13th Conference on Language Resources and Evaluation, LREC 2022, Marseille

DeepREF: A Framework for Optimized Deep Learning-based Relation Classification

Igor Nascimento[1], Rinaldo Lima[1], Adrian Chifu[2], Bernard Espinasse[2] , Sébastien Fournier[2]
[1] Universidade Federal Rural de Pernambuco (UFRPE), Recife, Brazil
[2] Aix-Marseille Université, Université de Toulon, CNRS, LIS UMR 7020, Marseille, France
{igor.lucena;rinaldo.jose}@ufrpe.br & {adrian.chifu; bernard.espinasse; sebastien.fournier}@lis-lab.fr

Abstract

Relation Extraction (RE) is an important basic Natural Language Processing (NLP) task for many applications, including search engines and question-answering systems. There are many studies in this subarea of NLP that continue to be explored, such as the ones concerned by SemEval shared tasks. For many years, several RE systems based on statistical models have been proposed, as well as the frameworks to develop them. We focus on frameworks allowing to develop such RE systems using deep learning models. Such frameworks make it possible to reproduce experiments using many deep learning models and preprocessing techniques. Currently, there are very few frameworks of this type. In this paper, we propose an open and optimizable framework called DeepREF, inspired by two other existing frameworks: OpenNRE and REflex. DeepREF allows the rapid development of deep learning models for Relation Classification (RC). In addition, it enables hyperparameter optimization, and the application of many preprocessing techniques on the input textual data. DeepREF provides means to boost the process of running deep learning models for RC tasks on different datasets and models. DeepREF is evaluated on three reference corpora and has demonstrated competitive results compared to other state-of-the-art RC systems.

Keywords: Relation Classification, Framework, DeepREF, Embeddings, SemEval, DDI, Optuna, NLP

1. Introduction

RE is a subfield of study in Information Extraction (IE). The main goal of an IE system is to extract specific pieces of information from some document repository, providing structured information as output. The information extracted from data is basically of three types: named entities, relations, and events. RE has many applications, including document summarization, machine translation, and the automatic construction of thesauri or semantic networks (Pawar et al., 2017).

There are many initiatives that tackle the RE challenge, but it is difficult to reproduce them due to the lack of implementation details in the papers introducing them. Such difficulties include the preprocessing method used, the model hyperparameters, and the details of the model implementation. Not all papers provide the implementation code and, even when it is the case, it is still difficult to reproduce these experiments on other datasets, in most of the time.

In order to develop and evaluate RE systems efficiently, dedicated RE Frameworks have been proposed. These RE Frameworks allow performing several experiments on many datasets. They provide such a level of modularity and extensibility that makes the preprocessing stage, with different datasets, easier to implement.

This paper focuses on frameworks for developing RE systems using deep learning models. Currently there are very few frameworks of this type. To mitigate that, we propose a new open and optimizable framework called DeepREF. Our framework is inspired by the OpenNRE (Han et al., 2019) and REflex (Chauhan et al., 2019), two existing frameworks for Relation Classification based on deep learning. DeepREF allows the rapid development of deep learning models for Relation Classification (RC). In addition, it enables hyperparameter optimization based on the Optuna Framework (Akiba et al., 2019), and the application of many preprocessing techniques on the input textual data. DeepREF also provides means to boost the process of running deep learning models for RC tasks on different datasets and models.

This paper is structured as follows. Section 2 first presents RE Frameworks using statistical techniques. Then, two existing deep learning model-based frameworks, (OpenNRE (Han et al., 2019) and REflex (Chauhan et al., 2019)) are presented in detail with their strengths and weaknesses. Their functionalities are compared, and new ones are defined to be included in our new RE framework, named DeepREF. In Section 3 we describe DeepREF framework, an open and optimizable framework for deep learning-based relation extraction, its software architecture along with its components, the neural embeddings available, and its hyperparameter optimization process. Section 4 presents an experimental evaluation of DeepREF on three reference corpora: the SemEval 2010 Task 8 corpus (Hendrickx et al., 2019), the SemEval 2018 Task 7 corpus (Gábor et al., 2018), and the DDI Extraction 2013 corpus (Herrero-Zazo et al., 2013). Finally, we conclude the paper (Section 5) presenting future work of this research.

Detailed information about DeepREF is available at: `https://github.com/igorvlnascimento/DeepREF`

2. Related Work

In this section, we present several RE frameworks for developing RE systems using statistical techniques. Then, two RE framework based on deep learning models, OpenNRE (Han et al., 2019) and REflex (Chauhan et al., 2019) are described in more details contrasting their strengths and weaknesses. Their functionalities are compared. Then, new functionalities available in DeepREF are defined.

2.1. Statistical oriented RE Frameworks

The framework for RE proposed by Muzaffar et al. (2015) is specialized on the medical domain and is based on machine learning approaches. Their main contribution was generating a feature set ranked by the Unified Medical Language System (UMLS). It is a set of files that brings together health and biomedical terminology (Bodenreider, 2004) and it is useful to extract concepts, relationships, and knowledge. They also use a hybrid approach using machine learning with bag of words, natural language processing and semantic representation to extract biomedical relations (Muzaffar et al., 2015).

The ENRE framework was designed to extract entities and relations between them from programming languages (Jin et al., 2019). This framework has the advantage of being easily extensible to new programming languages from different paradigms. This is important because, nowadays, the systems are built by companies employing various programming languages, with different paradigms (Jin et al., 2019).

Another framework uses Reinforcement Learning (RL) to extract relations. This framework worked in a hierarchical manner, decomposed into high and low-level tasks to both detect and classify relations between two entities. The use of RL outperforms some baseline models even for extracting overlapping relations (Takanobu et al., 2019).

The Relation Extraction Learning Framework (REEL) addresses the challenges of learning to extract new relations over user defined text collections. It employs machine learning toolkits for text processing. Some of its advantages include the fact that this framework is publicly available, and it is able to handle various input text formats (Barrio et al., 2014).

2.2. Deep Learning oriented RE Frameworks

Due to the successful use of deep learning in NLP, some deep learning-based RE frameworks have recently been proposed. To the best of our knowledge, there are only two frameworks for Deep Learning-based RE: OpenNRE (Han et al., 2019), and REflex (Chauhan et al., 2019), which we describe next.

The OpenNRE Framework

The OpenNRE framework is an open-source and extensible toolkit that allows easy implementation of deep learning-based RE models (Han et al., 2019). OpenNRE provides system encapsulation and model extensibility that make it easy to build new models from existing ones. The OpenNRE architecture allows training models with only a few lines of code. Although the model training and evaluation require data preparation and preprocessing, OpenNRE does not provide enough support for such tasks in its preprocessing module. This framework has functional modules constructed using both Tensorflow and Pytorch libraries (Han et al., 2019). This framework also provide some datasets already preprocessed, including SemEval 2010, Wiki80, and NYT 2010 corpora. OpenNRE framework provides the following deep learning models: CNN, PCNN, and BERT. The aforementioned models and their setting are based on the works of (Nguyen and Grishman, 2015; Zeng et al., 2015; Baldini Soares et al., 2019), respectively. The OpenNRE framework is well documented.

The REflex Framework

REflex is also an open-source and extensible framework for the development of RE deep learning-based models (Chauhan et al., 2019). This framework provides a module that preprocess data from different datasets, converting them into one specific representation format. As a result, it is possible to apply different types of preprocessing on the input data to be evaluated. REflex provides modules to deal with diverse data inputs, and evaluations on RE models including hyperparameter tuning, cross-validation, and statistical significance testing functions. These modules allow performing ablation studies, and the direct performance comparison of RE models. Moreover, reproducibility and extensibility are possible using the REflex framework. REflex provides a module to use, train, and evaluate models. However, it does not provide system encapsulation, and model extensibility similar to OpenNRE (Chauhan et al., 2019).

2.3. Towards a New Deep Learning-based RE Framework

In this section, we compare strengths and weaknesses of both OpenNRE and REflex. In addition, we contrast OpenNRE and REflex functionalities with the ones available in our framework DeepREF.

OpenNRE framework has three advantages compared to REflex framework. First, OpenNRE has implemented several state-of-the-art RE models, including attention mechanism, adversarial learning, and reinforcement learning. Second, OpenNRE provides good system encapsulation. It decomposes the Relation Classification pipeline into four stages: embedding construction, encoder, selector (for distant supervision), and classifier. For each stage, it has implemented many methods. System encapsulation makes it easy to train models by changing hyperparameters. Third, OpenNRE is extendable. Users can construct new RE models by choosing specific blocks provided along with the four stages as mentioned above and combining them freely, with only a few lines of codes.

On the other hand, REflex framework, contrary to OpenNRE, allows reproducing many experiments from different models and datasets. Moreover, REflex has a tailored preprocessing module for some public datasets, and it is also possible to extend to other datasets easily. Besides different preprocessing types (digit blinding, punctuation and stop words removal, entity and Named Entity Recognition (NER) blinding replacement), there are other ways to compare and improve model's result such as hyperparameter tuning, split bias train-test set to check statistical significance, and word embeddings (ELMo and BERT-tokens) (Chauhan et al., 2019). REflex provides more types of embeddings than OpenNRE.

Table 1 not only presents a comparison between OpenNRE and REflex functionalities, but also the functionalities available in DeepREF.

3. The DeepREF Framework

This section introduces DeepREF, a framework for deep learning-based relation classification. Its functional architecture is depicted in Figure 1. First, its architectural components are presented, followed by the embeddings available in it. Finally, its hyperparameter optimization is briefly discussed.

3.1. The DeepREF Framework Architecture

DeepREF implementation is mainly based on OpenNRE one. Some new converter classes for processing the DDI corpus were also integrated from the REflex modules. The DeepREF architecture is composed of four main modules: "NLP module", "Text Preprocessing module", "Sentence & token encoding module", and "Deep learning module". They are described next.

3.2. NLP module

In the NLP module, it is performed both lexical and syntactic analysis that annotate the input text, after tokenization, with POS tags, dependency labels, and NER. Stanza [1] tools for linguistic analysis were integrated as an option to be chosen from the framework for many text processing tasks. SpaCy [2] open-source library for advanced NLP in Python was also selected for dataset preprocessing using its simplest model pretrained (*en_core_web_sm*). Finally, this module performs a simple semantic analysis using Wordnet, in which, for any candidate entity in a sentence, we retrieve its direct hypernyms up to 2 levels above.

3.3. Text Preprocessing module

This module is mainly based on REflex framework with some further improvements. The following preprocessing tasks are available on REflex which was also integrated in our work: preprocessing (*punct_digit*[3], *punct_stop_digit*[4], *entity_blinding*[5]). In fact, we refactored REflex code to become possible to make a combination of all the aforementioned preprocessing tasks. For instance, its possible in DeepREF to combine punctuation, stop words removal, digit, entity blinding or NER blinding, since these two last tasks are mutually exclusive because they all change the entity words with a more generic one like "ENTITY" (for SemEval 2010 and SemEval 2018); and "DRUG" (for DDI 2013). Digit blinding also changes numeric tokens with a more general word: "NUMBER". In addition, text between brackets and parenthesis can be also removed from the text. We improved the *Preprocess* class to facilitate the inclusion of new datasets, and preprocessing them without replication code, as it was found in REFlex original code. At the end of this step, training data of each dataset is splitted randomly in where part of it can be defined as validation data.

3.4. Sentence & token encoding module - Embeddings

After the text preprocessing step, this module can perform both sentence and token encoding to generate text embeddings. Besides the BERT sentence/token embedding, DeepREF can also produce four other types of embeddings: position embedding, semantic knowledge (SK) embedding, a part-of-speech tagging, and dependency label.

Position embedding was extracted with the positions of the head and tail entity.

SK embedding was done by extracting the two hypernyms (up to 2 levels) of the tokens forming the entities retrieved using *WordNet* (Fellbaum, 1998) and NLTK (Loper and Bird, 2002).

POS embedding is built by extracting the POS tags of a sentence, and generating an embedding with the POS tag sequence of the sentence.

Deps embedding is built by extracting the dependency graph derived from the dependency parsing preprocessing task. Figure 2 shows an example of such a graph.

Thus we obtain all the dependencies in the sentence and produce an embedding with them. We generate the embedding of each type above by means of the *Embedding*[6] Pytorch function.

In the following, it is demonstrated how we generate and use these embeddings.

Each sentence in dataset has a sequence of tokens, i.e., $s = [w_1, ..., w_n]$, where n is the length of the sentence. Each token has a dense vector-representation, i.e., $v = [v_1, ..., v_n] \in \mathcal{R}^{d_w \times n}$, where d_w is the dimension of the word embedding.

In addition to the word embedding, we used position, SK, POS tags, and deps embeddings. To obtain the

[1] https://stanfordnlp.github.io/stanza

[2] https://spacy.io

[3] punctuation removal and digit blinding

[4] digit blinding, stop words and punctuation removal

[5] entity blinding replacement

[6] https://pytorch.org/docs/stable/generated/torch.nn.Embedding.html

	OpenNRE	***REflex***	***DeepREF***
NLP tool	None	SpaCy	SpaCy, **Stanza**
Evaluation metric	Micro-F1	Micro and Macro-F1; Confusion matrix	Micro, Macro and **Weighted**-F1; Confusion matrix
Domain	General	General, Biomedical and Clinical	General, Biomedical and **Scientific**
Learning Model	CNN, PCNN and BERT	CRCNN	CNN, PCNN, CRCNN, BERT, **GRU/BiGRU, LSTM/BiLSTM**
Embedding	Glove and BERTbase	Senna, ELMo, BERT-Tokens	Glove, Senna, BERTbase, **FastText Wiki,, FastText Crawl, BioBERT, SciBERT, Sentence-BERT, Semantic (WordNet), POS Tag, Dependency Labels**
Text preprocessing	None	Entity/NER/Digit blinding; punctuation and stopwords removal	Entity/NER/Digit blinding; **text between brackets or parenthesis**, punctuaction and stowords removal
Modularized	Yes	No	Yes
Setup	Easy	Difficult	Easy
Optimizable	No	Yes (without pruning algorithm)	Yes **(with pruning algorithm)**
Linguistic Preprocessing	tokenizaztion	tokenization, NER, POS tag	tokenization, NER, POS tag, **Dep. Parsing, WordNet**
Evaluation Dataset	SemEval 2010 Task-8. Wiki80, TACRED	SemEval Task-8, DDI, i2b2	SemEval 2010 Task-8, **SemEval 2018 Task-7**, DDI

Table 1: Comparison of the deep learning-based RC frameworks.

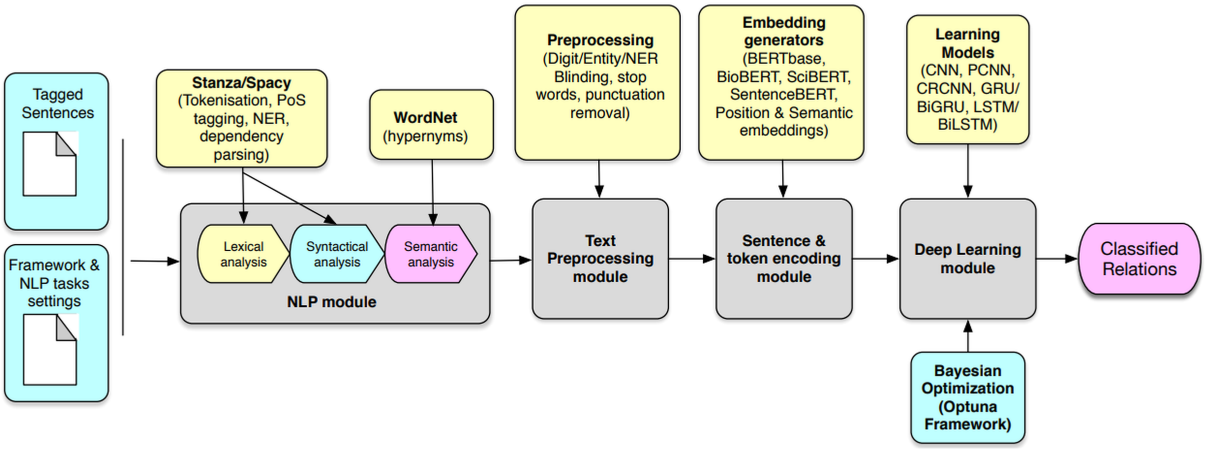
Tagged Sentences
Framework & NLP tasks settings
Stanza/Spacy (Tokenisation, PoS tagging, NER, dependency parsing)
WordNet (hypernyms)
Preprocessing (Digit/Entity/NER Blinding, stop words, punctuation removal)
Embedding generators (BERTbase, BioBERT, SciBERT, SentenceBERT, Position & Semantic embeddings)
Learning Models (CNN, PCNN, CRCNN, GRU/ BiGRU, LSTM/ BiLSTM)
Lexical analysis
Syntactical analysis
Semantic analysis
NLP module
Text Preprocessing module
Sentence & token encoding module
Deep Learning module
Classified Relations
Bayesian Optimization (Optuna Framework)

Figure 1: DeepREF Functional Architecture

word embedding of the entities, we access the final hidden layers corresponding to the word tokens in each entity mention to produce two vectors v_{eh} and v_{et} that corresponds to the word embeddings of the head and tail entities, respectively. The position, SK, POS tags and deps embeddings are actually the position, SK, POS tags and deps from the head and tail entities, i.e., $\mathbf{r}^e_{pos}$, $\mathbf{r}^e_{sk}$, $\mathbf{r}^e_{pt}$, $\mathbf{r}^e_{deps}$, respectively, where pos, sk, pt, $deps$ denote position, SK, POS tags and deps embeddings, respectively. The superscript e corresponds to the entity independently to be head or tail entities.

The position embedding is obtained by extracting the first entity index in the sentence. The SK embedding is obtained by extracting two hypernyms (up to level 2 in WordNet hierarchy) related to the entity. For instance, the entity ”company” has ”institution” as the closer hypernym, that has ”organism” as its closer hypernym. The POS tags and deps embeddings are obtained by extracting the POS tags and graph dependencies of the whole sentence, using the *SpaCy* NLP toolkit. We con-

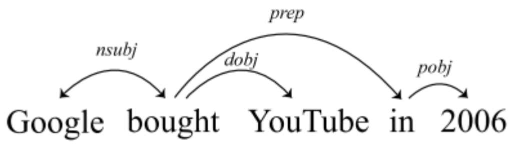

Figure 2: An example of a dependency graph (Nastase et al., 2013).

sider only the POS tags and deps for head and tail entities at this step. These embeddings is formally described as following:

$$s = [w_1, ..., w_{eh}, ..., w_{et}, ..., w_n, w_{skh}, w_{skt}] \tag{1}$$

$$v = [v_1, ..., v_{eh}, ..., v_{et}, ..., v_n, v_{skh}, v_{skt}] \tag{2}$$

$$\mathbf{r}^{eh} = [v_{eh}; v_{skh}; r^{eh}_{pos}; r^{eh}_{pt}; r^{eh}_{deps}] \tag{3}$$

$$\mathbf{r}^{et} = [v_{et}; v_{skt}; r^{et}_{pos}; r^{et}_{pt}; r^{et}_{deps}] \tag{4}$$

$$r^{e}_{embed} = \mathbf{W} \cdot \mathbf{X} + \mathbf{b} \tag{5}$$

$$\mathbf{r}^{e} = [\mathbf{r}^{eh}; \mathbf{r}^{et}] \tag{6}$$

where $r^{e}_{embed} \in \mathcal{R}^{d_e}$, d_e is the dimension of the r^{e}_{embed} and is equal to 5. $embed$ corresponds the type of embeddings and may be replaced by position, POS tags or deps embeddings. $v_i \in \mathcal{R}^{d_w}$ and d_w is equal to 768. Then, $d_{r^{eh}} = d_{r^{et}} = 2 \times d_w + 3 \times d_e$ and $d_{r^e} = 2 \times d_{r^{eh}} = 2 \times d_{r^{et}}$, where $d_{r^{eh}}$ is the dimension of head entity and $d_{r^{et}}$ is the dimension of tail entity. Formulas 3, 4 and 6 are concatenations of embeddings. In particular, we propose an enhanced entity-aware word embedding approach enriched with semantic features of head and tail entities, called E-BEM, which stands for enhanced BERT Entity Mention, an enhanced version of BERT-EM (Baldini Soares et al., 2019). E-BEM is simply the concatenation of the BERT-EM encoding with the SK embeddings.

3.5. Deep Learning module - Hyperparameter optimization

After the sentence and token encoding step, the Deep Learning module performs the learning process using deep learning neural networks available in DeepREF including CNN, PCNN, CRCNN, GRU/BiGRU, and LSTM/BiLSTM. The learning process is performed under a hyperparameters optimization step using the *Optuna* framework.

Hence, DeepREF has some advantages compared to other frameworks that do not employ hyperparameters optimization.

It uses, by default, SOTA hyperparameters optimization algorithms, like *Tree Parzen Estimators* and *HyperBand*, that has achieved competitive performance on finding the best hyperparameters (Yu and Zhu, 2020). These hyperparameters include Batch size, Learning rate, Weight decay[7], Maximum sentence length, Maximum number of training epochs.

Other hyperparameters used were taken from (Zeng et al., 2014; Zeng et al., 2015; Baldini Soares et al., 2019).

4. Experimental Evaluation

In this section, we briefly present the three datasets considered for evaluation of DeepREF framework, and then the experiments with their evaluation methodology and settings.

4.1. datasets

SemEval 2010 Task 8 is a dataset built with the goal to create a testbed for automatic classification of semantic relations. The SemEval 2010 focuses on semantic relation between pairs of nominals. There are 9 types of relations annotated on this dataset. Table 2 summarizes the relation types and their annotation statistics.

Relation	**Train frequency**	**Test frequency**	**Total per relation**
Cause-Effect	1003/12.5%	328/12.1%	1331/12.4%
Component-Whole	941/11.8%	312/11.5%	1253/11.7%
Entity-Destination	845/10.6%	292/10.7%	1137/10.6%
Entity-Origin	716/8.9%	258/9.5%	974/9.1%
Product-Producer	717/9.0%	231/8.5%	948/8.8%
Member-Collection	690/8.6%	233/8.6%	923/8.6%
Message-Topic	634/7.9%	261/9.6%	895/8.4%
Content-Container	540/6.8%	192/7.1%	732/6.8%
Instrument-Agency	504/6.3%	156/5.7%	660/6.2%
Other	1410/17.6%	454/16.7%	1864/17.4%
Total	8000	2717	10717

Table 2: SemEval 2010 Task-8 annotation statistics.

SemEval 2018 Task-7 focuses on semantic relation analysis of scientific *corpus*. Different from other SemEval datasets, such as SemEval 2017 Task-10, SemEval 2018 has more semantic relations and is composed of scientific paper abstracts. There are 6 discrete categories of semantic relations in a text (*Usage, Topic, Model-Feature, Part-Whole, Compare and Result*) scientific domain-specific. (Gábor et al., 2018).

SemEval 2018 used the macro-F1 metric to compare with models participating in the challenge, and we adopted this metric for comparison too. Table 3

[7] This hyperparameter is only optimized for CNNs and RNNs, not for BERT. We use the default weight decay value for BERT.

presents the results and distribution of the relations on SemEval 2018 Task-7 dataset, for its sub-tasks 1.1 and 1.2, respectively. We merged the training set from SemEval subtask 1.1 (ST1.1) and the training set from SemEval 2018 subtask 1.2 (ST1.2) into only one training set (SE2018) to augment data and improve results as already done in previous experiments on this same dataset by other authors (Rotsztejn et al., 2018). Then, we train the SemEval 2018 ST1.1 using the training set SE2018 and evaluating on the test set from ST1.1. For training SemEval 2018 ST1.2 task, we used SE2018 and evaluated using the test set from SemEval 2018 ST1.2.

Relation	Training frequency	Test frequency	Total per relation
Sub-task 1.1			
Usage	483/39.3%	175/49.3%	658/41.6%
Topic	18/1.5%	3/0.9%	21/1.3%
Model-Feature	326/26.5%	66/18.5%	392/24.8%
Part-Whole	234/19.1%	70/19.7%	304/19.2%
Compare	95/7.7%	21/6.0%	116/7.3%
Result	72/5.9%	20/5.6%	92/5.8%
Total	1228	355	1583
Sub-task 1.2			
Usage	470/37.7%	123/34.6%	593/37.0%
Topic	243/19.5%	69/19.4%	312/19.5%
Model-Feature	175/14.0%	75/21.1%	250/15.6%
Part-Whole	196/15.8%	56/15.8%	252/15.7%
Compare	41/3.2%	3/0.9%	44/2.7%
Result	123/9.8%	29/8.2%	152/9.5%
Total	1248	355	1603

Table 3: Distribution of annotated relations on SemEval 2018 Task-7

DDI Extraction 2013 *corpus* is a semantically annotated corpus of documents describing drug-to-drug interactions from 792 texts selected from DrugBank database, and 223 MedLine abstracts. There are 18.502 pharmacological substances and 5.028 DDIs, including pharmacokynetics (PK) and pharmacodynamics (PD) interactions. DDI 2013 *corpus* used micro-F1 metric to evaluate the models submitted to the shared task challenge. Table 4 displays the relations in this dataset.

Relation	Training frequency	Test frequency	Total per relation
Effect	1,687/41.9%	360/36.8%	2,047/41.0%
Mechanism	1,319/32.8%	302/30.9%	1,621/32.4%
Advice	826/20.6%	221/22.5%	1,047/20.9%
Int	188/4.7%	96/9.8%	284/5.7%
Total	4,020	979	4,999

Table 4: Distribution of annotated relations on DDI 2013 Corpus.

4.2. Experiments

We performed several experiments to find the best hyperparameters values for all datasets using the Google Colab Pro+ platform. Figure 3 shows the simple Deep Learning architecture, for illustration purposes, used in the experiments. To obtain the best combination of Hyperparameters (Table 6) and Preprocessing type (Table 7) for each dataset, we performed 50 trials in each dataset. These experiments found the best hyperparameters for each model using the *Optuna* framework (Table 8).

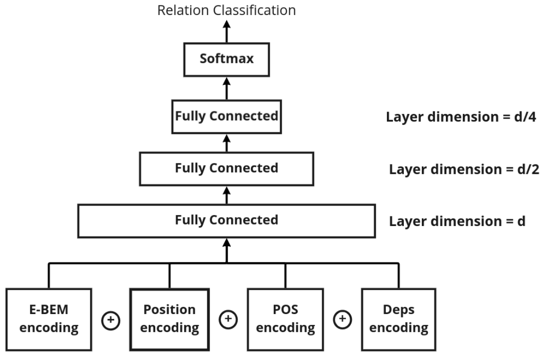

Figure 3: Experimental architecture used in the experiments.

The following code shows an example of the code necessary to run a training session in DeepREF:

```
import json
from opennre import config
from opennre.framework.train import
    Training

with open(config.
    BEST_HPARAMS_FILE_PATH.format('
    semeval2010'), 'r') as f:
     best_hparams = json.load(f)

train = Training('semeval2010', '
    micro_f1', best_hparams)
train.train()
```

Notice that one needs to write some few lines of code to train different models considering distinct experimentation settings (datasets, preprocessing types, and embedding types). As opposed to OpenNRE, the setup of the hyperparameters is done outside the code in a *json* file, as illustrated by the following code:

```
{
    "model": "bert",
    "pretrain": "bert-base-uncased",
    "preprocessing": [],
    "batch_size": 16,
    "lr": 2e-05,
    "max_length": 128,
    "max_epoch": 3,
    "position_embed": 0,
    "pos_tags_embed": 0,
    "deps_embed": 0,
    "sk_embed": 0
}
```

These are the neural network model, embedding, and hyperparameters employed for training in DeepREF.

4.3. Results and Discussion

Tables 9-12 summarize the results with and without hyperparameters optimization on SemEval 2010, SemEval 2018 (1.1), SemEval 2018 (1.2) and DDI 2013, respectively. All datasets evaluated yield better on punctuation and brackets/parenthesis removal preprocessing because such preprocessing steps remove noisy information from the sentences. The parenthesis removal only performed better on SemEval 2018 (1.1) and (1.2) because they have many sentences in which the entities are enclosed by parenthesis. DDI 2013 benefits most from entity blinding preprocessing because this type of preprocessing replaces the entities in a sentence to a more general word as ”DRUG”. Thus, the model has the potential to achieve improved results on unseen data. The position embedding yielded better performance for all datasets evaluated. Table 5 summarizes the ablation study on each dataset.

Dataset	Features	Micro-F1
DDI	E-BEM	91.44
	E-BEM+p	92.48
	E-BEM+p+pos	93.00
	E-BEM+p+pos+deps	**93.76**
SemEval 2010	E-BEM	87.28
	E-BEM+p	**88.50**
	E-BEM+p+pos	88.23
	E-BEM+p+pos+deps	88.14
Dataset	**Features**	**Macro-F1**
SemEval 2018 T1.1	E-BEM	81.87
	E-BEM+p	77.52
	E-BEM+p+pos	**83.88**
	E-BEM+p+pos+deps	80.00
SemEval 2018 T1.2	E-BEM	88.22
	E-BEM+p	**89.36**
	E-BEM+p+pos	87.71
	E-BEM+p+pos+deps	89.35

Table 5: Ablations studies for SemEval 2010, SemEval 2018 and DDI datasets using E-BEM without optimized hyperparameters. p = position embeddings; pos = POS tags embeddings; deps = dependency labels embeddings.

Hyperparameter	Default value	Distribution
max epoch	3	2-8
batch size	16	2-64
learning rate	2e-5	{1e-6, 0.1}
sentence length	128	32-256

Table 6: Hyperparameter distributions for hyperparameter optimization. The distribution written with {} means that the distribution is continuous. The distribution that has a “-” means that is a discrete distribution between some value to another inclusively. The batch size is different from OpenNRE’s default value due to Google Colab Pro+ RAM limitation used in our experiments.

dataset	Preprocessing type
SemEval 2010	d
SemEval 2018 (1.1)	b+p
SemEval 2018 (1.2)	sw+b+p
DDI	b+d+eb

Table 7: The best preprocessing type combination for each dataset evaluated. d = digit blinding; b = brackets or parenthesis removal; p = punctuaction removal; sw = stopwords removal; eb = entity blinding.

Hyperparameter	Best combination found
SemEval 2010 Task-8	
max epoch	14
batch size	8
sentence length	252
learning rate	2.220831734001225e-5
SemEval 2018 Task-7 subtask 1.1	
max epoch	5
batch size	2
sentence length	47
learning rate	9.720119898417658e-6
SemEval 2018 Task-7 subtask 1.2	
max epoch	8
batch size	7
sentence length	67
learning rate	2.7978590722954112e-5
DDI Extraction 2013	
max epoch	2
batch size	6
sentence length	243
learning rate	1.911610851819328e-06

Table 8: Best combination of hyperparameters after the hyperparameter optimization with 50 trials in each dataset.

A closer look at the results shows that, for all datasets, DeepREF either outperformed the other compared systems or had competitive results, except on the SemEval 2010 dataset. However, when focusing only on the frameworks, DeepREF generated the best learning models on the SemEval 2010 dataset (compared with OpenNRE and REflex), and DDI (compared with REflex). In general, the simple deep learning architecture (MLP) showed encouraging results since it was the best model on the SemEval 2018 (substasks 1.1 and 1.2) and DDI datasets, suggesting that the optimized learning process is quite effective.

On the other hand, the best performance systems on the SemEval 2010 dataset were implemented using more complex deep neural networks (DNN) such as CNN, GNN, and RNN with attention mechanism.

In conclusion, the embeddings proposed by DeepREF show evidence of being effective compared to many state-of-the-art relation classification systems.

Model	Micro-F1
QA (Cohen et al., 2021)	**91.90**
RIFRE (Zhao et al., 2021)	91.30
REDN (Li and Tian, 2020)	91.00
Skeleton-Aware BERT (Tao et al., 2019)	90.36
BERT-EM+MTB (Baldini Soares et al., 2019)	89.50
BERT-EM (Baldini Soares et al., 2019)	89.20
E-BEM Optimized (Ours)	88.62
E-BEM (Ours)	88.50
BERT-EM - OpenNRE (Han et al., 2019)	88.30
BERT-tokens - REflex (Chauhan et al., 2019)	86.69

Table 9: Performance results in terms of Micro-F1 of the models on SemEval 2010 Task-8 dataset.

Model	Macro-F1
E-BEM Optimized (Ours)	**84.56**
E-BEM (Ours)	83.88
ETH-DS3Lab (Rotsztejn et al., 2018)	81.72
UWNLP (Luan et al., 2018)	78.90
SIRIUS-LTG-UiO (Nooralahzadeh et al., 2018)	76.70
DMIR (Hettinger et al., 2018)	74.89
Talla (Pratap et al., 2018)	74.20

Table 10: Performance results in terms of Macro-F1 of the models on SemEval 2018 Task-7 subtask 1.1 dataset.

Model	Macro-F1
E-BEM Optimized (Ours)	**91.75**
ETH-DS3Lab (Rotsztejn et al., 2018)	90.40
E-BEM (Ours)	89.36
Talla (Pratap et al., 2018)	84.80
SIRIUS-LTG-UiO (Nooralahzadeh et al., 2018)	83.20
MIT-MEDG (Jin et al., 2018)	80.60
GU IRLAB (MacAvaney et al., 2018)	78.90

Table 11: Performance results in terms of Macro-F1 of the models on SemEval 2018 Task-7 subtask 1.2 dataset.

Model	Micro-F1
E-BEM Optimized (Ours)	**94.52**
E-BEM (Ours)	93.76
BERT-tokens - REflex (Chauhan et al., 2019)	91.31
DESC+MOL+SciBERT (Asada et al., 2020)	84.08
SciFive-large (Phan et al., 2021)	83.67
CharacterBERT (Boukkouri et al., 2020)	80.60
MOL+CNN (Asada et al., 2018)	72.55

Table 12: Performance results in terms of Micro-F1 of the models on the DDI 2013 dataset.

5. Conclusion and Future Work

Preprocessing the dataset, choosing the right embeddings and the right hyperparameter setting have a great influence on the quality of classification models. DeepREF proposes modules for all these tasks. It allows the addition of new datasets, the addition of embeddings, as well as an option to easily pretrain BERT weights, due to its modularity.

DeepREF yields better results compared to the other two frameworks, OpenNRE and REflex. Its performance results are also comparable to those of state-of-the-art models on several datasets, such as DDI, and SemEval 2018 Task 7.

As future work, we have the following development axes: a) integrating more datasets like TACRED (Zhang et al., 2017) into the framework; b) integrating other embeddings; c) considering more relation types, such as spatial relations (SpRL) or hypernym; d) dealing with bag-level relations, i.e., relations between entities belonging in different sentences; and, e) improving framework architecture.

6. Bibliographical References

Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework.

Asada, M., Miwa, M., and Sasaki, Y. (2018). Enhancing drug-drug interaction extraction from texts by molecular structure information.

Asada, M., Miwa, M., and Sasaki, Y. (2020). Using drug descriptions and molecular structures for drug–drug interaction extraction from literature. *Bioinformatics*, 37(12):1739–1746, 10.

Baldini Soares, L., FitzGerald, N., Ling, J., and Kwiatkowski, T. (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, July. Association for Computational Linguistics.

Barrio, P., Simões, G., Galhardas, H., and Gravano, L. (2014). Reel: A relation extraction learning framework. In *IEEE/ACM Joint Conference on Digital Libraries*, pages 455–456.

Bodenreider, O. (2004). The unified medical language system (umls): integrating biomedical terminology. *Nucleic Acids Research*, 32(90001).

Boukkouri, H. E., Ferret, O., Lavergne, T., Noji, H., Zweigenbaum, P., and Tsujii, J. (2020). Characterbert: Reconciling elmo and bert for word-level open-vocabulary representations from characters.

Chauhan, G., McDermott, M. B., and Szolovits, P. (2019). REflex: Flexible framework for relation extraction in multiple domains. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 30–47, Florence, Italy, August. Association for Computational Linguistics.

Cohen, A. D., Rosenman, S., and Goldberg, Y. (2021). Relation classification as two-way span-prediction.

Christiane Fellbaum, editor. (1998). *WordNet: an electronic lexical database*. MIT Press.

Gábor, K., Buscaldi, D., Schumann, A.-K., QasemiZadeh, B., Zargayouna, H., and Charnois, T. (2018). SemEval-2018 task 7: Semantic relation extraction and classification in scientific papers. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 679–688, New Orleans, Louisiana, June. Association for Computational Linguistics.

Han, X., Gao, T., Yao, Y., Ye, D., Liu, Z., and Sun, M. (2019). OpenNRE: An open and extensible toolkit for neural relation extraction. 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): System Demonstrations*, pages 169–174, Hong Kong, China, November. Association for Computational Linguistics.

Hendrickx, I., Kim, S. N., Kozareva, Z., Nakov, P., Séaghdha, D. O., Padó, S., Pennacchiotti, M., Romano, L., and Szpakowicz, S. (2019). Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals.

Herrero-Zazo, M., Segura-Bedmar, I., Martínez, P., and Declerck, T. (2013). The ddi corpus: An annotated corpus with pharmacological substances and drug–drug interactions. *Journal of Biomedical Informatics*, 46(5):914–920.

Hettinger, L., Dallmann, A., Zehe, A., Niebler, T., and Hotho, A. (2018). ClaiRE at SemEval-2018 task 7: Classification of relations using embeddings. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 836–841, New Orleans, Louisiana, June. Association for Computational Linguistics.

Jin, D., Dernoncourt, F., Sergeeva, E., McDermott, M., and Chauhan, G. (2018). MIT-MEDG at SemEval-2018 task 7: Semantic relation classification via convolution neural network. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 798–804, New Orleans, Louisiana, June. Association for Computational Linguistics.

Jin, W., Cai, Y., Kazman, R., Zheng, Q., Cui, D., and Liu, T. (2019). Enre: A tool framework for extensible entity relation extraction. In *2019 IEEE/ACM 41st International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*, pages 67–70.

Li, C. and Tian, Y. (2020). Downstream model design of pre-trained language model for relation extraction task.

Loper, E. and Bird, S. (2002). Nltk: The natural language toolkit. *CoRR*, cs.CL/0205028.

Luan, Y., Ostendorf, M., and Hajishirzi, H. (2018). The UWNLP system at SemEval-2018 task 7: Neural relation extraction model with selectively incorporated concept embeddings. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 788–792, New Orleans, Louisiana, June. Association for Computational Linguistics.

MacAvaney, S., Soldaini, L., Cohan, A., and Goharian, N. (2018). GU IRLAB at SemEval-2018 task 7: Tree-LSTMs for scientific relation classification. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 831–835, New Orleans, Louisiana, June. Association for Computational Linguistics.

Muzaffar, A., Azam, F., and Qamar, U. (2015). A relation extraction framework for biomedical text using hybrid feature set. *Computational and Mathematical Methods in Medicine*, 2015, 08.

Nastase, V., Nakov, P., Saghdha, D., and Szpakowicz, S. (2013). *Semantic Relations Between Nominals*. Morgan & Claypool Publishers.

Nguyen, T. H. and Grishman, R. (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, June. Association for Computational Linguistics.

Nooralahzadeh, F., Øvrelid, L., and Lønning, J. T. (2018). SIRIUS-LTG-UiO at SemEval-2018 task 7: Convolutional neural networks with shortest dependency paths for semantic relation extraction and classification in scientific papers. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 805–810, New Orleans, Louisiana, June. Association for Computational Linguistics.

Pawar, S., Palshikar, G. K., and Bhattacharyya, P. (2017). Relation extraction : A survey.

Phan, L. N., Anibal, J. T., Tran, H., Chanana, S., Bahadroglu, E., Peltekian, A., and Altan-Bonnet, G. (2021). Scifive: a text-to-text transformer model for biomedical literature.

Pratap, B., Shank, D., Ositelu, O., and Galbraith, B. (2018). Talla at SemEval-2018 task 7: Hybrid loss optimization for relation classification using convolutional neural networks. In *Proceedings of The 12th International Workshop on Semantic Evalua-*

tion, pages 863–867, New Orleans, Louisiana, June. Association for Computational Linguistics.

Rotsztejn, J., Hollenstein, N., and Zhang, C. (2018). ETH-DS3Lab at SemEval-2018 task 7: Effectively combining recurrent and convolutional neural networks for relation classification and extraction. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 689–696, New Orleans, Louisiana, June. Association for Computational Linguistics.

Takanobu, R., Zhang, T., Liu, J., and Huang, M. (2019). A hierarchical framework for relation extraction with reinforcement learning. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19. AAAI Press.

Tao, Q., Luo, X., and Wang, H. (2019). Enhancing relation extraction using syntactic indicators and sentential contexts.

Yu, T. and Zhu, H. (2020). Hyper-parameter optimization: A review of algorithms and applications.

Zeng, D., Liu, K., Lai, S., Zhou, G., and Zhao, J. (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, August. Dublin City University and Association for Computational Linguistics.

Zeng, D., Liu, K., Chen, Y., and Zhao, J. (2015). Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1753–1762, Lisbon, Portugal, September. Association for Computational Linguistics.

Zhang, Y., Zhong, V., Chen, D., Angeli, G., and Manning, C. D. (2017). 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, September. Association for Computational Linguistics.

Zhao, K., Xu, H., Cheng, Y., Li, X., and Gao, K. (2021). Representation iterative fusion based on heterogeneous graph neural network for joint entity and relation extraction. *Knowledge-Based Systems*, 219:106888.