Application of Fine-Tuning BERT for Named Entity Recognition (NER).

1. Introduction

Named Entity Recognition (NER) is a fundamental task in natural language processing (NLP) with many applications, from information retrieval to question answering and text summarization. In recent years, pre-trained language models like BERT (Bidirectional Encoder Representations from Transformers) have revolutionized NER by providing a robust foundation for extracting named entities from text. Fine-tuning BERT for NER tasks has become a prominent approach, enabling models to achieve state-of-the-art results across various domains and languages.

This exploration delves into the exciting domain of NER and its intersection with fine-tuning BERT, showcasing how this combination of advanced techniques has led to remarkable advancements in entity recognition. By harnessing BERT's contextual understanding and linguistic capabilities, NER models can accurately discern and classify named entities, facilitating more refined information extraction and knowledge discovery from textual data.

2. Strategies

The following strategies are going to help me keep the track of this project:

1. Establish an objective: Gain a deep understanding of your dataset, including its distribution, class imbalance, and any specific challenges related to your task.
2. Collect and preprocess data: Find the suitable dataset to help you achieve your task. Carefully preprocess your data, including cleaning, tokenization, and handling missing values, to ensure it is ideal for the model.
3. Select a pre-trained model: Choose a pre-trained model architecture well-suited for your specific task, whether NLP, computer vision, or another domain.
4. Data splitting: Divide your data into training, validation, and test sets. Ensure that the splits are representative and maintain class balance.
5. Validation of the model: Make sure that your fine-tuned model has good quality testing metrics such as training loss, training token accuracy, test loss, and test token accuracy. The previous statistics give you an idea of how good your model is.
6. Documentation: Maintain detailed documentation of experiments, including hyperparameters, results, and insights, using tools like Jupyter notebooks or experiment tracking platforms.

3. Methodology

Regarding the strategies mentioned previously, it is essential to note that until now, we can establish our objective and identify which dataset is the best for doing our approach and which pre-trained model we can use for fine-tuning. The other strategies can be shown during the development of this approach and can be more dependent on the results.

3.1. Objective

To fine-tune the BERT base model (uncased) using the CoNLL-2003 dataset for named entity recognition (NER). The goal is to adapt the pre-trained BERT model to identify and classify name entities in text data accurately.

3.2. Dataset.

I plan to use the CoNLL-2003, published in the Hugging Face repository. You can find it here: https://huggingface.co/datasets/conll2003.

id (string)	tokens (sequence)	pos_tags (sequence)	chunk_tags (sequence)	ner_tags (sequence)
"0"	["EU", "rejects", "German", "call", "to"…	[22, 42, 16, 21, 35, 37,…	[11, 21, 11, 12, 21, 22,…	[3, 0, 7, 0, 0, 0, 7, 0, 0…
"1"	["Peter", "Blackburn"]	[22, 22]	[11, 12]	[1, 2]
"2"	["BRUSSELS", "1996-08-22"]	[22, 11]	[11, 12]	[5, 0]
"3"	["The", "European", "Commission", "said",…	[12, 22, 22, 38, 15, 22,…	[11, 12, 12, 21, 13, 11,…	[0, 3, 4, 0, 0, 0, 0, 0, 0…
"4"	["Germany", "'s", "representative", "to"…	[22, 27, 21, 35, 12, 22,…	[11, 11, 12, 13, 11, 12,…	[5, 0, 0, 0, 0, 3, 4, 0, 0…
"5"	["\"", "We", "do", "n't", "support",…	[0, 28, 41, 30, 37, 12,…	[0, 11, 21, 22, 22, 11,…	[0, 0, 0, 0, 0, 0, 0, 0, 0…
"6"	["He", "said", "further",…	[28, 38, 16, 16, 21, 38,…	[11, 21, 11, 12, 12, 21,…	[0, 0, 0, 0, 0, 0, 0, 0, 0…

< Previous 1 2 3 ... 141 Next >

Figure 1. Sample viewer of ConLL-2003 train dataset

CoNLL-2003 (see Figure 1) is a famous dataset for named entity recognition. The dataset contains three files:

- Train (14k rows)
- Validation (3.25k rows)
- Test (3.45k rows)

Each file contains four columns separated by a single space.

- Tokens: Each word that contains the sentence
- POS: is the part of speech
- Chunk: a group of consecutive terms or tokens in a sentence
- NER: name entity recognition

I chose this dataset because it is famous for training and understanding how to use NER better. Also, it gives me an idea of which components we need for further works related to Name Entity Recognition. The objective of this model is to label into four tags: persons, locations, organizations, and names of miscellaneous entities that do not belong to the previous three groups.

3.3. Fine-tune a pre-trained model.

I plan to use a BERT-based model (uncased) because it is one of the most popular models in the Hugging Face repository, with around 40M downloads. You can find it here: https://huggingface.co/bert-base-uncased.

Bert base model is a pre-trained model on the English language using a masked language modeling (MLM) objective, which means that this model can predict the future token depending on the sentence. For example, “Paris is the [MASK] of France,” and the model gives a probability of which token is appropriate to replace [MASK] token: capital - 0.997, heart – 0.001, center – 0.000, centre – 0.000, city – 0.000. (See Figure 2)

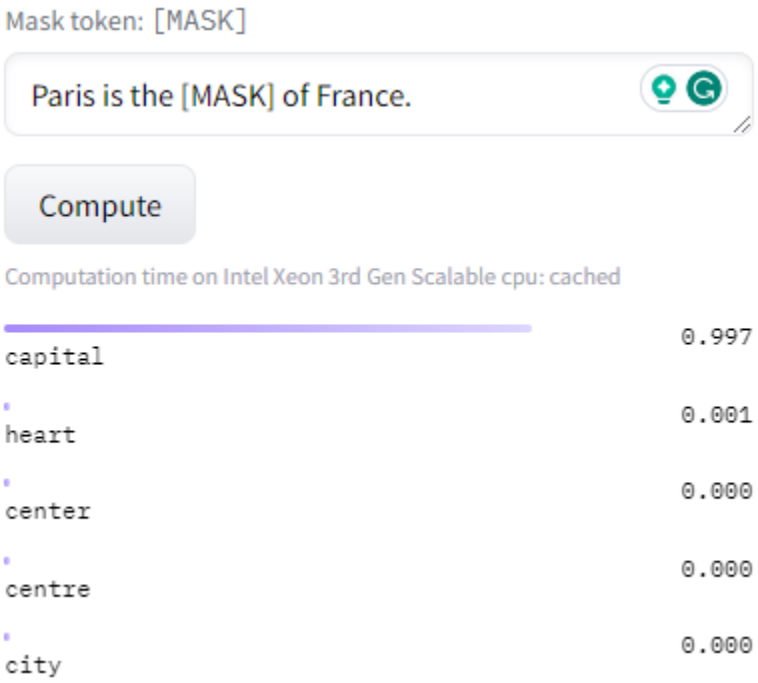

Figure 2. Example of BERT-based model (uncased)

This model will be fine-tuned to predict or identify which token belongs to an entity. For example, the sentence “My name is Wolfgang, and I live in Berlin.” can be identified by the tokens: Wolfgang as PERSON and Berlin as LOCATION (see Figure 3).

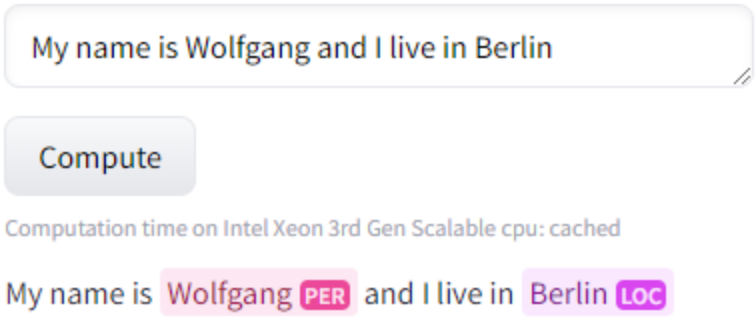

Figure 3. Example of the fine-tuned model

4. Future work

Once this approach is done, we must continue this work and develop tools that help our research team in the different projects. For this reason, I would like to propose a couple of mini-projects:

1. Explore using generative text to create synthetic data (improving Thomas's approach) using Llama, GPT, and Falcon, better structuring the code, and understanding each parameter to optimize hyperparameters.
2. Create an interface (like Hugging Face) using Streamlit, where we can use the different NLP models and a chatbot to integrate into Referencer.
3. Develop a model which can identify actors, factors, and mechanisms. I have an idea that if we can use sentences and with the POS or NER, we can create some rules such as:
    a. An actor can be a noun (POS) or a person, location, or organization (NER)
    b. A factor can be an adjective (POS)
    c. A mechanism can be a verb (POS)

    Or, to create our dataset, we can use packages such as Spicy or Stanza to identify tokens, POS, and chunks and manually label the NER tag.

5. Deadline

I expect to finish fine-tuning on **Thursday, September 7.**