# Improving Persian Relation Extraction Models by Data Augmentation

**Moein Salimi Sartakhti**
Shahid Beheshti University
Tehran, Iran
sartakhti.salimi@gmail.com

**Romina Etezadi**
Shahid Beheshti University
Tehran, Iran
ro.etezadi@mail.sbu.ac.ir

**Mehrnoush Shamsfard**
Shahid Beheshti University
Tehran, Iran
m-shams@sbu.ac.ir

## Abstract

Relation extraction that is the task of predicting semantic relation type between entities in a sentence or document is an important task in natural language processing. Although there are many researches and datasets for English, Persian suffers from sufficient researches and comprehensive datasets. The only available Persian dataset for this task is PERLEX, which is a Persian expert-translated version of the SemEval-2010-Task-8 dataset. In this paper, we present our augmented dataset and the results and findings of our system, participated in the Persian relation Extraction shared task of NSURL 2021 workshop. We use PERLEX as the base dataset and enhance it by applying some text preprocessing steps and by increasing its size via data augmentation techniques to improve the generalization and robustness of applied models. We then employ two different models including ParsBERT and multilingual BERT for relation extraction on the augmented PERLEX dataset. Our best model obtained 64.67% of Macro-F1 on the test phase of the contest and it achieved 83.68% of Macro-F1 on the test set of PERLEX.

## 1 Introduction

The task of detecting semantic relations between entities in a text is called Relation Extraction (RE). RE plays an important role in various natural language processing (NLP) tasks such as Information Extraction, Knowledge Extraction, Question Answering, Text Summarization, etc. According to the literature, RE tasks can be divided into two categories: sentence-level and document-level. The goal of the sentence-level RE task is to obtain the relation between two known entities (predefined entities) in a sentence. Nevertheless, the document-level RE task aims to extract the relationship among several entities in a long text which usually contains multiple sentences. According to the differences mentioned earlier, document level relation extraction is more complicated than sentence-level.

In the RE task, entities are string literals that are marked in the sentence and the aim is to identify a limited number of predefined relationships between these entities from the input text. Different tasks can benefit from using RE. For example, suppose that the goal of an information extraction system is to extract corporations located in Iran from a text. For this purpose, the RE component may use the *located-in* predicate and *Iran* as the object of the relation to allow this information to be extracted. Moreover, suppose a question answering system, which is going to answer a question about the cause of an event. It may exploit an RE task in which the relationship is *Cause-Effect* and the object should be that specific event (Asgari-Bidhendi et al., 2021).

Another important application of RE is knowledge base creation. A knowledge base includes a set of entities and relationships between them. Most of the available large knowledge bases such as Yago (Suchanek et al., 2007), Freebase (Bollacker et al., 2008), DBpedia (Auer et al., 2007), and Wikidata (Vrandei and Krtzsch, 2014) are encoded in English. In Persian, there is a knowledge base (knowledge graph) called Farsbase (Asgari-Bidhendi et al., 2019). There are some standard RE datasets for the English language, such as SemEval-2010-Task 8 and TACRED. For Persian which is a low resource language in this field, the only RE dataset (up to authors' knowledge) is PERLEX, which is an expert-translated version of SemEval- 2010-Task-8 dataset.

PERLEX has 10717 sentences and there is a relation and two entities in each sentence. In PERLREX, the boundaries of each entity have been specified by certain tokens. For example, the first en-

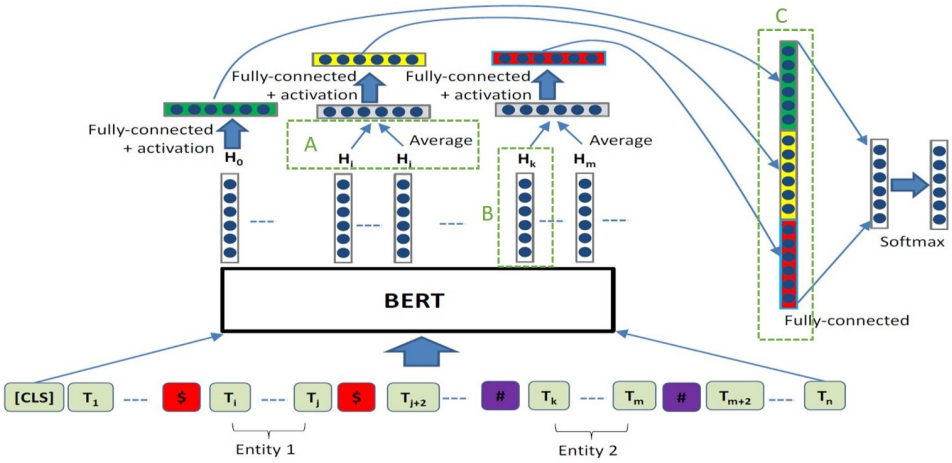


Figure 1: R-BERT structure.

tity uses the tags <e1> and </e1> for the start and end of the entity (also <e2> and </e2> are used for the second entity). Table 1 shows some examples of annotated sentences.

Our contributions in this work are as follows: (1) Using text augmentation techniques to increase the size of the PERLEX dataset. (2) Preprocessing the PERLEX to fix some of the issues which improves the performance of the latest Persian relation extractor. In this paper, a relation extraction system is presented which is submitted to the Second Workshop on NLP Solutions for Under Resourced Languages (NSURL 2021). Some modifications on available models are adopted and the effects of each modification on the total generalization and robustness are reported. The remainder of this paper is organized as follows: the methodology is described in Section 2. Section 3 shows the experimental results. Section 4 concludes the paper.

## 2 Methodology

### 2.1 Data Preprocessing

Although there are many datasets for English and other rich-resource languages, Persian has no comprehensive available resources for the RE task. Data annotating is a challenging, time-consuming, and cost-consuming task. Therefore, in the data preprocessing step we try to leverage techniques like text augmentation to increase the size of PRELEX. Some preprocessing is also applied to PERLEX. The preprocessing and text augmentation steps are shown in Figure 2.

The preprocessing and text augmentation procedure both includes three sub-steps. Text preprocessing sub-steps are listed below:

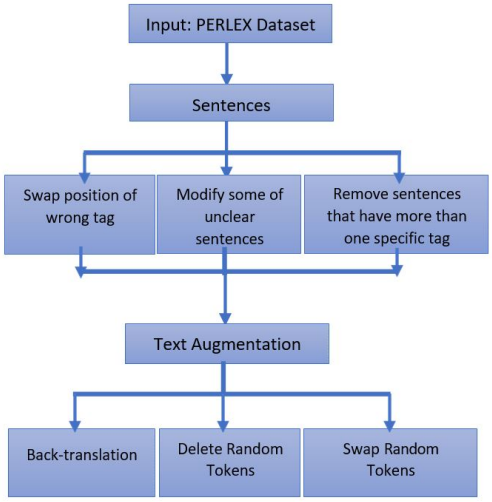


Figure 2: Text preprocessing and text augmentation procedure.

- Swap position of the wrong tag
- Modify the unclear sentences
- Remove sentences which have more than one specific tag

As PERLEX is translated semi-automatically, there are some problems in it, such as:

- Some of the sentences have more than one tag <e1> or </e1> or <e2> or </e2>. As it is supposed that each sentence contains one relation, such sentences are filtered. 975 sentences have this problem and are removed from the dataset (See the 4th sentence in Table 1).

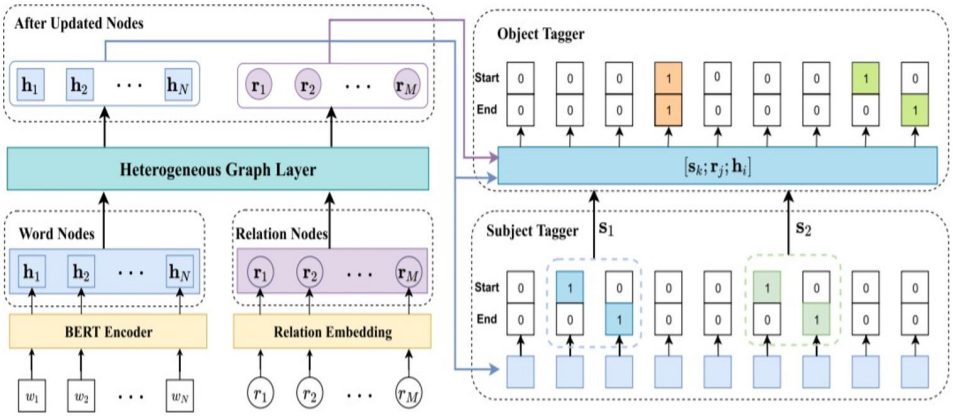


Figure 3: RIFRE structure.

Table 1: Some correct and wrong examples of the PERLEX.

| Relation Type | Sentence |
|---|---|
| Product-Producer(e2,e1) | The <e1>company</e1> fabricates plastic <e2>chairs</e2>. |
| Message-Topic(e1,e2) | The major theme of the <e1>book</e1> is the <e2>beauty</e2> of a dream. |
| Entity-Destination(e1,e2) | He has just sent <e1>spam</e1> to the <e2>clients</e2>. |
| Message-Topic(e1,e2) | I read the <e1>report</e1> from Somalia on the <e2>agreement</e2> reached by faction leaders on the form of a future government that has <e2>been</e2> warmly welcomed. |

- In all of the sentences that <e2> (<e1>) comes exactly before </e1> (</e2>), position of these tags is swapped. This issue is fixed by detecting these sentences and swapping the tokens. 344 sentences have this problem.

- Some of the unclear translated sentences in PERLEX have been modified.

After the data preprocessing step, some noise are added and the text augmentation techniques are applied to increase the size of the PERLEX. Some of the employed techniques are listed below:

- Deleting a token in each sentence randomly

- Swapping positions of some tokens randomly

- Using the Back-translation method (Shleifer, 2019) in order to increase the size of PERLEX dataset.

There are different ways for back-translating. For example, one way can be the translation of sentences to English, then to Arabic, and finally, return sentence to Persian. However, in this paper, each sentence is translated from Persian to English and then it is back-translated to Persian by using the python API of the google translate package[1]. Therefore, this method can increase PERLEX size from 9381 to 18762. Reaching 18762 sentences for Persian is an important achievement in the RE task.

## 2.2 Applied Models

This section describes different models that the data augmentation is applied on them: R-BERT (Wu and He, 2019) and RIFRE (Zhao et al., 2021). After the preprocessing and text augmentation steps, two state-of-the-art models R-BERT and RIFRE are employed.

**R-BERT:** The main structure of R-BERT is shown in Figure 1. For a sentence with two target entities e1 and e2, $ has been inserted at both the beginning and end of the first entity, and # at both the beginning and end of the second entity. Also, there is a [CLS] symbol at the beginning of each sentence. We finetune the pre-trained ParsBERT (Farahani et al., 2021) and Multilingual BERT (Libovick et al., 2019) models on the augmented PERLEX. In addition, table 2 shows

[1]https://pypi.org/project/googletrans/

Table 2: Parameters settings for the R-BERT model.

| Parameters | Value |
|---|---|
| Batch size | 16 |
| Max sentence length | 128 |
| Adam learning rate | 2e-5 |
| Number of epochs | 10 |
| Dropout rate | 0.1 |

Table 3: Parameters settings for the RIFRE model.

| Parameters | Value |
|---|---|
| Batch size | 16 |
| Max sentence length | 128 |
| Adam learning rate | 1e-1 |
| Number of epochs | 10 |
| Dropout rate | 0.1 |

other hyperparameters of R-BERT. Furthermore, we experiment with different combination of embeddings produced by R-BERT to reach the best model (See embeddings A, B,and C in Figure 1).

Some of the modifications on the R-BERT are listed below:

- R-BERT_V1: Average all of the three final embeddings in the fully connected layer rather than a concatenation of them (see Figure 1-C).
- R-BERT_V2: Concatenation all of the embeddings of tokens in each entity rather than average them (Figure 1-A).
- Using the last (first) token instead of average all of the embeddings of tokens in the entities (Figure 1-B).
- Using the Multilingual BERT and ParsBERT to reach the best decision

**RIFRE:** This work proposes a representation iterative fusion based on a heterogeneous graph neural network for joint entity and relation extraction. As shown in Figure 3, RIFRE models relations and words as nodes on the graph and update the nodes through a message passing mechanism. The model performs relation extraction after nodes are updated. First, the subject tagger is used to detect all possible subjects on the word nodes. Then, RIFRE combines each word node with the candidate subject and relation, and the object tagger is used to tag the object on the new word nodes. In this paper, RIFRE is adopted with the ParsBERT and Multilingual BERT.

## 3 Evaluation

There are three main ways to evaluate the RE classification results:

- Taking into account both variations of each class (18 classes in total).

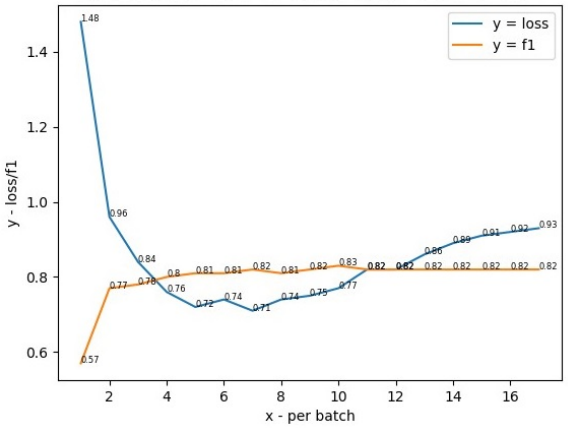


Figure 4: F-Score and Loss per epochs on the V1 R-BERT

- Using only one variation of each class (and considering directionality).
- Using only one variation of each class (and ignoring directionality).

Moreover, there are two approaches to calculate F1-score: Micro-averaging and Macro-averaging. In this dataset, those pairs of entities that do not fall into any of the main nine classes are labeled as the "Other" class. The "Other" class is not participated in the evaluation phase. In this section, the official evaluation method is used for the SemEval-2010-Task-8 dataset, which is (9+1)-way classification with macro-averaging F1-score measurement while directionality is taken into account. This (9+1)-way means that the nine main classes plus Other in training and testing is considered, but "Other" is ignored to calculate the F1-scores.

## 4 Results

### 4.1 Development Phase

In the development phase, PERLEX dataset is used and some improvements are achieved. Table 2 shows the major parameters used in R-BERT experiments. Hyperparameters of the RIFRE are

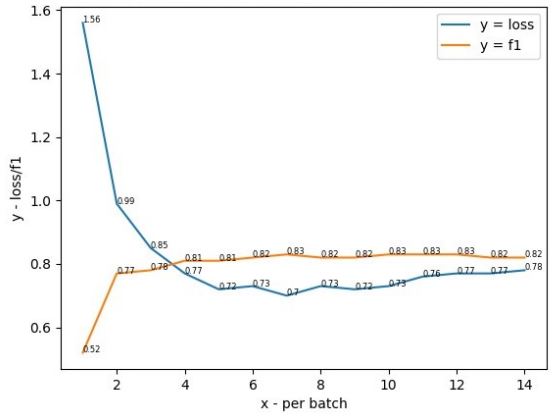


Figure 5: F-Score and Loss per epochs on the V2 R-BERT

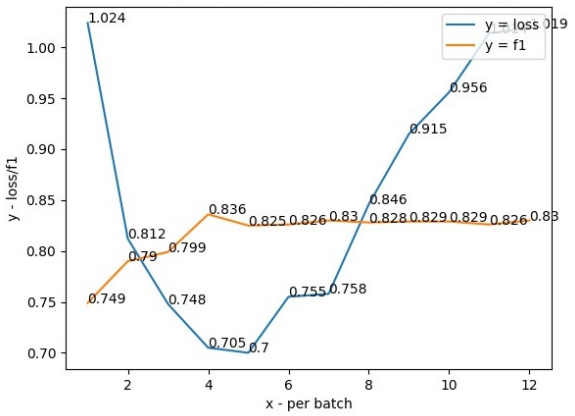


Figure 6: F-Score and Loss per epochs on the V3 R-BERT

shown in Table 3. Table 4 shows the performance of the various models which are used. R-BERT model produces the best results, while RIFRE model produces the worst according to table 4. Figures 4, 5 and 6 show the loss and F1-score value per epochs. According to these evaluations, simple R-BERT has better results than V1, V2, and V3 variation of the R-BERT. As table 4 shows all of the results, the best model is the simple R-BERT which has achieved F1-Score 83.68 on the test set.

### 4.2 Test Phase

Finally, results show that the proposed model reaches 64.67 of Macro-F1 score on the shared task test data in NSURL contest.

## 5 Conclusion

In this paper, the PERLEX dataset is used which is a Persian expert-translated version of the

Table 4: Performance of the models on PERLEX.

| Models | F1-score |
|---|---|
| Simple R-BERT | 83.86% |
| R-BERT_V1 | 83.02% |
| R-BERT_V2 | 83.11% |
| R-BERT_V3 | 83.08% |
| RIFRE | 79.54% |

Table 5: Performance of the models on different relations types in PERLEX.

| Relation Types | F1-score |
|---|---|
| Cause-Effect | 61.70% |
| Content-Container | 59.26% |
| Entity-Destination | 76.01% |
| Entity-Origin | 58.04% |
| Instrument-Agency | 75.54% |
| Member-Collection | 32.85% |
| Message-Topic | 76.06% |
| Other | 40.95% |

”SemEval-2010-Task-8” dataset. As data annotating is a challenging, time-consuming and cost-consuming task, we employ some of the text preprocessing and text augmentation techniques such as back-translation, deleting random tokens, and swapping random tokens. The Preprocessing and text augmentation could increase F-Score by about four percent in comparison to the last and best work on Persian. After preparing the PERLEX, we apply two state-of-the-art models namely R-BERT and RIFRE. In addition, we extend the R-BERT model by changing the R-BERT structure. Pre-trained BERT models that are tested in this paper are ParsBERT and Multilingual Bert. Results show that ParsBERT based on the simple R-BERT structure had a better result than other variations of the R-BERT models and RIFRE. The contributions in this paper are using text augmentation techniques to increase the size of the PERLEX dataset, and preprocessing the PERLEX dataset to fix some of the issues which improves the performance of the latest Persian relation extractor.

## References

Majid Asgari-Bidhendi, Ali Hadian, and Behrouz Minaei-Bidgoli. 2019. Farsbase: The persian knowledge graph. *Social Work*, 10(6):1169–1196.

Majid Asgari-Bidhendi, Mehrdad Nasser, Behrooz Janfada, and Behrouz Minaei-Bidgoli. 2021. Perlex: A

bilingual persian-english gold dataset for relation extraction. *Scientific Programming*, 2021:1–8.

Sren Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: a nucleus for a web of open data. In *ISWC'07/ASWC'07 Proceedings of the 6th international The semantic web and 2nd Asian conference on Asian semantic web conference*, volume 4825, pages 722–735.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250.

Mehrdad Farahani, Mohammad Gharachorloo, Marzieh Farahani, and Mohammad Manthouri. 2021. Parsbert: Transformer-based model for persian language understanding. *Neural Processing Letters*, pages 1–17.

Jindrich Libovick, Rudolf Rosa, and Alexander Fraser. 2019. How language-neutral is multilingual bert? *arXiv preprint arXiv:1911.03310*.

Sam Shleifer. 2019. Low resource text classification with ulmfit and backtranslation. *arXiv preprint arXiv:1903.09244*.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706.

Denny Vrandei and Markus Krtzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of The ACM*, 57(10):78–85.

Shanchan Wu and Yifan He. 2019. Enriching pretrained language model with entity information for relation classification. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 2361–2364.

Kang Zhao, Hua Xu, Yue Cheng, Xiaoteng Li, and Kai Gao. 2021. Representation iterative fusion based on heterogeneous graph neural network for joint entity and relation extraction. *Knowledge Based Systems*, 219:106888.