Project - Name Entity Recognition using BERT

Project - Named-Entity Recognition

Hoang Manh Truong

2024-02-09

Table of contents

1 Introduction **2**
- 1.1 Deep Transfer Learning . . . 2
- 1.2 BERT . . . 2
- 1.3 Dataset . . . 3
- 1.4 Metrics . . . 3

2 Training & Evaluation Pipelines **4**
- 2.1 Preprocessing . . . 4
- 2.2 Tokenization . . . 4
- 2.3 BERT Fine-tuning . . . 4

3 Analysis **4**
- 3.1 Results . . . 4
- 3.2 Discussion . . . 5

4 Conclusion **5**

1 Introduction

1.1 Deep Transfer Learning

The definition of deep transfer learning is as follows: given a learning task T_t based on D_t, and a related but different learning task T_s based on D_s, where D_s and D_t are the source and target domains, respectively, deep transfer learning aims to improve the learning of the target predictive function $f_t(\cdot)$ in T_t using the knowledge in D_s and T_s, where $D_s \neq D_t$ and/or $T_s \neq T_t$. Among the various deep transfer learning techniques, network-based deep transfer learning is the most widely used. It is based on the assumption that the source and target domains share the same feature space but have different marginal probability distributions. Recently, pre-trained language models, such as GPT and BERT, with large amounts of unlabeled data and fine-tuning in downstream tasks have made a breakthrough in NLP domain (Figure 1).

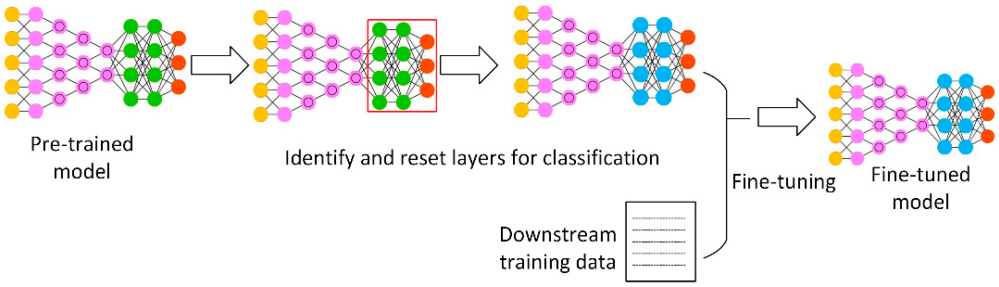

Figure 1: The typical process of network-based deep transfer learning.

1.2 BERT

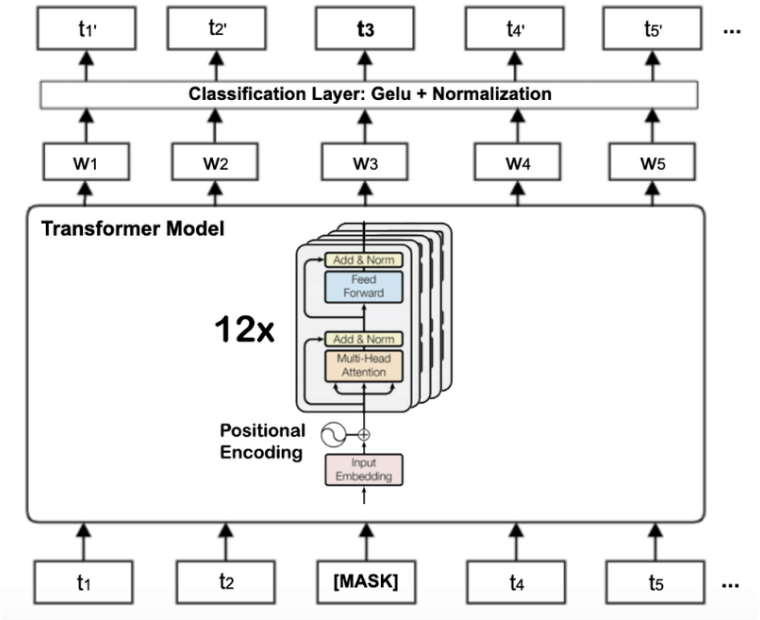

Figure 2: Overall base architecture of BERT with twelve encoder blocks.

Language models can be roughly categorized into N-gram language models and neural language models. While classical neural models, including Word2Vec, is still widely used today, BERT improves natural language pre-training by using mask-based objectives and a Transformer-based architecture (Figure 2), which has successfully improved many state-of-the-art results for various natural language tasks. Since more powerful models like GPT3 are not open-source and not available to public, BERT can be regarded as one of the best pre-trained language models for downstream tasks.

1.3 Dataset

There are 2 dataset provided for the project, the training dataset and the testing dataset:

Dataset	No. of words/labels	No. of sentences/phrases
Training	219552	23499
Testing	55042	5946

Each word (or token) is assigned a NER label that can be one of:

- **B-MISC**: Beginning of a miscellaneous entity that doesn't fall under standard categories (like person, organization, or location).
- **I-MISC**: Inside or continuation of a miscellaneous entity.
- **B-PER**: Beginning of a person's name.
- **I-PER**: Inside or continuation of a person's name. Used for multi-word names.
- **O**: Outside of any named entity.
- **B-LOC**: Beginning of a geographical location name.
- **I-LOC**: Inside or continuation of a geographical location name. Used for multi-word locations.
- **B-ORG**: Beginning of an organization name.
- **I-ORG**: Inside or continuation of an organization name. Used for multi-word organizations.

The **B-** prefix indicates the beginning of an entity, the **I-** prefix indicates that the word is inside an entity, and O indicates a word that is not part of a named entity.

1.4 Metrics

Since the task of NER belongs to the group of multi-label classification problems, the following metrics are used to evaluate the performance of the models:

- **Precision**: Precision measures the proportion of correctly identified named entities out of all entities the model identified. It is calculated using the formula:

$$Precision = \frac{TP}{TP + FP}$$

  where `TP` is the number of True Positives (correctly identified entities), and `FP` is the number of False Positives (incorrectly identified entities).

- **Recall**: Recall assesses the proportion of actual named entities that the model correctly identified. The formula for Recall is:

$$Recall = \frac{TP}{TP + FN}$$

  where `FN` is the number of False Negatives (entities that were not identified).

- **F1 Score**: The F1 Score is the harmonic mean of Precision and Recall, providing a balance between the two. It is particularly useful when the class distribution is uneven. F1 Score is calculated as:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

- **Accuracy**: Accuracy measures the overall correctness of the model across all classifications, calculated by:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

  where `TN` is the number of True Negatives (correctly identified non-entities). However, in NER, Accuracy is less informative due to the high number of true negatives (non-entity tokens), which can skew the metric.

2 Training & Evaluation Pipelines

2.1 Preprocessing

The data was ingested and preprocessed in several steps:

- **Verify and clean up the data**: training data and validation data are different, as one is separated by `;;;` and one is separated by a new line. Moreover, there are several samples that are mislabeled into "O O", when it should have been "O". In this case, we simply replace with the correct labels.
- **Read the data into sentences (or phrases)**: In order to tokenize and train the data effectively, it needs to be transformed into sentences or phrases. To do this, we use the comma and the default separator (`;;;` or `\n`) to group words that belong in the same sentence together.

2.2 Tokenization

The data is tokenized using the `bert-base-uncased` tokenizer from HuggingFace, which is a platform for building, training, and deploying machine learning models, particularly focusing on pre-trained models like transformers. With this, we can transform the sentence into even futher broken-down pieces of sub-words. For example, the sentence *"He said a proposal last month by EU Farm Commissioner Franz Fischler to ban sheep brains spleens and spinal cords from the human and animal food chains was a highly specific and precautionary move to protect human health "'* can be tokenized into:

```
['he', 'said', 'a', 'proposal', 'last', 'month', 'by', 'eu', 'farm', 'commissioner', 'franz', 'fis',
```

In this example, we can see that the word "precautionary" is split into 4 sub-words: `'pre', '##ca', '##ution', '##ary'`.

2.3 BERT Fine-tuning

We use the same model `bert-base-uncased` for fine-tuning of the NER training dataset. The fine-tuning process can be summarized as follows:

- Optimization Setup: We define the optimizer `AdamW` with the following parameters:
  - Learning rate: $5e-5$. We also set up a learning rate scheduler to adjust the learning rate based on the number of warmup steps and total training steps.
  - Adam ϵ: $1e-8$
  - Weight decay: 0.0
- Training Loop: Iterate over the dataset for 10 epochs. During each batch, the following steps are executed:
  - Forward pass: Compute the model's output and loss.
  - Backward pass: Compute the gradient of the loss with respect to model parameters.
  - Gradient Clipping: Clip gradients to a maximum norm to prevent exploding gradients.
  - Optimization Step: Update model parameters using the optimizer.
  - Learning Rate Scheduling: Update the learning rate based on the predefined schedule.
  - Zero the gradients to prepare for the next step.

3 Analysis

3.1 Results

After the fine-tuning process, we evaluate the model on the validation dataset. The following results are obtained in Table 2 and Table 3:

Table 2: Accuracy, Precision, Recall and F1-score on validation dataset.

Metric	Value
Accuracy	96.18%
Precision	78.03%

Metric	Value
Recall	82.56%
F1	80.23%

Table 3: Detailed Metrics by Class.

Class	Precision	Recall	F1-Score	Support
LOC	0.85	0.91	0.88	1837
MISC	0.87	0.83	0.85	922
ORG	0.59	0.67	0.63	1341
PER	0.82	0.85	0.83	1846

3.2 Discussion

The fine-tuning of BERT for Named Entity Recognition (NER) classification has yielded promising results, as evidenced by the metrics presented in the general and detailed results tables. Particularly, when looking at different classes, we can see that:

- **Location (LOC)** entities achieved high precision and recall, leading to an F1 score of 0.88. This indicates the model's strong capability in identifying geographical entities, likely due to distinct contextual and syntactical patterns associated with such entities.
- **Miscellaneous (MISC)** entities also showed strong performance with an F1 score of 0.85. The precision and recall balance suggests that the model is reasonably effective in identifying entities that do not fall into the more standard categories.
- **Organizations (ORG)** is not very high at 0.63, as the lowest precision and F1 score among the categories. This could be due to the diverse nature of organization names and possible overlaps with other entity types, indicating a need for model improvement in this area.
- **Person (PER)** names were well-recognized, with an F1 score of 0.83, reflecting the model's effectiveness in identifying individual names, possibly due to clear patterns and contextual cues.

4 Conclusion

In this project, we have applied deep transfer learning and fine-tuning BERT for the task of Name Entity Recognition. The fine-tuned BERT model demonstrates strong potential in NER tasks, with particularly impressive results in identifying LOC and PER entities. The relatively lower performance on ORG entities suggests an area for further model refinement. Future work could explore more advanced techniques for handling ambiguous entities, additional contextual features, or more sophisticated post-processing rules to improve precision and recall across all entity types.