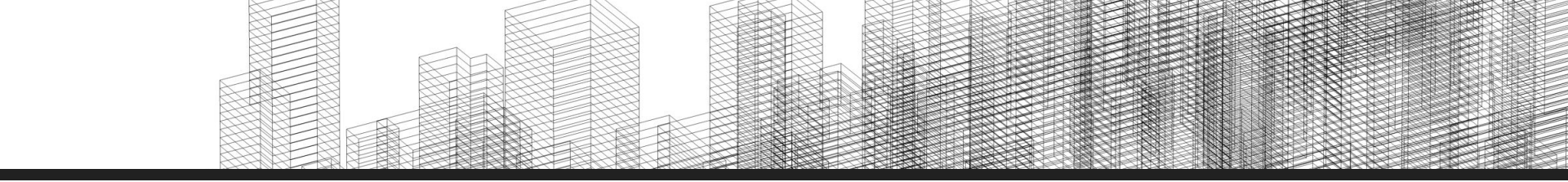


Named Entity Recognition

CPSC - 7373 Artificial Intelligence
Final Project Presentation
Veronika Gudipati

Introduction

- Global data creation is projected to grow to more than 180 zettabytes by 2025.
- Information extraction is the process of extracting information from unstructured textual sources to enable finding entities as well as classifying and storing them in a database.

Problem

Identifying predefined entities in the input text

Example:

Luke Rawlence joined Aiimi as a data scientist in Milton Keynes, after finishing his computer science degree at the University of Lincoln.

Goal

Luke Rawlence joined Aiimi as a data scientist in Milton Keynes, after finishing his computer science degree at the University of Lincoln.

NAMED ENTITY RECOGNITION

Named Entity Recognition

- Subtask of information extraction
- Locate and classify named entities mentioned in unstructured text into pre-defined categories
- These entities can be various things from a person to something very specific like a biomedical term

Dataset

- CoNLL2003 is one of the most evaluated English NER datasets
- The data was taken from the Reuters Corpus
- Preprocessed data
- 3 files for train, test and validation

Dataset

CoNLL2003 dataset concentrates on four types of named entities:

1. Person (PER)
2. Location (LOC)
3. Organizations (ORG) and
4. Misc (names of miscellaneous entities that do not belong to the previous three groups.)

CoNLL - 2003 data format

- IOB tagging scheme is used
- Parts - Of - Speech (POS) tag
- NER tags are used for classifying and recognizing the entities.
- ORG, PER, LOC, MISC, O are the tags used for Organization, Person, Location, Miscellaneous, Other(word is not an entity)

WORD	POS	CHUNK	NER
U.N.	NNP	I-NP	I-ORG
official	NN	I-NP	O
Ekeus	NNP	I-NP	I-PER
heads	VBZ	I-VP	O
for	IN	I-PP	O
Baghdad	NNP	I-NP	I-LOC
.	.	O	O

IOB tagging

- IOB - inside, outside, beginning
- I-TYPE =>the word is inside a phrase of type TYPE.
- B-TYPE => beginning of a new phrase of type TYPE

```
Alex B-PER
is O
going O
to O
Los B-LOC
Angeles I-LOC
in O
California B-LOC
```

Model

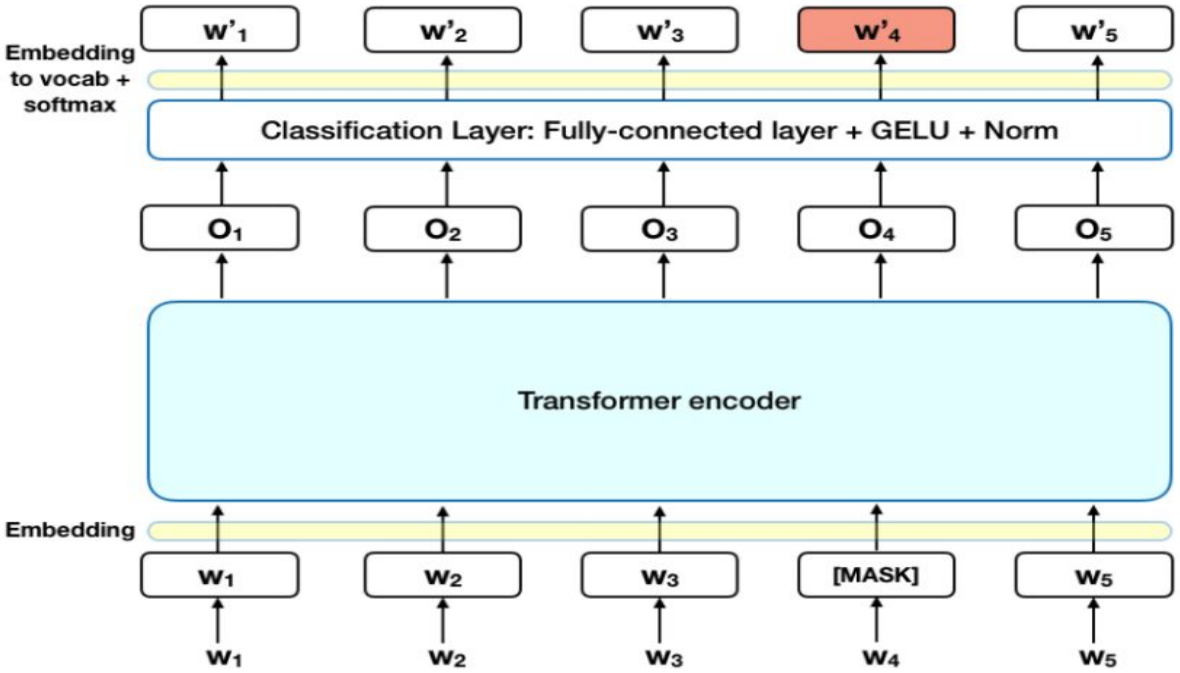
- Bert-base-cased
- Case-sensitive: it makes a difference between english and English
- BERT(Bidirectional Encoder Representations from Transformers) is a transformers model pretrained on a large corpus (English Wikipedia, and the Brown Corpus) of English data in a self-supervised fashion

Model

- The transformer is the part of the model that gives BERT its increased capacity for understanding context and ambiguity in language.
- Masked language modeling (MLM) -> hide a word in a sentence and then have the program predict what word has been hidden (masked) based on the hidden word's context.

Masked Language Modelling (MLM)

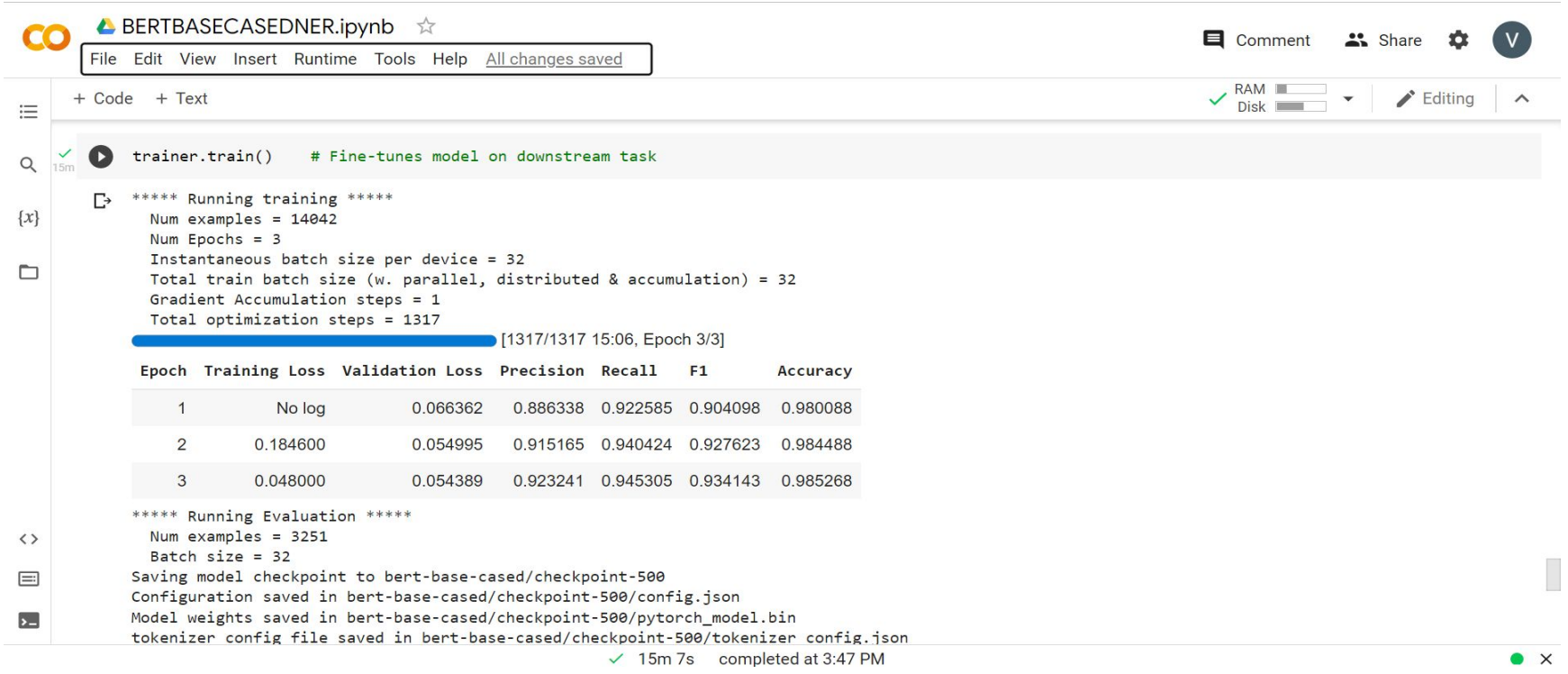
- 15% of the words in each sequence are replaced with a [MASK] token.
- The model then attempts to predict the original value of the masked words, based on the context provided by the other, non-masked, words in the sequence.



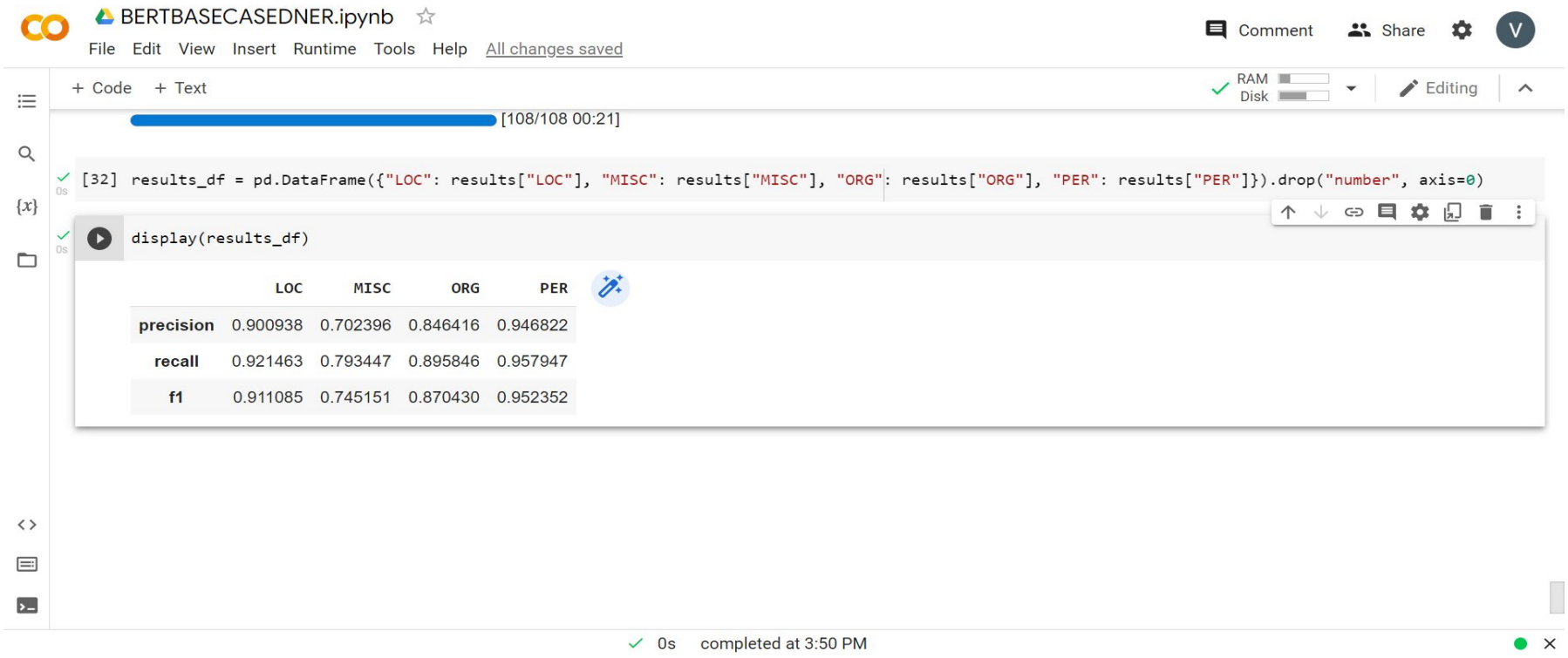
Steps in code implementation

1. Importing required packages
2. Load Dataset
3. Tokenizer
4. Align tokens and labels
5. Importing model
6. Fine tuning
7. Metrics

Results -1:


BERTBASECASEDNER.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
V
+ Code + Text
RAM
Disk
Editing
trainer.train() # Fine-tunes model on downstream task
***** Running training *****
Num examples = 14042
Num Epochs = 3
Instantaneous batch size per device = 32
Total train batch size (w. parallel, distributed & accumulation) = 32
Gradient Accumulation steps = 1
Total optimization steps = 1317
[1317/1317 15:06, Epoch 3/3]
Epoch Training Loss Validation Loss Precision Recall F1 Accuracy
1 No log 0.066362 0.886338 0.922585 0.904098 0.980088
2 0.184600 0.054995 0.915165 0.940424 0.927623 0.984488
3 0.048000 0.054389 0.923241 0.945305 0.934143 0.985268
***** Running Evaluation *****
Num examples = 3251
Batch size = 32
Saving model checkpoint to bert-base-cased/checkpoint-500
Configuration saved in bert-base-cased/checkpoint-500/config.json
Model weights saved in bert-base-cased/checkpoint-500/pytorch_model.bin
tokenizer config file saved in bert-base-cased/checkpoint-500/tokenizer_config.json
15m 7s completed at 3:47 PM

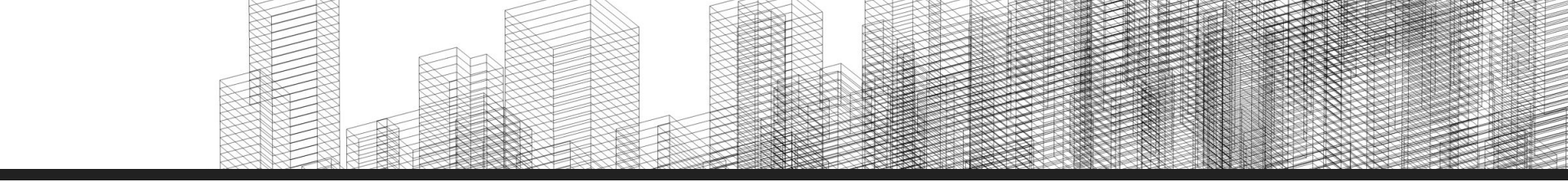
Results - 2:



	LOC	MISC	ORG	PER
precision	0.900938	0.702396	0.846416	0.946822
recall	0.921463	0.793447	0.895846	0.957947
f1	0.911085	0.745151	0.870430	0.952352

Results - 3:


BERTBASECASEDNER.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
matplotlib.pyplot.title("Performance on Test data")
matplotlib.pyplot.show()
Performance on Test data
1.0
0.8
0.6
0.4
0.2
0.0
LOC
MISC
ORG
PER
precision
recall
f1
0s completed at 3:52 PM



Future Scope

- Implementing data preprocessing for user data.
- Extending the model for other user defined entities.
- Improving the usability of model for other datasets.

Questions?

Thank you!

References

- https://deepai.org/dataset/conll-2003-english
- https://huggingface.co/datasets/conll2003
- Inside–outside–beginning (tagging) - Wikipedia
- https://iq.opengenus.org/bert-cased-vs-bert-uncased/
- https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270
- https://towardsdatascience.com/named-entity-recognition-with-nltk-and-spacy-8c4a7d88e7da
- https://en.wikipedia.org/wiki/Named-entity_recognition
- A Guide to Text Preprocessing Using BERT (analyticsindiamag.com)