Azaad@BND at SemEval-2023 Task 2: How to Go from a Simple Transformer Model to a Better Model to Get Better Results in Natural Language Processing

Reza Ahmadi, Shiva Arefi, Mohammad Jafarabad
Bandar Abbas Islamic Azad University
{ Reza.zx@live.com, Shiva.Arefi@outlook.com, tcsms@yahoo.com }

Abstract

In this article, which was prepared for the sameval2023 competition (task number 2), information about the implementation techniques of the transformer model and the use of the pre-trained BERT model in order to identify the named entity (NER) in the English language, has been collected and also the implementation method is explained.

Finally, it led to an F1 score of about 57% for Fine-grained and 72% for Coarse-grained in the dev data.In the final test data, F1 score reached 50%.

1 Introduction

The purpose of this competition is to extract the named entity (Named-Entity Recognition) within a text.

NER is basically a token classification task where each token is classified into one or more predefined categories. For example, persons, locations, corporations, etc. should be extracted and classified from within the text. [1]

In this competition, the basis of classification is divided into two categories: general classification and partial classification. For example, each Location category has 4 sub-categories (subclassifications) including: Facility, OtherLOC, HumanSettlement, Station, and all classifications are available and accessible on the competition dataset page [2].
The total classifications are as follows:

```
all_tags={

'PER':['OtherPER','SportsManager','Cleric','Politician','Athlete','Artist','Scientist'],

'LOC':['Facility','OtherLOC','HumanSettlement','Station'],

'GRP':['MusicalGRP','PublicCorp','PrivateCorp','OtherCorp','AerospaceManufacturer','SportsGRP','CarManufacturer','TechCORP','ORG'],

'PROD':['OtherPROD','Drink','Food','Vehicle','Clothing'],

'CW':['VisualWork','MusicalWork','WrittenWork','ArtWork','Software','OtherCW'],

'MED':['Medication/Vaccine','MedicalProcedure','AnatomicalStructure','Symptom','Disease']
}
```

Table 1 - mapping of the tags

2 Related Work

This article is related to the previous year's SemEval-2022 Task 11 article [4], which is to identify the named entity in the general category [PER, LOC, CORP, GRP, PROD, CW].

About 55 teams have participated in this task and the best team achieved the score of F1=0.91 in the English language that we want. According to what is mentioned in the article, "most of the teams have used external databases such as Wikipedia, Gazetteer. Also, they have been more interested in the pre-trained XLM-RoBERTa model."

Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023), pages 2184–2187

3 Data

The data set used for training and developing the model is the same data that is provided in the Dataset section [1] of the competition, which are labeled with the CoNLL format and the entities are labeled with the IOB method.

IOB: Inside–outside–beginning (tagging) is a common tagging for tagging tokens in computational linguistics .The B- prefix before a tag specify that the tag is the first of a chunk, and an I- prefix before a tag specify that the tag is inside a chunk. The B- tag specify that a tag is followed by a tag of the same type without O tokens between them. An O tag shows that a token belongs to no entity chunk.

After checking, due to the imbalance of the classes used in the training data, in order for the machine to understand more about the data and classifications, techniques such as Oversampling (increasing data with less density) should be used on the data in the pre-processing stage of the data. The balance of the categories could be maintained.

Also, in order to improve machine learning, we first calculated the ratio of the number of famous words to the length of the sentence, and according to the obtained ratio, we added the sentence to the dataset several time. Here, experimentally, if obtained ratio was more than 90%, 6 times were repeated, if the ratio was more than 80%, 3 times were repeated, and for a ratio greater than 65%, one repetition was sufficient.

4 Methodology

4.1 Transformers

Transformers are in many cases replacing convolutional and recurrent neural networks (CNN and RNN), which were the most popular types of deep learning models until five years ago. Like most neural networks, transformer models are essentially large encoder/decoder blocks that process the data.

The structure of an encoder layer in a transformer layer is such that each encoder consists of two separate sub-layers, the first layer is the attention layer and the second layer is a feedforward neural network.

The structure of a decoder class, like the encoder, consists of two layers: self-attention and FNN, with the difference that in the decoder there is an intermediate layer called encoder-decoder attention, which helps the machine not focus on the word being learned. Pay attention to related words.

The output of the decoder is a vector, so the last layer of the transformer needs to be a softmax layer, because this layer divides the values into the probability distribution that the output of each element of the vector is in the range of 0 to 1, and the sum of all these elements must be one.

Transformers use positional encoders to label data elements entering and leaving the network. Attention units follow these labels and compute some sort of algebraic map of how each element relates to the other elements. Attention queries are usually executed in parallel by computing a matrix of equations in what is called multi-headed attention.

In this project, a simple transformer model was first used for training based on general classification, which by increasing the number of layers of the transformer model and also better setting things such as the length of tokens, the number of attention heads, and the number of FFN layers, led to an F1 score of about 65%. became. Of course, this model did not give us good results for partial classification where the number of categories was about 36 categories. A better solution is to use a pre-trained esophageal transfer model such as BERT, RoBERTa, ALBERT, …

4.2 BERT

Bidirectional Encoder Representations from Transformers (Bidirectional Encoder Representations from Transformers) or BERT, which uses transformers, is an attention mechanism that learns the textual relationships between words (or subwords) in the text [3].

The BERT model is actually a group of transformer model encoders that have been studied. In the BERT base model, there are 12 like transformer blocks and 12 attention layers, and in this base model there are 768 hidden nodes in FFN. Fine-tuning method is used here to use this model. In this method, the input of the model is a

list of tokens with a length of 512 tokens. These tokens are passed through the 12 mentioned layers and at the end a vector It is returned as output with a length of 768.

4.3 Used in this project

In this project, the training was done using the TFAutoModelForTokenClassification model, and the keras library and the Adam method with a learning rate of 0.0001, as well as the loss function from the SparseCategoricalCrossentropy method available in the keras library's loss methods, were used to adjust the optimizer, which is in the next part of the output results. is brought

During the training learning process, the batch_size value was set to 40, that is, the machine divides the data set into 40 groups and updates the weights after learning each of these groups.
Considering that the total length of the categories was 17760, the number of categories was equal to 444.

Also, here the number of rounds of the learning process was chosen as 3.
The duration of each training process was about 1500 seconds, the specifications of a system on a virtual machine (VMware Workstation Pro) are:

- OS: Fedora Linux 36 (Workstation Edition)
- Memory: 12 GB
- Processor :11th Gen Intel® Core™ i7-1165G74
- Graphics: SVGA3D; build: RELEASE; LLVM;

5 Results

5.1 Dev data results

The results obtained for the fine-grained section are with precision=0.64, recall=0.54, F1=0.57, and also these results for the coarse-grained section are with precision=0.76, recall=0.68, F1=0.72.

Class	Precision	Recall	F1
Facility	0.625	0.7692	0.6897
OtherLOC	0.8333	0.3125	0.4545
HumanSettlement	0.8696	0.7339	0.796
Station	0.75	0.9	0.8182
VisualWork	0.7255	0.6066	0.6607
MusicalWork	0.6923	0.5902	0.6372
WrittenWork	0.8049	0.6111	0.6947
ArtWork	0.2667	0.3077	0.2857
Software	0.6154	0.6154	0.6154
OtherCW	0	0	0
MusicalGRP	0.6944	0.6757	0.6849
PublicCorp	0.5238	0.3929	0.449
PrivateCorp	0.8333	0.4545	0.5882
OtherCorp	0	0	0
AerospaceManufacturer	0.8889	0.8	0.8421
SportsGRP	0.85	0.8293	0.8395
CarManufacturer	0.625	0.7692	0.6897
TechCORP	0	0	0
ORG	0.6849	0.641	0.6623
Scientist	0.3333	0.2667	0.2963
Artist	0.7857	0.7783	0.782
Athlete	0.6739	0.7848	0.7251
Politician	0.6341	0.4906	0.5532
Cleric	0.4167	0.3333	0.3704
SportsManager	0.7778	0.4375	0.56
OtherPER	0.5057	0.4835	0.4944
Clothing	0.5	0.5	0.5
Vehicle	0.5789	0.55	0.5641
Food	0.5	0.4211	0.4571
Drink	0.7273	0.7273	0.7273
OtherPROD	0.5882	0.4082	0.4819
Medication/Vaccine	0.4444	0.6667	0.5333
MedicalProcedure	0.6667	0.4615	0.5455
AnatomicalStructure	0.375	0.3529	0.3636
Symptom	1	0.1	0.1818
Disease	0.4	0.2222	0.2857
Macro Average Performance	0.6421	0.5453	0.5706

Table 2 - dev data fine-grained Performance

Class	Precision	Recall	F1
LOC	0.8387	0.7919	0.8146
Medicine	0.619	0.5132	0.5612
PER	0.9266	0.8919	0.9089
PROD	0.6556	0.5413	0.593
CW	0.7676	0.6605	0.71
GRP	0.791	0.7294	0.7589
Macro Average Performance	0.7664	0.688	0.7244

Table 3 - dev data coarse-grained Performance

5.2 Test data results

The results obtained for the fine-grained section are with precision=0.53, recall=0.44, F1=0.47, and also these results for the coarse-grained section are with precision=0.73, recall=0.62, F1=0.67.

The results obtained according to the match log show that both on the Dev data and on the Test data, the best performance of the machine is PER and LOC classification, and the highest score is on the Medicine and PROD categories.

Class	Precision	Recall	F1
Facility	0.6138	0.5884	0.6008
OtherLOC	0.6049	0.2626	0.3662
HumanSettlement	0.8142	0.8361	0.825
Station	0.7384	0.6347	0.6826
VisualWork	0.6829	0.5121	0.5853
MusicalWork	0.7325	0.6444	0.6856
WrittenWork	0.5941	0.4928	0.5387
ArtWork	0.3786	0.2307	0.2867
Software	0.6728	0.5227	0.5883
MusicalGRP	0.5853	0.5773	0.5813
PublicCorp	0.5075	0.5149	0.5112
PrivateCorp	0.0264	0.0296	0.0279
AerospaceManufacturer	0.2597	0.3547	0.2999
SportsGRP	0.7593	0.7965	0.7775
CarManufacturer	0.436	0.307	0.3603
ORG	0.6009	0.5642	0.5819
Scientist	0.4428	0.3636	0.3993
Artist	0.7245	0.7637	0.7436
Athlete	0.7544	0.769	0.7617
Politician	0.6256	0.4957	0.5531
Cleric	0.5129	0.3857	0.4403
SportsManager	0.6479	0.6315	0.6396
OtherPER	0.4195	0.3885	0.4034
Clothing	0.5266	0.3788	0.4406
Vehicle	0.3937	0.2034	0.2682
Food	0.5852	0.2152	0.3146
Drink	0.2263	0.0467	0.0775
OtherPROD	0.4341	0.312	0.3631
Medication/Vaccine	0.6194	0.4311	0.5084
MedicalProcedure	0.5366	0.253	0.3439
AnatomicalStructure	0.5861	0.5406	0.5624
Symptom	0.0156	0.0102	0.0124
Disease	0.5797	0.4652	0.5162
Macro Average Performance	0.5345	0.4401	0.4742

Table 4 - Test data Fine-grained Performance

Class	Precision	Recall	F1
CW	0.7527	0.6056	0.6712
LOC	0.8096	0.7735	0.7911
GRP	0.6904	0.6763	0.6833
PER	0.8992	0.8762	0.8875
Medicine	0.6628	0.4903	0.5637
PROD	0.5897	0.3337	0.4262
Macro Average Performance	0.7341	0.626	0.6705

Table 5 - Test data coarse-grained Performance

6 Conclusion

Conclusion Using pre-trained models such as BERT can give us much better output (prediction) about an NLP problem, you just need to fine-tune the model and do a good pre-processing on the data to be closer to the final goal. Let's become and the work will go better. Of course, in the future, we plan to implement these tests on the BERT-large model, which requires more computing time.

References

[1] Besnik Fetahu, Sudipta Kar, Zhiyu Chen, Oleg Rokhlenko, and Shervin Malmasi. 2023b. SemEval2023 Task 2: Fine-grained Multilingual Named Entity Recognition (MultiCoNER 2). In Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023). Association for Computational Linguistics.

[2] Besnik Fetahu, Zhiyu Chen, Sudipta Kar, Oleg Rokhlenko, and Shervin Malmasi. 2023a. MultiCoNER v2: a Large Multilingual dataset for Finegrained and Noisy Named Entity Recognition.

[3] Deep Learning with Python 1st Edition - Francois Chollet – 2017.

[4] Shervin Malmasi, Anjie Fang, Besnik Fetahu, Sudipta Kar, and Oleg Rokhlenko. 2022. SemEval-2022 task 11: Multilingual complex named entity recognition (MultiCoNER). In Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), pages 1412–1437, Seattle, United States. Association for Computational Linguistics.