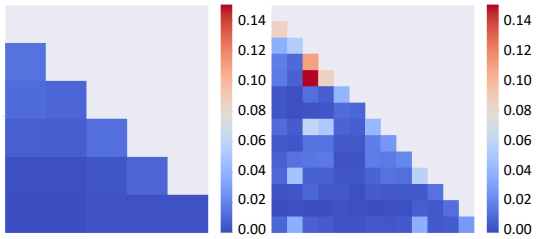

Figure 4: Average decoder self-attention weights among attention heads of different decoding steps (except $\langle bos \rangle$ tag). Left: BART-large for relation extraction; Right: BART-large for text summarization. The self-attention weights of the last decoder layer indicate that BART−RE shows less reliance on previous decoder inputs.

Dependency on Decoder Inputs One of the advantages of sequence generation methods is that previous decoder inputs are visible for the current decoder, thus it is helpful to generate fluent and accurate outputs. But in the case of RE, we find that BART does not pay much attention to previous decoder inputs. Actually, both RE and summarization models spend a high attention score on the $\langle bos \rangle$ tag. In Figure 4 we exclude the attention weight of $\langle bos \rangle$ tag to get a balanced weight distribution. It turns out that the self-attention weights of the BART−RE are much less than those of the summarization model. In another word, for the given relation name *state or province of birth*, when generating the token 'birth', BART−RE almost does not rely on previous decoder outputs. Instead, it is mainly dependent on the $\langle bos \rangle$ tag. This result indicates our previous statement that solving RE with Seq2Seq does not seem like an actual generation process. BART−RE just employs the representation of $\langle bos \rangle$ tag as a probe to generate relation names from encoder states.

The above conclusion also can be drawn from another perspective. We explored the cross-attention weights' distribution among each decoding step. In Figure 5, we employ cross-entropy to measure the similarity of the cross-attention weights' distribution. The results are in line with our expectation. The overall low cross-entropy of BART−RE indicates that the representations obtained from the self-attention mechanism have a high correlation in semantics. These representations do not contain much diverse information for generating relation names.

Similarity of Decoder Hidden States As shown in Figure 6, corresponding to low cross-entropy among cross-attention weights, the last decoder hidden state of BART−RE has a

[10]Reader can find details in Appendix A.4

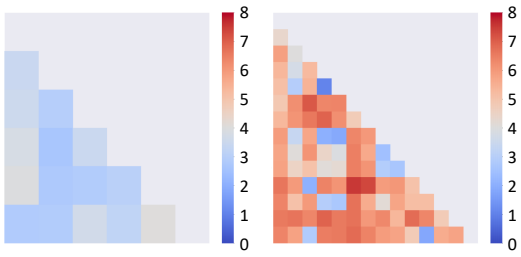

Figure 5: Cross-entropy of the cross-attention weights' distribution among decoding steps. In the BART-RE (Left), low cross-entropy indicates that the distributions of attention weights among different steps contain similar information, while summarization task (Right) has relatively unique cross attention weight distributions.

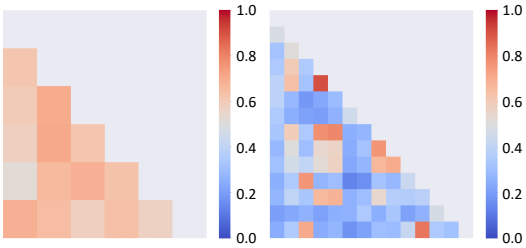

Figure 6: Cosine similarity of the last decoder hidden states between different decoding steps. Left: BART-RE; Right: summarization task. It is complementary to the Figure 5 in the previous subsection. Similar cross-attention weight distributions deduce to similar decoder hidden states.

high score in the case of cosine similarity. Significantly, it is unusual that a preposition "of" and a noun "province" get so close in semantic space. On the other hand, the decoder hidden space is more like a classification space, where the tokens of the same relation belong to the same cluster instead of close in a semantic space.

4.2 Effects on Decoder Hidden States When Enriching Relation Names

In this subsection, we want to explore the changes in the decoder hidden states after enriching relation names. We visualize the last decoding layer of BART-RE and RELA by applying PCA. From Figure 7, we can see that BART-RE could distinguish different relation names in the semantic space. Meanwhile, RELA makes different relations more separable. We believe RELA learns more useful information from label augmentation and finally outputs better decoder hidden states.

5 Related Work

Most previous RE models are classification-based approaches. Early works on RE used machine learning with handcraft features or existing NLP tools, and these method rely on task-specific features (Boschee, Weischedel, and Zamanian 2005; Nguyen and Grishman 2014). Recently, researchers used several deep learning networks for relation extraction, from CNNs and LSTMs (Miwa and Bansal 2016; Guo, Zhang, and Lu 2019; Ye et al. 2019) to Transformer-based approaches (Soares et al. 2019; Li et al. 2020; Xue et al. 2021; Zhou and Chen 2021; Roy and Pan 2021). These methods design novel components based on PLMs or pre-trained language model with external entity-annotated corpus, and achieves impressive results on several relation extraction datasets. Some works have explored Seq2Seq approaches for RE tasks. REBEL (Cabot and Navigli 2021) pre-trained a BART-large model with weakly supervised external datasets, then designed triplets linearization to generate reasonable outputs. REBEL mainly focuses on joint entity and relation extraction. TANL (Paolini et al. 2021) is a seq2seq-based method based on T5. They took structured prediction as a translation task and designed special prompts for different tasks.

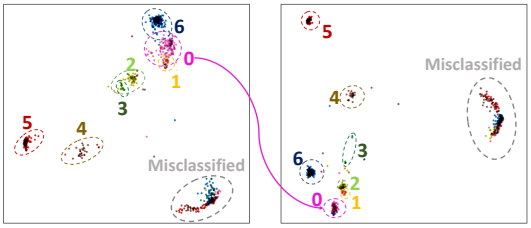

Figure 7: Low dimension representations of average decoder hidden states (Corresponding label names: 0-city of residence, 1-city of death, 2-state or provinces of residence, 3-state or province of birth, 4-organization parent, 5-organization alternate name, 6-employee of). Left: BART-RE; Right: RELA. We find that the representations of different labels are further separated in RELA.

Our research is orthogonal to all the above works. We attempt to solve RE from a sequence-to-sequence perspective by introducing the BART to generate the relation name and its augmented information directly.

6 Conclusion

In this paper, we first explore the advantages of utilizing sequence-to-sequence models for relation extraction task. We find that Seq2Seq architectures are capable of learning semantic information and correlation information from relation names. Then, we propose **R**elation **E**xtraction with **L**abel **A**ugmentation (RELA), a simply yet effective method to extend the relation names for better performances. Finally, we present an in-depth analysis of the BART's behavior when dealing with RE and provide insightful conclusions. We hope that all these contents could encourage the community to make further exploration and breakthrough towards better Seq2Seq-based RE models.

Acknowledgements

This research is supported by the National Key Research And Development Program of China (No. 2021YFC3340101).

References

Beltagy, I.; Lo, K.; and Cohan, A. 2019. SciBERT: A Pretrained Language Model for Scientific Text. In *EMNLP-IJCNLP*.

Boschee, E.; Weischedel, R.; and Zamanian, A. 2005. Automatic information extraction. In *Proceedings of the International Conference on Intelligence Analysis*.

Cabot, P. H.; and Navigli, R. 2021. REBEL: Relation Extraction By End-to-end Language generation. In *EMNLP*.

Cao, N. D.; Izacard, G.; Riedel, S.; and Petroni, F. 2021. Autoregressive Entity Retrieval. In *ICLR*.

Chen, Y.; Zhang, Y.; and Huang, Y. 2022. Learning Reasoning Patterns for Relational Triple Extraction with Mutual Generation of Text and Graph. In *ACL*.

Guo, Z.; Zhang, Y.; and Lu, W. 2019. Attention Guided Graph Convolutional Networks for Relation Extraction. In *ACL*.

Hendrickx, I.; Kim, S. N.; Kozareva, Z.; Nakov, P.; Séaghdha, D. Ó.; Padó, S.; Pennacchiotti, M.; Romano, L.; and Szpakowicz, S. 2010. SemEval-2010 Task 8: Multi-Way Classification of Semantic Relations between Pairs of Nominals. In *SemEval@ACL*.

Josifoski, M.; Cao, N. D.; Peyrard, M.; Petroni, F.; and West, R. 2022. GenIE: Generative Information Extraction. In *NAACL*.

Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; and Zettlemoyer, L. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *ACL*.

Li, B.; Ye, W.; Huang, C.; and Zhang, S. 2021. Multi-view Inference for Relation Extraction with Uncertain Knowledge. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI*.

Li, B.; Ye, W.; Sheng, Z.; Xie, R.; Xi, X.; and Zhang, S. 2020. Graph Enhanced Dual Attention Network for Document-Level Relation Extraction. In *COLING*.

Loshchilov, I.; and Hutter, F. 2019. Decoupled Weight Decay Regularization. In *ICLR*.

Lu, Y.; Liu, Q.; Dai, D.; Xiao, X.; Lin, H.; Han, X.; Sun, L.; and Wu, H. 2022. Unified Structure Generation for Universal Information Extraction. In *ACL*.

Luan, Y.; He, L.; Ostendorf, M.; and Hajishirzi, H. 2018. Multi-Task Identification of Entities, Relations, and Coreference for Scientific Knowledge Graph Construction. In *EMNLP*.

Mao, Y.; Shen, Y.; Yang, J.; Zhu, X.; and Cai, L. 2022. Seq2Path: Generating Sentiment Tuples as Paths of a Tree. In *ACL*.

Miwa, M.; and Bansal, M. 2016. End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures. In *ACL*.

Nguyen, T. H.; and Grishman, R. 2014. Employing Word Representations and Regularization for Domain Adaptation of Relation Extraction. In *ACL*.

Paolini, G.; Athiwaratkun, B.; Krone, J.; Ma, J.; Achille, A.; Anubhai, R.; dos Santos, C. N.; Xiang, B.; and Soatto, S. 2021. Structured Prediction as Translation between Augmented Natural Languages. In *ICLR*.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Köpf, A.; Yang, E. Z.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; and Chintala, S. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *NeurIPS*.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; and Sutskever, I. 2019. Language Models are Unsupervised Multitask Learners. In *OpenAI blog*.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.*

Roy, A.; and Pan, S. 2021. Incorporating medical knowledge in BERT for clinical relation extraction. In *EMNLP*.

Saxena, A.; Kochsiek, A.; and Gemulla, R. 2022. Sequence-to-Sequence Knowledge Graph Completion and Question Answering. In *ACL*.

Soares, L. B.; FitzGerald, N.; Ling, J.; and Kwiatkowski, T. 2019. Matching the Blanks: Distributional Similarity for Relation Learning. In *ACL*.

Sun, T.; Liu, X.; Qiu, X.; and Huang, X. 2021. Paradigm Shift in Natural Language Processing. *CoRR*, abs/2109.12575.

Sun, T.; Liu, X.; Qiu, X.; and Huang, X. 2022. Paradigm Shift in Natural Language Processing. *Int. J. Autom. Comput.*

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to Sequence Learning with Neural Networks. In *NeurIPS*.

Xue, F.; Sun, A.; Zhang, H.; and Chng, E. S. 2021. GDPNet: Refining Latent Multi-View Graph for Relation Extraction. In *AAAI*.

Yamada, I.; Asai, A.; Shindo, H.; Takeda, H.; and Matsumoto, Y. 2020. LUKE: Deep Contextualized Entity Representations with Entity-aware Self-attention. In *EMNLP*.

Yan, H.; Gui, T.; Dai, J.; Guo, Q.; Zhang, Z.; and Qiu, X. 2021. A Unified Generative Framework for Various NER Subtasks. In *ACL/IJCNLP*.

Ye, W.; Li, B.; Xie, R.; Sheng, Z.; Chen, L.; and Zhang, S. 2019. Exploiting Entity BIO Tag Embeddings and Multi-task Learning for Relation Extraction with Imbalanced Data. In *ACL*.

Zhang, S.; Ng, P.; Wang, Z.; and Xiang, B. 2022a. REKnow: Enhanced Knowledge for Joint Entity and Relation Extraction. *CoRR*, abs/2206.05123.

Zhang, S.; Shen, Y.; Tan, Z.; Wu, Y.; and Lu, W. 2022b. De-Bias for Generative Extraction in Unified NER Task. In *ACL*.

Zhang, Y.; Zhong, V.; Chen, D.; Angeli, G.; and Manning, C. D. 2017. Position-aware Attention and Supervised Data Improve Slot Filling. In *EMNLP*.

Zhou, W.; and Chen, M. 2021. An Improved Baseline for Sentence-level Relation Extraction. *CoRR*, abs/2102.01373.