

OpenNRE: An Open and Extensible Toolkit for Neural Relation Extraction

Xu Han, Tianyu Gao, Yuan Yao, Demin Ye, Zhiyuan Liu, Maosong Sun

October 2020

Introduction

OpenNRE is an open-source and extensible toolkit that provides a unified framework to implement neural models for relation extraction (RE).

OpenNRE provides various functional RE modules based on both TensorFlow and PyTorch to maintain sufficient modularity and extensibility, making it becomes easy to incorporate new models into the framework.

Application Scenarios

- sentence-level RE;
- bag-level RE;
- document-level RE;
- few-shot RE.

NER
Ernest Miller Hemingway was an American journalist \Rightarrow *Ernest Miller Hemingway was an American journalist*

Sentence-level RE
Ernest Hemingway was raised in Oak Park, Illinois \Rightarrow [*Ernest Hemingway*] $\xrightarrow{\text{place of birth}}$ [*Oak Park, Illinois*]

Bag-level RE
 In 1921, *Ernest Hemingway* married *Hadley Richardson*, the first of his four wives
Hadley Richardson was the first wife of American author *Ernest Hemingway*

 \Rightarrow [*Ernest Hemingway*] $\xrightarrow{\text{spouse}}$ [*Hadley Richardson*]

Document-level RE
Mark Twain and Olivia Langdon corresponded throughout 1868. She rejected his first marriage proposal, but they were married in Elmira, New York in February 1870. Then, Twain owned a stake in the Buffalo Express newspaper and worked as an **editor** and **writer**. While they were living in **Buffalo**, their son **Langdon** died of diphtheria at the age of 19 months. They had three daughters: **Susy Clemens**, **Clara Clemens**, and **Jean Clemens**.

Few-shot RE

Supporting Set

- London* is the capital of the *U.K* (capital of)
- Washington* is the capital of the *U.S.A* (capital of)
- Newton* served as the president of *the Royal Society* (member of)
- Leibniz* was a member of *the Prussian Academy of Sciences* (member of)
- Samuel Langhorne Clemens*, better known by his pen name *Mark Twain* (birth name)
- Alexei Maximovich Peshkov*, primarily known as *Maxim Gorky* (birth name)

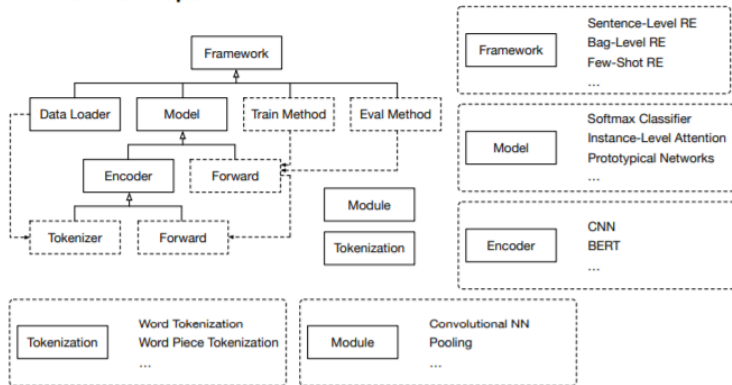
Query

- Euler* was elected a foreign member of *the Royal Swedish Academy of Sciences* (?)

Figure 1: The examples of all application scenarios in OpenNRE.

Toolkit Design and Implementation

Architecture of OpenNRE



Experiment and Evaluation

Two datasets of sentence-level RE were used: SemEval 2010 Task-8 and Wiki80.

Model	Wiki80	SemEval
CNN	63.93	71.11
BERT	84.57	84.02
BERT-Entity	86.61	84.21

Table 1: Accuracies of various models on Wiki80 and SemEval 2010 Task-8 under the single sentence setting.

Model	F1	F1 (*)
BERT	0.880	-
BERT-Entity	0.883	0.892

Table 2: Micro F1 scores of various models on SemEval 2010 Task-8 under the sentence-level RE setting. “(*)” indicates the original results from Soares et al. (2019).

“BERT” refers to using entity markers in input and taking [CLS] as output.

“BERT-Entity” refers to using entity markers in input and taking entity start as output.

Implementation

Clone the git repository, install and upload the library OpenNRE

```
[1] !git clone https://github.com/thunlp/OpenNRE.git
```

```
[2] from google.colab import drive  
drive.mount('/content/drive/', force_remount=True)
```

```
[3] %cd '/content/OpenNRE'  
!pip install -r requirements.txt
```

```
[4] !python setup.py install
```

```
[5] import opennre
```

Implementation

Select model and upload it

- `wiki80_cnn_softmax`: trained on wiki80 dataset with a CNN encoder.
- `wiki80_bert_softmax`: trained on wiki80 dataset with a BERT encoder.
- `wiki80_bertentity_softmax`: trained on wiki80 dataset with a BERT encoder (using entity representation concatenation).
- `tacred_bert_softmax`: trained on TACRED dataset with a BERT encoder.
- `tacred_bertentity_softmax`: trained on TACRED dataset with a BERT encoder (using entity representation concatenation).

```
[6] model = opennre.get_model('wiki80_cnn_softmax')
```

```
↳ 2020-10-13 05:58:24,490 - root - INFO - Initializing word embedding with word2vec.
```

You can do relation extraction with the following format:

```
[7] model.infer({'text': 'He was the son of Máel Dúin mac Máele Fithrich, and grandson of the high king Áed Uaridnach (died 612)',  
               'h': {'pos': (18, 46)}, 't': {'pos': (78, 91)}})
```

```
↳ ('father', 0.7500484585762024)
```

The `infer` function takes one dict as input. The `text` key represents the sentence and `h` / `t` keys represent head and tail entities, in which `pos` (position) should be specified.

The model will return the predicted result as: `('father', 0.7500484585762024)`

Implementation

I upload and tested other models

```
[8] model11 = opennre.get_model('wiki80_bert_softmax')
```

```
[9] model12 = opennre.get_model('wiki80_bertentity_softmax')
```

```
[10] model13 = opennre.get_model('tacred_bert_softmax')
```

```
[11] model14 = opennre.get_model('tacred_bertentity_softmax')
```


Implementation

```
m0 = model.infer({'text': "Donald J. Trump is the 45th President of the United States.", 'h': {'pos': (0, 15)}, 't': {'pos': (45, 58)}})
m01 = model.infer({'text': "Donald J. Trump is the 45th President of the United States.", 'h': {'pos': (45, 58)}, 't': {'pos': (0, 15)}})
m1 = model.infer({'text': "Donald J. Trump is the 45th President of the United States.", 'h': {'pos': (0, 15)}, 't': {'pos': (45, 58)}})
m12 = model1.infer({'text': "Donald J. Trump is the 45th President of the United States.", 'h': {'pos': (45, 58)}, 't': {'pos': (0, 15)}})
m2 = model2.infer({'text': "Donald J. Trump is the 45th President of the United States.", 'h': {'pos': (0, 15)}, 't': {'pos': (45, 58)}})
m21 = model2.infer({'text': "Donald J. Trump is the 45th President of the United States.", 'h': {'pos': (45, 58)}, 't': {'pos': (0, 15)}})
m3 = model3.infer({'text': "Donald J. Trump is the 45th President of the United States.", 'h': {'pos': (0, 15)}, 't': {'pos': (45, 58)}})
m31 = model3.infer({'text': "Donald J. Trump is the 45th President of the United States.", 'h': {'pos': (45, 58)}, 't': {'pos': (0, 15)}})
m4 = model4.infer({'text': "Donald J. Trump is the 45th President of the United States.", 'h': {'pos': (0, 15)}, 't': {'pos': (45, 58)}})
m41 = model4.infer({'text': "Donald J. Trump is the 45th President of the United States.", 'h': {'pos': (45, 58)}, 't': {'pos': (0, 15)}})
```

```
[21] print('{0:<30}{1:<40}{2:>10}'.format('wiki80_cnn_softmax:', "Donald J. Trump, United States", str(m0)))
print('{0:<30}{1:<40}{2:>10}'.format('wiki80_bert_softmax:', "Donald J. Trump, United States", str(m1)))
print('{0:<30}{1:<40}{2:>10}'.format('wiki80_bertentity_softmax:', "Donald J. Trump, United States", str(m2)))
print('{0:<30}{1:<40}{2:>10}'.format('taced_bert_softmax:', "Donald J. Trump, United States", str(m3)))
print('{0:<30}{1:<40}{2:>10}'.format('taced_bertentity_softmax:', "Donald J. Trump, United States", str(m4)))
print('{0:<30}{1:<40}{2:>10}'.format('wiki80_cnn_softmax:', "United States, Donald J. Trump", str(m01)))
print('{0:<30}{1:<40}{2:>10}'.format('wiki80_bert_softmax:', "United States, Donald J. Trump", str(m12)))
print('{0:<30}{1:<40}{2:>10}'.format('wiki80_bertentity_softmax:', "United States, Donald J. Trump", str(m21)))
print('{0:<30}{1:<40}{2:>10}'.format('taced_bert_softmax:', "United States, Donald J. Trump", str(m31)))
print('{0:<30}{1:<40}{2:>10}'.format('taced_bertentity_softmax:', "United States, Donald J. Trump", str(m41)))
```

```
❏ wiki80_cnn_softmax: Donald J. Trump, United States ('country of citizenship', 0.8769035935401917)
wiki80_bert_softmax: Donald J. Trump, United States ('country of citizenship', 0.7819002270698547)
wiki80_bertentity_softmax: Donald J. Trump, United States ('country of citizenship', 0.977445662021637)
taced_bert_softmax: Donald J. Trump, United States ('per:countries_of_residence', 0.5316286683082581)
taced_bertentity_softmax: Donald J. Trump, United States ('per:countries_of_residence', 0.7729834318161011)
wiki80_cnn_softmax: United States, Donald J. Trump ('head of government', 0.8833723664283752)
wiki80_bert_softmax: United States, Donald J. Trump ('head of government', 0.9970136880874634)
wiki80_bertentity_softmax: United States, Donald J. Trump ('head of government', 0.9920777678489685)
taced_bert_softmax: United States, Donald J. Trump ('org:top_members/employees', 0.6293102502828276)
taced_bertentity_softmax: United States, Donald J. Trump ('org:top_members/employees', 0.8308277726173401)
```

Online system

THUNLP

Relation Extraction

Supervised Relation Extraction

Few-Shot Relation Extraction

Bag-level Relation Extraction

Document Relation Extraction

Donald J. Trump is the 45th President of the United States.

Head United States

Tail Donald J. Trump

NER Model

TagMe

Entity Suggestion

Model

BERT

Example

Choose Example

Submit

United States | Donald J. Trump has a relation of head of government with the probability of 0.9980

Thank you for attention!