GenRE: A Generative Model for Relation Extraction

Anonymous ACL submission

Abstract

Relation extraction (RE) is an important information extraction task which provides essential information to many NLP applications such as knowledge base population and question answering. In this paper, we present a novel generative model for relation extraction (which we call GenRE), where RE is modeled as a sequence-to-sequence generation task. We explore various encoding schemes for the source and target sequences, and design effective schemes that enable GenRE to achieve state-of-the-art performance on three benchmark RE datasets. In addition, we introduce negative sampling and decoding scaling techniques which provide a flexible tool to tune the precision and recall performance of our GenRE model. Our approach can be extended to extract all relation triples from a sentence in one pass. Although the one-pass approach incurs certain performance loss, it is much more computationally efficient.

1 Introduction

Relation extraction (RE) is a fundamental information extraction task that seeks to detect and characterize semantic relationships between pairs of entities or events from natural language text. It provides important information for many NLP applications such as knowledge base population (Ji and Grishman, 2011) and question answering (Xu et al., 2016).

Relation extraction has been studied in two settings. In the first setting, gold entities are provided, and the RE task (also known as *relation classification* or *slot filling*) is to classify the relationships between given pairs of entities in sentences (Hendrickx et al., 2010; Zhang et al., 2017b).

In the second setting, no gold entities are provided, and one needs to consider both entity recognition and relation extraction (Doddington et al., 2004). This can be tackled via a *pipeline* approach: first an entity recognition model is applied to extract entities, and then a relation extraction model is applied to classify the relationships between all pairs of predicted entities (Kambhatla, 2004; Zhou et al., 2005; Chan and Roth, 2011; Zhong and Chen, 2021). Alternatively, this can be addressed by a *joint* approach where entity recognition and relation extraction are modeled and solved jointly (Li and Ji, 2014; Miwa and Bansal, 2016; Luan et al., 2019; Li et al., 2019; Wadden et al., 2019; Lin et al., 2020; Wang and Lu, 2020).

Given a sentence with a pair of gold or predicted entities, RE is naturally formulated as a classification task. It is a very challenging task since relation extraction relies heavily on both syntactic and semantic information, with possibly multiple entities and relations existing in one sentence. Many RE models have been developed to improve the performance on popular benchmark RE datasets such as ACE05 (Walker et al., 2006), SemEval 2010 Task 8 (Hendrickx et al., 2010) and TACRED (Zhang et al., 2017b). RE models have evolved from feature-based statistical models (Kambhatla, 2004; Zhou et al., 2005; Chan and Roth, 2011; Li and Ji, 2014), to neural network models that use pretrained word embeddings/language models (Zeng et al., 2014; dos Santos et al., 2015; Miwa and Bansal, 2016; Zhang et al., 2017a; Wu and He, 2019; Baldini Soares et al., 2019; Zhong and Chen, 2021).

In this paper, we present a novel generative model for relation extraction (named GenRE), which treats RE as a sequence-to-sequence (seq2seq) generation task. Given an input sentence and a pair of entities in it, the GenRE model generates an output *relation triple* which consists of the two entities and a relation type that specifies their relationship. Compared with classification based RE approaches, the generative approach has the capability of encoding entity information in the target sequence. Experiment results show that our

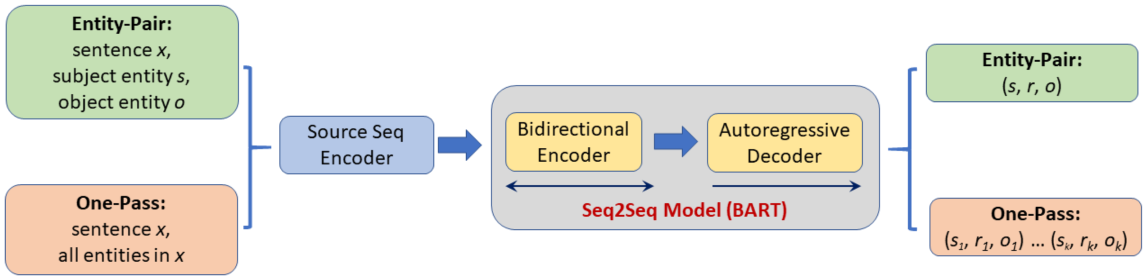

Figure 1: Overview of the Generative Relation Extraction (GenRE) model.

GenRE model outperforms previous RE models on three benchmark RE datasets. Moreover, our approach enables the adoption of standard fine-tuning procedures with pre-trained seq2seq language models (Wolf et al., 2020), without the need of designing ad-hoc architectures, hence facilitating the deployment in information extraction systems.

While the idea of using seq2seq models for RE was studied before (Zeng et al., 2018, 2020; Nayak and Ng, 2020; Zhang et al., 2020), the previous works focused on end-to-end relation extraction that jointly extracts entities and relations from sentences. In this paper we focus on relation extraction with entities (gold or predicted) provided, and we show that it is important to encode entity information both in the source and in the target sequences to achieve the best performance.

We summarize our main contributions as follows:

- We have explored various encoding representations for the source and target sequences, and designed effective schemes that enable our GenRE model to achieve state-of-the-art performance on three popular benchmark RE datasets: ACE05, SemEval 2010 Task 8 and TACRED.

- We have shown that negative sampling during training improves the recall performance of the GenRE model. We have also developed a novel decoding scaling scheme during inference to improve the precision performance. These together provide a flexible tool to tune both the precision and the recall performance of our GenRE model.

- Our approach can be extended to extract all relation triples from a sentence in one pass. Although the one-pass approach incurs certain performance loss, it greatly reduces the training and decoding time, as we show in Section 3.5.

2 Generative Relation Extraction

For given pairs or all pairs of entities in a sentence, the RE task is to detect and characterize the relationships between those pairs of entities (Doddington et al., 2004; Hendrickx et al., 2010; Zhang et al., 2017b).

When entities (gold or predicted) are provided, we consider two approaches of encoding the entity information. With the *entity-pair* approach, each time we encode one pair of entities in the source sequence; while with the *one-pass* approach, we encoded all the entities in the source sequence.

In Figure 1 we show the overview of our Generative Relation Extraction (GenRE) model. First, an input sentence and i) a pair of entities (under the entity-pair approach) or ii) all the entities (under the one-pass approach) in the sentence are encoded to a source sequence (src_seq) via a source encoding module. Then, the source sequence is passed to a seq2seq model. In this paper, we use BART (Lewis et al., 2020a) as the seq2seq model. BART uses the standard seq2seq Transformer architecture (Vaswani et al., 2017) with multiple bidirectional encoder layers and autoregressive decoder layers. BART was pre-trained as a denoising autoencoder that maps a corrupted document to the original document. Since BART has an autoregressive decoder, it can be fine-tuned for sequence generation tasks. For our GenRE model, we use BART to generate either i) a relation triple that consists of the subject entity, the object entity, and the relation type between the two entities (under the entity-pair approach) or ii) all the relation triples in the sentence (under the one-pass approach).

2.1 Entity-Pair Approach

Under the entity-pair approach, each time we encode the entity information of one pair of entities of an input sentence in the source sequence.

Let $\mathbf{x} = (x_1, x_2, ..., x_n)$ be an input sentence with n tokens. Let $\mathbf{s} = (x_{s_1}, ..., x_{s_2})$ and $\mathbf{o} = (x_{o_1}, ..., x_{o_2})$ be a pair of entities in the input sentence $\mathbf{x}$, where $\mathbf{s}$ is the subject (head) entity and $\mathbf{o}$ is the object (tail) entity, with entity types T_s and T_o, respectively.

Let $R = \{r_1, r_2, ..., r_K\}$ be the set of predefined relation types. We use a null relation type r_0 (e.g., r_0="None") to indicate that the two entities under consideration do not have a relationship belonging to one of the K relation types.

We have explored various schemes to encode the entity information in the source sequence. The first scheme is to use some special tokens to mark the start and end of the entities to encode entity location information as in (Wu and He, 2019; Baldini Soares et al., 2019):

$$f_1(\mathbf{x}, \mathbf{s}, \mathbf{o}) = (x_1, ..., \$, x_{s_1}, ..., x_{s_2}, \$, ..., \&, x_{o_1}, ..., x_{o_2}, \&, ..., x_n) \quad (1)$$

The second scheme is to use the entity type of an entity to mark the start and end of that entity, in order to encode both the entity location and entity type information in the source sequence as in (Ni et al., 2020; Zhong and Chen, 2021):

$$f_1(\mathbf{x}, \mathbf{s}, \mathbf{o}) = (x_1, ..., T_s, x_{s_1}, ..., x_{s_2}, T_s, ..., T_o, x_{o_1}, ..., x_{o_2}, T_o, ..., x_n) \quad (2)$$

In Equations (1) and (2) we assume that $\mathbf{s}$ appears before $\mathbf{o}$ (i.e., $1 \le s_1 \le s_2 < o_1 \le o_2 \le n$). If $\mathbf{s}$ appears after $\mathbf{o}$, the positions of the two entities will be switched.

Next, to model the *direction* of a relation (i.e., to encode which of the two entities is the subject entity and which is the object entity), we create the following sub-sequence:

$$f_2(\mathbf{s}, \mathbf{o}) = [\mathbf{s} \,\#\, T_s \,,\, \mathbf{o} \,\#\, T_o] \quad (3)$$

so that the subject entity always appears before the object entity in this sub-sequence.

In addition to sentence entity encoding f_1 and subject-object encoding f_2, we find that adding the list of relation types is helpful to the GenRE model:

$$f_3(R) = [r_1 - r_2 - \cdots - r_K] \quad (4)$$

The final encoding of the source sequence (i.e., the input to the generative model) is the concatenation of the three sub-sequences:

$$src\_seq = f_1(\mathbf{x}, \mathbf{s}, \mathbf{o}) \oplus f_2(\mathbf{s}, \mathbf{o}) \oplus f_3(R) \quad (5)$$

For the target sequences, we have also explored various choices. First we find that adding the subject and object entities to the target sequence (i.e., generating a relation triple) is better than generating the relation type only. Among the different orders of the relation triple that we have tried, we find that generating the relation triple with the order "subject, relation, object" is the most effective:

$$tgt\_seq = [\mathbf{s} \mid r(\mathbf{s}, \mathbf{o}) \mid \mathbf{o}] \quad (6)$$

where $r(\mathbf{s}, \mathbf{o})$ is the relation type that specifies the relationship between the subject entity $\mathbf{s}$ and the object entity $\mathbf{o}$[1]. If $r(\mathbf{s}, \mathbf{o}) = r_0$ (null relation type), the triple is a *negative* example; otherwise, the triple is a *positive* example.

2.1.1 Improving Recall via Sampling Negative Training Examples

It can be very challenging to achieve a sufficient *recall* for RE models (Zhang et al., 2017b). Let R_{gold} be the set of gold positive relation triples in an RE dataset. Let R_{pred} be the set of predicted positive relation triples of an RE model when applied on the dataset. The *precision* and *recall* of the RE model on the dataset are defined as:

$$precision = \frac{|R_{pred} \cap R_{gold}|}{|R_{pred}|}$$

$$recall = \frac{|R_{pred} \cap R_{gold}|}{|R_{gold}|}$$

where $|A|$ is the size (cardinality) of set A.

To improve the recall, one can try to let the model predict more positive relation triples to increase the number of true positives (a *true positive* is a predicted positive relation triple that matches a gold positive relation triple, i.e., a triple in $|R_{pred} \cap R_{gold}|$).

We find that sampling negative training examples during training is very effective for improving

[1]The model is trained to generate output that contains the special characters '[', ']', and '|'.

the recall of the GenRE model. Specifically, we keep all the positive training examples while randomly sampling a fraction α of the total negative training examples for training the GenRE model. α is called the *negative sampling ratio* which is a number between 0 and 1. As we decrease α, the GenRE model will be trained with fewer negative examples and higher positive-to-negative ratio, and it would generate more positive relation triples during inference, hence improving the recall. We observe that sampling negative training examples, however, might reduce the precision. In the next subsection we present a scheme to improve the precision.

2.1.2 Improving Precision via Decoding Scaling

When the GenRE model generates more positive relation triples and gets more true positives, the recall can be improved. However, this may also increase the number of false positives and reduce the precision. We propose a novel *decoding scaling* scheme that utilizes the sequence scores of the top N generated target sequences to improve the precision.

For an input source sequence $\mathbf{z}$, we let the GenRE model generate top N target sequences (relation triples) $\mathbf{y}_1, ..., \mathbf{y}_N$ with the highest sequence scores, where the *sequence score* of $\mathbf{y}_i$ is the conditional probability of $\mathbf{y}_i$ given $\mathbf{z}$: $P(\mathbf{y}_i|\mathbf{z})$. Note that in normal decoding, we just let the GenRE model generate the best target sequence $\mathbf{y}_1$ and use that as the prediction.

If a relation triple includes a non-null relation type in R, we call it a positive triple; otherwise we call it a negative triple. There are two cases to consider:

(1) If the top N triples are all positive or all negative, the scheme simply selects the top positive or negative triple with the highest sequence score. This is the same as in normal decoding.

(2) If the top N triples include both positive and negative triples, let $\mathbf{y}^*_+$ and $\mathbf{y}^*_-$ be the best positive and negative triple, respectively. We select the triple $\mathbf{y}^*$ as the prediction as follows:

$$\mathbf{y}^* = \begin{cases} \mathbf{y}^*_+, & \text{if } \dfrac{P(\mathbf{y}^*_+|\mathbf{z})}{P(\mathbf{y}^*_-|\mathbf{z})} \geq \beta \\ \mathbf{y}^*_-, & \text{otherwise} \end{cases} \tag{7}$$

β is called the *decoding scaling factor*. When $\beta =$ 1 (no scaling), the scheme will just select the best generated triple as in normal decoding. When $\beta >$ 1, the scheme will select the best positive triple $\mathbf{y}^*_+$ only if its score is greater than the score of the best negative triple $\mathbf{y}^*_-$ by a margin, so the predicted positive triple is more likely to be a true positive. Therefore, the total number of false positives will be reduced, hence improving the precision.

2.2 One-Pass Approach

Under the one-pass approach, each time we encode the information of all the entities of the input sentence $\mathbf{x}$ in the source sequence. The target sequence also includes all the positive relation triples in the input sentence.

Let $E(\mathbf{x})$ be the set of all entities in $\mathbf{x}$, where an entity $\mathbf{e}_i = (x_{i_1}, ..., x_{i_2}) \in E(\mathbf{x})$ is a span in $\mathbf{x}$, with entity type T_i. Let $R(\mathbf{x})$ be the set of all positive relation triples in $\mathbf{x}$, where a relation triple $\mathbf{t}_j = (\mathbf{s}_j, r_j, \mathbf{o}_j) \in R(\mathbf{x})$ consists of a subject entity $\mathbf{s}_j \in E(\mathbf{x})$, an object entity $\mathbf{o}_j \in E(\mathbf{x})$, and their relation type $r_j \in R$.

First we extend the entity type marking to all the entities in the input sentence as follows:

$$\begin{aligned} & f_1(\mathbf{x}, E(\mathbf{x})) \\ = & \Big(x_1, ..., T_i, x_{i_1}, ..., x_{i_2}, T_i, ..., x_n, \forall \mathbf{e}_i \in E(\mathbf{x})\Big) \end{aligned} \tag{8}$$

Then we encode the list of entities with their entity types as follows:

$$f_2(E(\mathbf{x})) = [\mathbf{e}_i \;\#\; T_i \;, \forall \mathbf{e}_i \in E(\mathbf{x})] \tag{9}$$

We also include the list of relation types as in (4). The final encoding of the source sequence is:

$$src\_seq = f_1(\mathbf{x}, E(\mathbf{x})) \oplus f_2(E(\mathbf{x})) \oplus f_3(R) \tag{10}$$

The encoding of the target sequence is:

$$tgt\_seq = \oplus_{(\mathbf{s}_j, r_j, \mathbf{o}_j) \in R(\mathbf{x})} [\mathbf{s}_j \mid r_j \mid \mathbf{o}_j] \tag{11}$$

In case there is no positive relation triple in $\mathbf{x}$ (i.e., $R(\mathbf{x}) = \emptyset$), we set tgt_seq to be "[None | None | None]".

If no entity information is provided, then the encoding of the source sequence is the concatenation of the input sentence and the list of relation types:

$$src\_seq = \mathbf{x} \oplus f_3(R) \tag{12}$$

The encoding of the target sequence is the same as in (11).

Note that for an input sentence with m entities, the entity-pair approach will create $m(m-1)$ src_seq and tgt_seq pairs, while the one-pass approach will create just one src_seq and tgt_seq pair, which is more computationally efficient.

3 Experiments

3.1 Datasets

We evaluate our GenRE model on three popular benchmark RE datasets: ACE05, SemEval 2010 Task 8, and TACRED.

The ACE05 dataset (Walker et al., 2006) is a benchmark RE dataset developed by the Linguistic Data Consortium (LDC) for the purpose of Automatic Content Extraction (ACE) technology evaluation. ACE05 defines 7 entity types and 6 relation types between the entities. We use the same training, development, and test data split in prior works (Li and Ji, 2014; Miwa and Bansal, 2016; Luan et al., 2019; Zhong and Chen, 2021).

The SemEval 2010 Task 8 dataset (Hendrickx et al., 2010) is a benchmark dataset for evaluating classification of semantic relations between pairs of nominals. It defines 9 relation types and one null relation type "Other". The dataset includes 8000 training examples and 2717 test examples. We randomly select 1000 examples from the training set for development and keep the remaining 7000 for training.

TACRED (Zhang et al., 2017b) is a large supervised RE dataset obtained via crowdsourcing and targeted for TAC KBP relations. It defines 42 relation types (including a null relation type "no_relation") and includes over 100K examples. The dataset was recently revised and improved in (Alt et al., 2020) by reducing the annotation errors. In our experiments we use this revised version, which includes 68,124 training examples, 22,631 development examples, and 15,509 test examples.

3.2 Implementation Details

We use HuggingFace's pytorch implementation of transformers (Wolf et al., 2020). We build the GenRE models on top of the BART-Large model (Lewis et al., 2020a), which has $L = 24$ transformer layers (12 encoder and decoder layers each), with hidden state vector size $H = 1024$, number of attention heads $A = 16$, and 406M parameters.

We use the development sets to tune the hyperparameters. We learn the model parameters using Adam (Kingma and Ba, 2015), with a learning rate $l = 3e\text{-}5$, a training batch size of $b = 16$ for ACE05 and SemEval 2010 Task 8, and $b = 8$ for TACRED. We train the GenRE models for 10 epochs with the entity-pair approach and 20 epochs with the one-pass approach. All experiments were conducted on a 2 NVIDIA V100 GPUs computer.

3.3 Main Results

Our best GenRE model with the entity-pair approach uses source sequence encoding (5) with entity type markers (2) and target sequence encoding (6) with the order "subject, relation, object". The results of our GenRE model reported in Tables 1-3 include the *mean* and *standard deviation* of the performance over 5 runs with different random seeds.

In Table 1 we compare our GenRE model with previous approaches on the ACE05 test set. As in prior works we use micro-averaged F_1 score as the evaluation metric. For entity recognition, a predicted entity is considered correct if its predicted entity span and entity type are both correct. For relation extraction with predicted entities, following (Li and Ji, 2014; Wang and Lu, 2020; Zhong and Chen, 2021), we use two evaluation metrics: 1) *Rel*: a predicted relation is considered correct if the two predicted entity spans and the predicted relation type are correct; 2) *Rel+*: a predicted relation is considered correct if the two predicted entity spans and entity types as well as the predicted relation type are all correct. Our entity recognition model is an ensemble of RoBERTa (Liu et al., 2019) based sequence labeling models with voting.

As shown in Table 1, our GenRE model achieves the state-of-the-art performance on ACE05. GenRE improves the previous best approach PURE by 0.8 F_1 point on the Rel metric and by 1.2 F_1 point on the Rel+ metric, without using any cross-sentence information.

In Table 2 we compare our GenRE model with previous approaches on the SemEval 2010 Task 8 test set. As in prior works we use gold entities and the SemEval 2010 Task 8 official scoring metric which is macro-averaged F_1 score for the 9 relation types (excluding the null relation type "Other") and takes directionality into account. Our GenRE model achieves the state-of-art performance. While the best model (BERT$_{EM}$+MTB) in (Baldini Soares et al., 2019) used 600 million relation statement pairs derived from English Wikipeida to pre-train the model, GenRE achieves better performance without using any additional

Model	Entity	Rel	Rel+
(Li and Ji, 2014)	80.8	52.1	49.5
SPTree (Miwa and Bansal, 2016)	83.4	-	55.6
(Katiyar and Cardie, 2017)	82.6	55.9	53.6
(Zhang et al., 2017a)	83.6	-	57.5
MRT (Sun et al., 2018)	83.6	-	59.6
(Li et al., 2019)	84.8	-	60.2
(Dixit and Al-Onaizan, 2019)	86.0	-	62.8
DYGIE (Luan et al., 2019)*	88.4	63.2	-
DyGIE++ (Wadden et al., 2019)*	88.6	63.4	-
(Lin et al., 2020)	88.8	67.5	-
(Wang and Lu, 2020)	89.5	67.6	64.3
PURE - single sentence (Zhong and Chen, 2021)	89.7	69.0	65.6
PURE - cross sentence (Zhong and Chen, 2021)*	90.9	69.4	67.0
GenRE (ours)	90.4	**70.2** $\pm$ 0.4	**68.2** $\pm$ 0.5

Table 1: Micro F_1 scores on the ACE05 test set. For GenRE we report the *mean* and *standard deviation* of the performance over 5 runs. *These models use cross-sentence information.

Model	Macro $\mathbf{F_1}$
CNN (Zeng et al., 2014)	82.7
Attention Bi-LSTM (Zhou et al., 2016)	84.0
CR-CNN (dos Santos et al., 2015)	84.1
Bi-LSTM (Zhang et al., 2015)	84.3
Hierarchical Attention RNN (Xiao and Liu, 2016)	84.3
Entity Attention Bi-LSTM (Lee et al., 2019)	85.2
Attention CNN (Shen and Huang, 2016)	85.9
TRE (Alt et al., 2019)	87.1
SpanRel (Jiang et al., 2020)	87.4
Multi-Attention CNN (Wang et al., 2016)	88.0
KnowBERT-W+W (Peters et al., 2019)*	89.1
R-BERT (Wu and He, 2019)	89.25
BERT_{EM} (Baldini Soares et al., 2019)	89.2
BERT_{EM}+MTB (Baldini Soares et al., 2019)*	89.5
GenRE (ours)	**89.7** $\pm$ 0.3

Table 2: Macro F_1 scores on the SemEval 2010 Task 8 test set. For GenRE we report the *mean* and *standard deviation* of the performance over 5 runs. *These models use additional data derived from Wikipedia/WordNet to pre-train their models.

Model	Micro $\mathbf{F_1}$
LSTM (masked) (Zhang et al., 2017b)	63.9
LSTM + BERT (masked) (Alt et al., 2020)	73.4
CNN (masked) (Nguyen and Grishman, 2015)	66.5
CNN + BERT (masked) (Alt et al., 2020)	74.3
TRE (Alt et al., 2019)	75.3
SpanBERT (Joshi et al., 2020)	78.0
KnowBERT-W+W (Peters et al., 2019)*	79.3
GenRE (ours)	**80.6** $\pm$ 0.6

Table 3: Micro F_1 scores on the TACRED-Revised test set. For GenRE we report the *mean* and *standard deviation* of the performance over 5 runs. *This model was pre-trained with additional data derived from Wikipedia and WordNet.

data.

In Table 3 we compare our GenRE model with previous approaches on the revised TACRED test set. As in prior works we use gold entities and micro-averaged F_1 score as the evaluation metric. Again, our GenRE model achieves the state-of-the-art performance without using any additional data.

3.4 Ablation Studies

In this subsection we study the contributions of different components on the GenRE model.

3.4.1 Source and Target Sequence Encoding

In Table 4 we show the performance of the GenRE model on the ACE05 development set under different source and target sequence encoding schemes. There are two observations:

- For the source sequence encoding, it is important to encode the entity information in the input sentence using entity markers. The special token markers (1) that encode the entity location information improved the performance by 1.1 F_1 points, and the entity type markers (2) that encode both the entity location and type information improved the performance by 5.7 F_1 points, compared with not using any entity markers.

- For the target sequence encoding, it is beneficial to add the subject and object entities to the target sequence, which helps the GenRE model to generate more accurate relation types. This improved the performance by 4+ F_1 points compared with encoding the relation type only ($[r]$). Among the different orderings of the relation triple that we have tried, the order "subject, relation, object" ($[s|r|o]$) achieved the best performance.

3.4.2 Negative Sampling and Decoding Scaling

In Table 5 we show the performance of the GenRE model on the ACE05 development set under different negative sampling ratio α and decoding scaling factor β (we let the GenRE model generate top $N = 5$ target sequences). The key observations are:

- For a fixed decoding scaling factor β (a column in Table 5), as we decrease the negative sampling ratio α (i.e., keep fewer negative training examples during training), the recall is improved. The recall reached the highest value at $\alpha = 0.8$, and more aggressive negative sampling could reduce the recall.

- For a fixed negative sampling ratio α (a row in Table 5), as we increase the decoding scaling factor β (so the predicted positive triple is more likely to be a true positive triple), the precision is improved. However, increasing β hurts the recall.

- Negative sampling and decoding scaling provide a flexible tool to tune the precision and recall performance of our GenRE model. If we want to achieve a high recall, we would keep $\beta = 1$ (no decoding scaling) and select an optimal α: in this case $\alpha = 0.8$ gives the best recall performance of 79.3. On the other hand, if we want to have a higher precision, we would keep $\alpha = 1$ (no negative sampling) and pick a larger β. We can also use the development set to find the optimal α and β that achieve the highest F_1 score.

Source	Target	P	R	F_1
no entity marker	$[s\|r\|o]$	70.2	68.8	69.5
special token marker	$[s\|r\|o]$	73.2	68.2	70.6
entity type marker	$[\mathbf{s}\|\mathbf{r}\|\mathbf{o}]$	**74.4**	**76.1**	**75.2**
entity type marker	$[r]$	71.2	66.9	69.0
entity type marker	$[r\|s\|o]$	71.8	75.3	73.5
entity type marker	$[s\|o\|r]$	73.7	72.8	73.3

Table 4: Performance (precision, recall, F_1 score) of the GenRE model on the ACE05 development set (with gold entities) under different source and target sequence encoding schemes.

3.5 Entity-Pair vs. One-Pass Approach

A sentence can have multiple entities and relation triples in the ACE05 data, so we use ACE05 to compare the performance and computational cost of our GenRE model under the entity-pair approach and the one-pass approach.

As shown in Table 6, the entity-pair approach has a clear advantage over the one-pass approach on performance (nearly 10 F_1 points gain). On the other hand, since the one-pass approach creates just one source sequence for a sentence and extracts all the relation triples from the sentence in one-pass, it has a smaller number of training/test examples and hence lower computational cost (6x faster for training and 15x faster for decoding) compared with the

	$\beta = 1$			$\beta = 1.1$			$\beta = 1.2$			$\beta = 1.3$		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
$\alpha = 1$	72.1	75.0	73.5	74.2	73.0	73.6	76.2	71.4	73.8	**77.5**	69.0	73.0
$\alpha = 0.9$	71.6	78.7	75.0	73.7	75.9	74.7	76.6	73.3	74.9	**77.5**	70.4	73.8
$\alpha = 0.8$	68.2	**79.3**	73.4	71.7	78.2	74.8	74.4	76.1	**75.2**	76.4	73.3	74.8
$\alpha = 0.7$	66.6	78.7	72.2	69.9	76.9	73.2	72.1	74.6	73.3	74.5	72.5	73.5

Table 5: Performance of the GenRE model on the ACE05 development set (with gold entities) under different negative sampling ratio α and decoding scaling factor β.

Setup		**Performance**			**Computational Cost**	
Approach	Entities	P	R	F_1	Training	Decoding
entity-pair	*gold*	74.4	76.1	75.2	559 mins	459 secs
entity-pair	*predicted*	66.7	68.3	67.5	559 mins	456 secs
one-pass	*gold*	66.0	65.8	65.9	90 mins	32 secs
one-pass	*predicted*	63.4	56.5	59.7	90 mins	28 secs
one-pass	*no*	55.8	51.9	53.8	80 mins	29 secs

Table 6: Performance and computational cost of the GenRE model under the entity-pair approach and the one-pass approach on the ACE05 development set. Training and decoding time is measured on a 2 NVIDIA V100 GPUs computer with a batch size of 16, 10 training epochs for entity-pair and 20 training epochs for one-pass. *gold*: the gold entities are given during inference. *predicted*: the predicted entities are given during inference. *no*: no entities are given during inference (and training).

entity-pair approach. Another key observation is that adding entity information (even predicted) can significantly improve the performance compared with no entities provided.

4 Related Work

Many RE models have been previously developed to improve the performance on benchmark RE datasets such as ACE05, SemEval 2010 Task 8 and TACRED. Earlier RE models require extensive feature engineering to derive and combine various lexical, syntactic and semantic features (Kambhatla, 2004; Zhou et al., 2005; Chan and Roth, 2011; Li and Ji, 2014). Later neural network based RE models have become dominant, including CNN based models (Zeng et al., 2014; dos Santos et al., 2015; Nguyen and Grishman, 2015), RNN based models (Zhang et al., 2015; Xiao and Liu, 2016; Miwa and Bansal, 2016), and most recently transformer based models (Wu and He, 2019; Baldini Soares et al., 2019; Zhong and Chen, 2021).

Seq2seq models have long been used for NLP tasks such as machine translation (Sutskever et al., 2014; Cho et al., 2014) and text summarization (Rush et al., 2015; Chopra et al., 2016). Recently, generative approaches based on seq2seq models have been proven competitive in NLP applications such as question answering, fact checking, relation linking and intent classification (Lewis et al., 2020b; Petroni et al., 2021; Rossiello et al., 2021; Ahmad et al., 2021). While seq2seq models were also applied to RE in (Zeng et al., 2018, 2020; Nayak and Ng, 2020; Zhang et al., 2020), the previous works focused on end-to-end relation extraction that jointly extracts entities and relations from sentences. Our work focuses on relation extraction with entities (gold or predicted) provided, and we show that it is important to encode entity information both in the source and in the target sequences.

5 Conclusion

In this paper we presented a novel generative model for relation extraction (GenRE). We showed the importance of encoding entity information in the source and target sequences and designed effective encoding schemes. We also introduced negative sampling and decoding scaling techniques which provide a flexible tool to tune the precision and recall performance of the model. Our GenRE model achieves state-of-the-art performance on three popular benchmark RE datasets with a consistent, extensible, and successful approach. Our approach can be extended to extract all relation triples from a sentence in one-pass which is much more computationally efficient.

References

Wasi Ahmad, Jianfeng Chi, Tu Le, Thomas Norton, Yuan Tian, and Kai-Wei Chang. 2021. Intent classification and slot filling for privacy policies. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4402–4417, Online. Association for Computational Linguistics.

Christoph Alt, Aleksandra Gabryszak, and Leonhard Hennig. 2020. TACRED revisited: A thorough evaluation of the TACRED relation extraction task. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1558–1569, Online. Association for Computational Linguistics.

Christoph Alt, Marc Hübner, and Leonhard Hennig. 2019. Improving relation extraction by pre-trained language representations. In *1st Conference on Automated Knowledge Base Construction, AKBC 2019, Amherst, MA, USA, May 20-22, 2019*.

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2895–2905, Florence, Italy. Association for Computational Linguistics.

Yee Seng Chan and Dan Roth. 2011. Exploiting syntactico-semantic structures for relation extraction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 551–560, Portland, Oregon, USA. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Sumit Chopra, Michael Auli, and Alexander M. Rush. 2016. Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 93–98, San Diego, California. Association for Computational Linguistics.

Kalpit Dixit and Yaser Al-Onaizan. 2019. Span-level model for relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5308–5314, Florence, Italy. Association for Computational Linguistics.

George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. 2004. The automatic content extraction (ACE) program – tasks, data, and evaluation. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal. European Language Resources Association (ELRA).

Cícero dos Santos, Bing Xiang, and Bowen Zhou. 2015. Classifying relations by ranking with convolutional neural networks. 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 626–634, Beijing, China. Association for Computational Linguistics.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2010. SemEval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 33–38, Uppsala, Sweden. Association for Computational Linguistics.

Heng Ji and Ralph Grishman. 2011. Knowledge base population: Successful approaches and challenges. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1148–1158, Portland, Oregon, USA. Association for Computational Linguistics.

Zhengbao Jiang, Wei Xu, Jun Araki, and Graham Neubig. 2020. Generalizing natural language analysis through span-relation representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2120–2133, Online. Association for Computational Linguistics.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Nanda Kambhatla. 2004. Combining lexical, syntactic, and semantic features with maximum entropy models for information extraction. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 178–181, Barcelona, Spain. Association for Computational Linguistics.

Arzoo Katiyar and Claire Cardie. 2017. Going out on a limb: Joint extraction of entity mentions and relations without dependency trees. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 917–928, Vancouver, Canada. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings* *of the 3rd International Conference on Learning Representations (ICLR)*, ICLR '15.

Joohong Lee, Sangwoo Seo, and Yong Suk Choi. 2019. Semantic relation classification via bidirectional LSTM networks with entity-aware attention using latent entity typing. *CoRR*, abs/1901.08163.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020a. 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.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020b. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Qi Li and Heng Ji. 2014. Incremental joint extraction of entity mentions and relations. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 402–412, Baltimore, Maryland. Association for Computational Linguistics.

Xiaoya Li, Fan Yin, Zijun Sun, Xiayu Li, Arianna Yuan, Duo Chai, Mingxin Zhou, and Jiwei Li. 2019. Entity-relation extraction as multi-turn question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1340–1350, Florence, Italy. Association for Computational Linguistics.

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. A joint neural model for information extraction with global features. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7999–8009, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Yi Luan, Dave Wadden, Luheng He, Amy Shah, Mari Ostendorf, and Hannaneh Hajishirzi. 2019. A general framework for information extraction using dynamic span graphs. 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 3036–3046, Minneapolis, Minnesota. Association for Computational Linguistics.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using LSTMs on sequences and tree structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1105–1116, Berlin, Germany. Association for Computational Linguistics.

Tapas Nayak and Hwee Tou Ng. 2020. Effective modeling of encoder-decoder architecture for joint entity and relation extraction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8528–8535.

Thien Huu Nguyen and Ralph Grishman. 2015. Relation extraction: Perspective from convolutional neural networks. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 39–48, Denver, Colorado. Association for Computational Linguistics.

Jian Ni, Taesun Moon, Parul Awasthy, and Radu Florian. 2020. Cross-lingual relation extraction with transformers. *CoRR*, abs/2010.08652.

Matthew E. Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. Knowledge enhanced contextual word representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 43–54, Hong Kong, China. Association for Computational Linguistics.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 2021. KILT: a benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2523–2544, Online. Association for Computational Linguistics.

Gaetano Rossiello, Nandana Mihindukulasooriya, Ibrahim Abdelaziz, Mihaela A. Bornea, Alfio Gliozzo, Tahira Naseem, and Pavan Kapanipathi. 2021. Generative relation linking for question answering over knowledge bases. In *The Semantic Web - ISWC 2021 - 20th International Semantic Web Conference, ISWC 2021, Virtual Event, October 24-28, 2021, Proceedings*, volume 12922 of *Lecture Notes in Computer Science*, pages 321–337. Springer.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, Lisbon, Portugal. Association for Computational Linguistics.

Yatian Shen and Xuanjing Huang. 2016. Attention-based convolutional neural network for semantic relation extraction. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2526–2536, Osaka, Japan. The COLING 2016 Organizing Committee.

Changzhi Sun, Yuanbin Wu, Man Lan, Shiliang Sun, Wenting Wang, Kuang-Chih Lee, and Kewen Wu. 2018. Extracting entities and relations with joint minimum risk training. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2256–2265, Brussels, Belgium. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 3104–3112, Cambridge, MA, USA. MIT Press.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*, pages 6000–6010.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, relation, and event extraction with contextualized span representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5784–5789, Hong Kong, China. Association for Computational Linguistics.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. ACE 2005 multilingual training corpus.

Jue Wang and Wei Lu. 2020. Two are better than one: Joint entity and relation extraction with table-sequence encoders. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1706–1721, Online. Association for Computational Linguistics.

Linlin Wang, Zhu Cao, Gerard de Melo, and Zhiyuan Liu. 2016. Relation classification via multi-level attention CNNs. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1298–1307, Berlin, Germany. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Shanchan Wu and Yifan He. 2019. Enriching pre-trained language model with entity information for relation classification. CIKM '19, page 2361–2364, New York, NY, USA. Association for Computing Machinery.

Minguang Xiao and Cong Liu. 2016. Semantic relation classification via hierarchical recurrent neural network with attention. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1254–1263, Osaka, Japan. The COLING 2016 Organizing Committee.

Kun Xu, Siva Reddy, Yansong Feng, Songfang Huang, and Dongyan Zhao. 2016. Question answering on Freebase via relation extraction and textual evidence. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2326–2336, Berlin, Germany. Association for Computational Linguistics.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2335–2344, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Daojian Zeng, Haoran Zhang, and Qianying Liu. 2020. Copymtl: Copy mechanism for joint extraction of entities and relations with multi-task learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9507–9514.

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.

Meishan Zhang, Yue Zhang, and Guohong Fu. 2017a. End-to-end neural relation extraction with global optimization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1730–1740, Copenhagen, Denmark. Association for Computational Linguistics.

Ranran Haoran Zhang, Qianying Liu, Aysa Xuemo Fan, Heng Ji, Daojian Zeng, Fei Cheng, Daisuke Kawahara, and Sadao Kurohashi. 2020. Minimize exposure bias of Seq2Seq models in joint entity and relation extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 236–246, Online. Association for Computational Linguistics.

Shu Zhang, Dequan Zheng, Xinchen Hu, and Ming Yang. 2015. Bidirectional long short-term memory networks for relation classification. In *Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation*, pages 73–78, Shanghai, China.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017b. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 35–45, Copenhagen, Denmark. Association for Computational Linguistics.

Zexuan Zhong and Danqi Chen. 2021. A frustratingly easy approach for entity and relation extraction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 50–61, Online. Association for Computational Linguistics.

GuoDong Zhou, Jian Su, Jie Zhang, and Min Zhang. 2005. Exploring various knowledge in relation extraction. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 427–434, Ann Arbor, Michigan. Association for Computational Linguistics.

Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 207–212, Berlin, Germany. Association for Computational Linguistics.