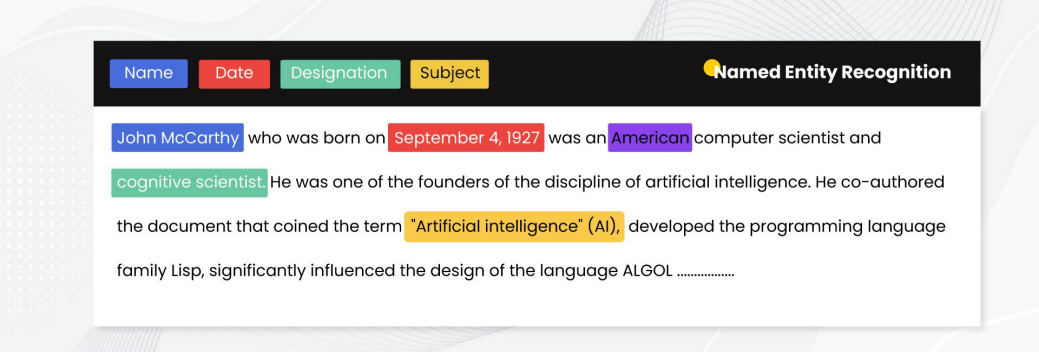
CAP 6640 Group 5

Information Extraction from documents using Named Entity Recognition

Nikhil Sreedhar, Tsogjavkhlan Odbayar, Eduardo Bourget, Elmaddin Azizli

Problem Overview



Problem statement

- To perform NER (Named Entity Recognition) on datasets of information from various kinds of text documents to identify and categorize crucial information, to perform effective information extraction on those documents. Compare the performances of multiple models based on their predictive accuracy.

Problem importance and relevance

- Information extraction serves as a rather important computational approach to process and categorize important words and/or phrases from a body of text. This can have important effects and outcomes in tasks for a variety of industries. In the legal industry, NER can identify, categorize, and extract information pertaining to key evidence such as people, locations, times, dates, organizations, etc. This can help legal professionals, such as attorneys, quickly identify key information from a large body of text, which saves time.

Data collection

Our main dataset was known as the Groningen Meaning Bank dataset. This dataset is found on Kaggle and has two separate dataset files. One is a larger 157MB dataset with many features per data sample. Another is a smaller 15MB dataset that has been simplified with respect to the amount of features. The link to this dataset is as follows:

https://www.kaggle.com/datasets/abhinavwalia95/entity-annotated-corpus/data

Another dataset we used as a stretch goal was a legal NER dataset. This dataset had a size of 14MB and contained information from Indian court judgements. The link to this dataset is as follows:

https://www.kaggle.com/datasets/hhxxzzby/ner-legal

Data quality for both datasets was strong as both datasets not only contained lots of high quality samples of sentences and words, they also contained many features, enabling flexibility and multiple options for feature engineering approaches. Furthermore, the high quality of the data contributes to the decent to high performance we observe from most of the models.

Dataset specifications

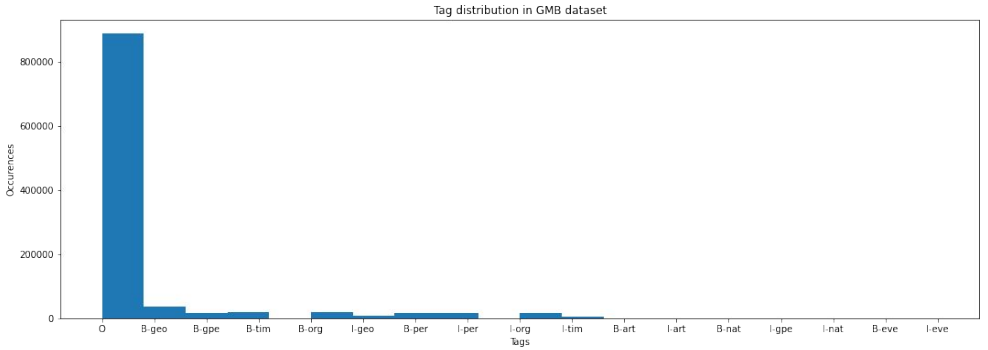
Used an annotated dataset for NER incorporating GMB (Groningen Meaning Bank) corpus.

- The Groningen Meaning Bank (GMB) is a dataset of multi-sentence texts, together with annotations for parts-of-speech, named entities, lexical categories and other natural language structural phenomena.

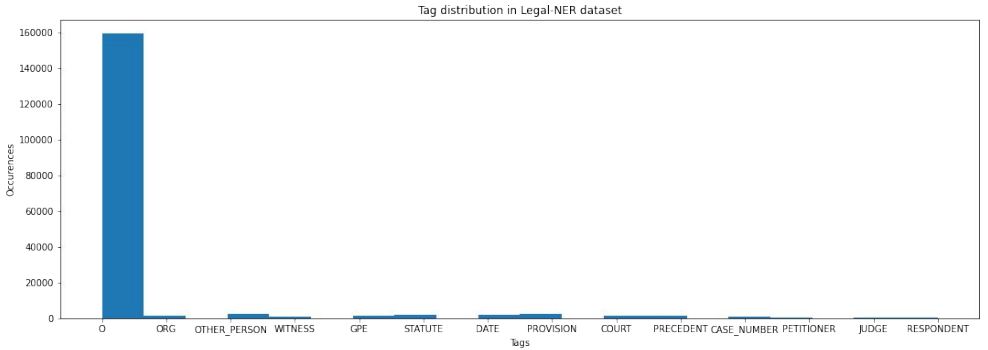
Attempted to use legal-NER dataset associated with the paper "Named Entity Recognition in Indian court judgments" which contains a corpus of 46545 annotated legal named entities mapped to 14 legal entity types for NER on legal entities.

Data overview

The annotated version of GMB dataset contains 1048575 words along with its tags



However, the legal-NER dataset contains only 176606 words and their tags



Data preprocessing and Feature Engineering

For preprocessing and feature engineering of GMB dataset:

- Propagate previous row/column value on encountered NaN missing values using pandas.ffill() method.
- Feature engineering:
  - Each row contains a single word and a single tag.
  - We aggregate over rows on the same sentence to obtain a list of words and a list of tags for each sentence.
- Pad input sequences and truncate lists to max_len.
- Tokenize word list with AutoTokenizer
  - For Bi-LSTM, tokenization was done by splitting the resulting sentence list into word tokens per each sentence in the sentence list after the feature engineering approach of grouping the words as per their sentence index.
- Encode word tokens to numerical format
  - For Bi-LSTM, words and NER tags were encoded to their index position numeric value in their respective vocabulary structure.
- Add special tokens both ends of both tokenized word list and target tag list
- Pad lists until max_len

Data preprocessing

For preprocessing of legal-NER dataset:

- Parse JSON data
- Propagate previous row/column value on encountered NaN missing values using pandas.ffill() method.
- Feature engineering:
  - Each row contains a sentence and labels and positions for named entities
  - We loop over named entities in a sentence and use nltk.word_tokenize on words other than named entities to add it to our word list
- Remaining preprocessing steps are the same as before:
  - Truncate lists to max_len
  - Tokenize word list with AutoTokenizer
  - Add special tokens both ends of both tokenized word list and target tag list
  - Pad lists until max_len

Models

Bi-LSTM:

- Input layer, Embedding layer to generate dense word vectors, No dropout. 128 LSTM units. Time distributed dense operation per each timestep of input sequence.

BERT base NER

DeBERTa

RoBERTa

DistilBERT

Bidirectional LSTM

We used a Bi-LSTM model with the following configuration:

Layer (type)	Output Shape	Param #
input_layer (InputLayer)	(None, 140)	0
embedding (Embedding)	(None, 140, 128)	3,862,016
bidirectional (Bidirectional)	(None, 140, 256)	263,168
time_distributed (TimeDistributed)	(None, 140, 17)	4,369

Total params: 4,129,553 (15.75 MB)

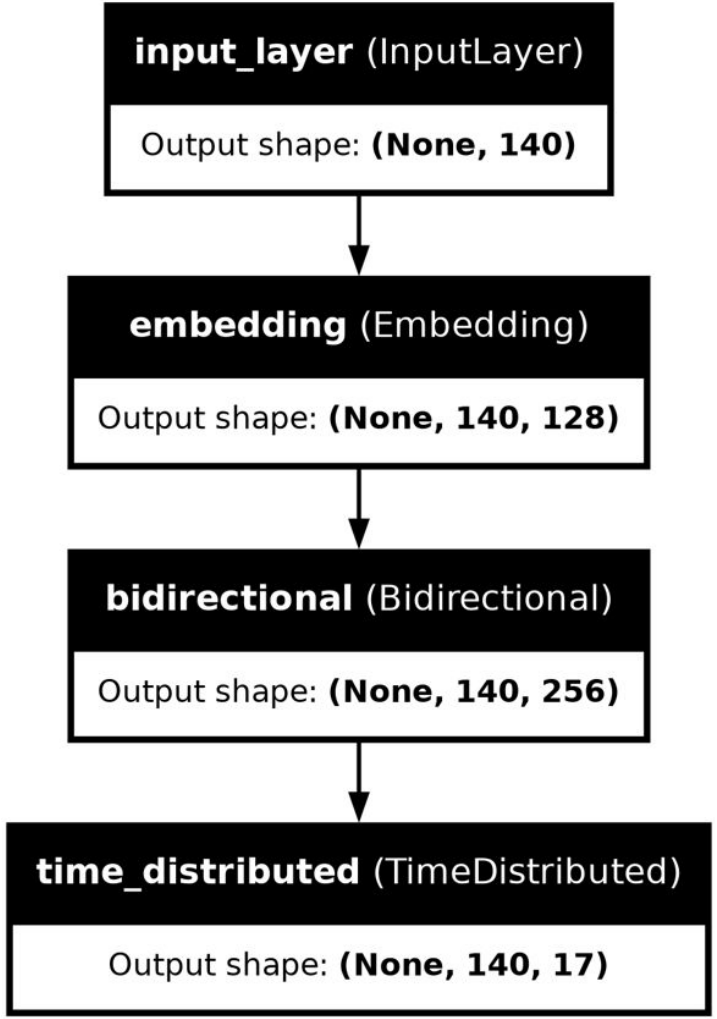
Trainable params: 4,129,553 (15.75 MB)

Non-trainable params: 0 (0.00 B)

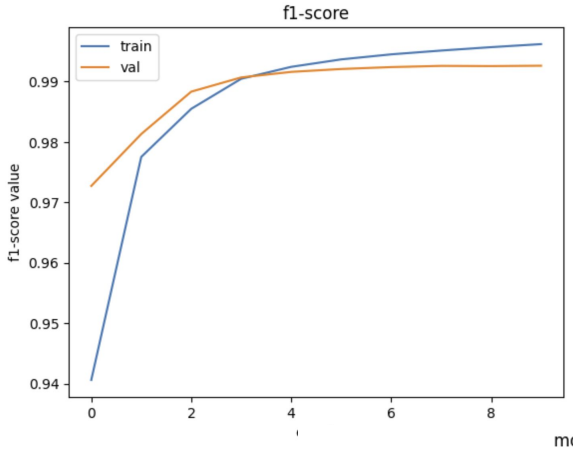
Training was done for 10 epochs using batch size of 256 and a learning rate of 1e-3.

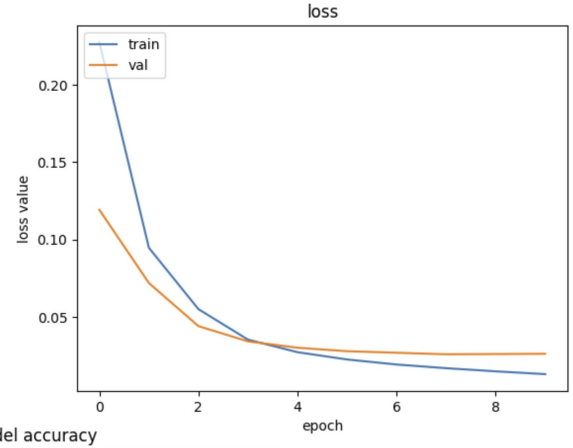
For this model, the words of each sentence along with the tags corresponding to each word/phrase were converted into numeric format for input into the model.

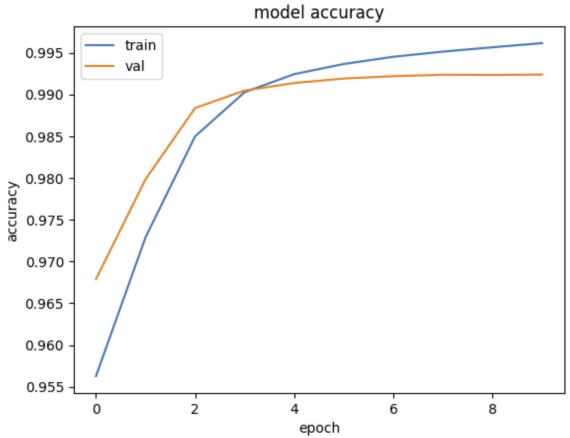
From there, the input sequences are padded with 0s at the end if they are shorter in length than the input size or truncated to the max possible length of all input sequences if they are longer than the input size.



Bi-LSTM results



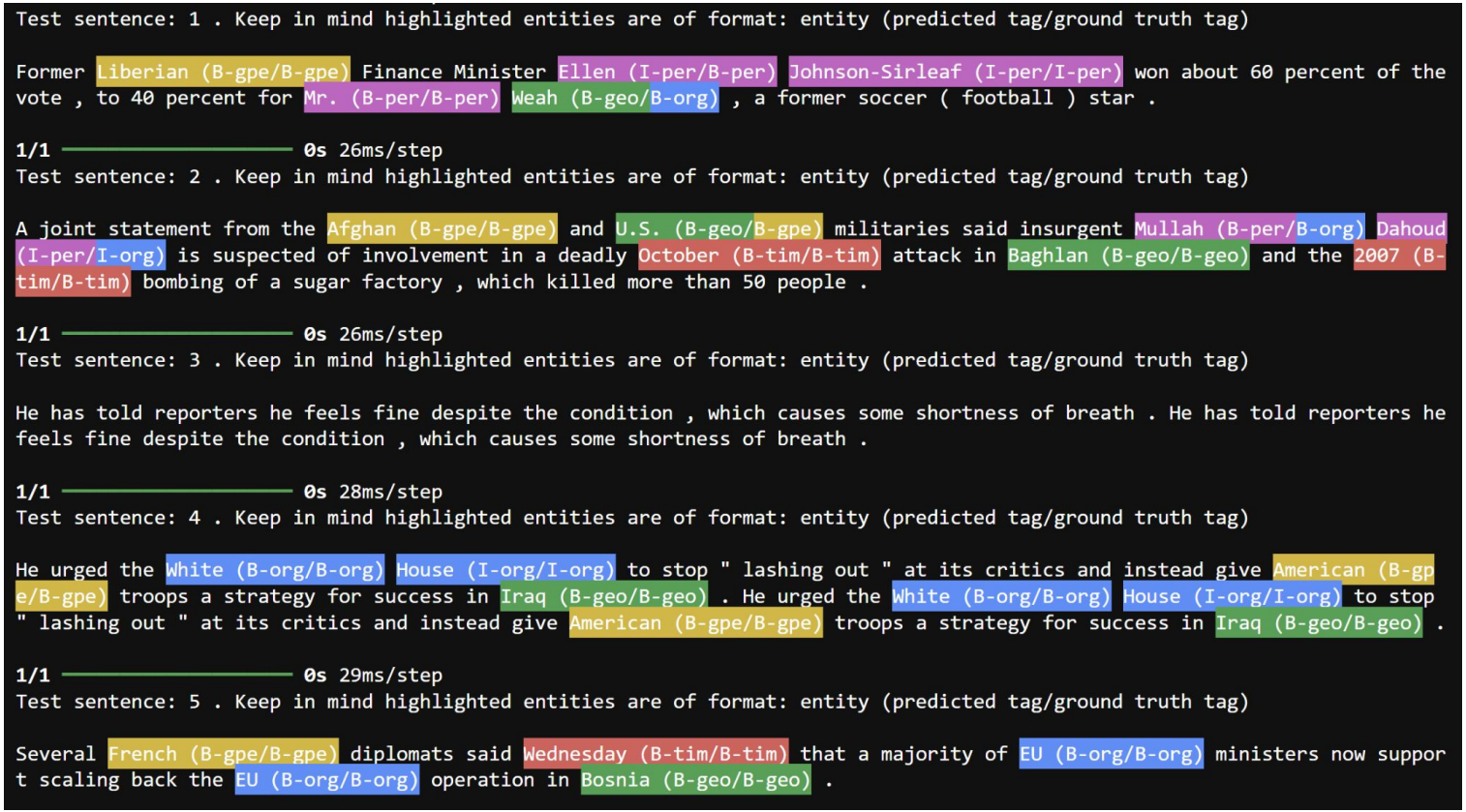




	precision	recall	f1-score	support
B-art	0.00	0.00	0.00	108
B-eve	0.92	0.18	0.30	67
B-geo	0.83	0.89	0.86	7483
B-gpe	0.95	0.92	0.94	3233
B-nat	0.00	0.00	0.00	39
B-org	0.81	0.63	0.71	4068
B-per	0.84	0.76	0.80	3352
B-tim	0.91	0.86	0.89	3948
I-art	0.00	0.00	0.00	94
I-eve	0.00	0.00	0.00	47
I-geo	0.77	0.81	0.79	1525
I-gpe	0.00	0.00	0.00	41
I-nat	0.00	0.00	0.00	19
I-org	0.80	0.72	0.76	3455
I-per	0.86	0.83	0.85	3381
I-tim	0.76	0.69	0.72	1095
O	1.00	1.00	1.00	953085
accuracy			0.99	985040
macro avg	0.56	0.49	0.51	985040
weighted avg	0.99	0.99	0.99	985040

f1 score on testing set: 0.9919360849360044

Bi-LSTM NER visualization



dslim/bert-base-NER

This model is pre-trained on CoNLL 2003 dataset.

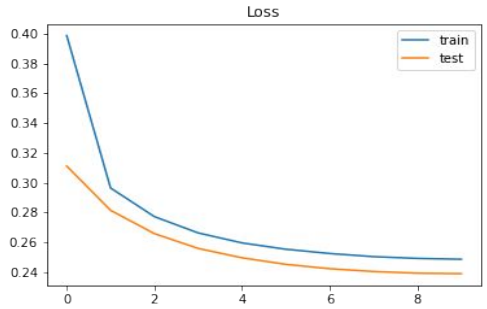
- It has been trained to recognize four types of entities: location (LOC), organizations (ORG), person (PER) and Miscellaneous (MISC).
- The English data was taken from the Reuters Corpus.
  - This corpus consists of Reuters news stories between August 1996 and August 1997.
  - The text for the German data was taken from the ECI Multilingual Text Corpus.
- We changed the entity mapping to match the labels of the 8 labels of the model for increased performance.

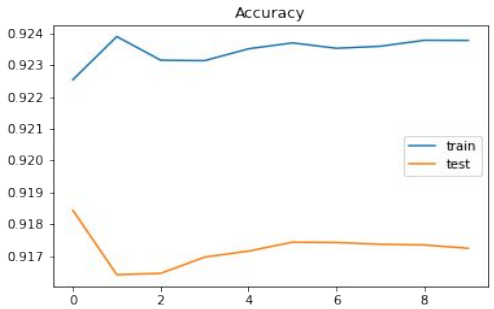
We fine-tuned this model on both of the datasets with same model configuration and hyperparameters(epoch=10, AdamW → lr=5e-5, eps=1e-12) and achieved:

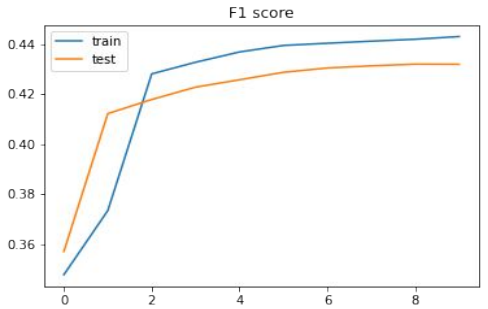
- 91.7% accuracy, 43.2% f1 score on GMB dataset
- 83.7% accuracy, 45.9% f1 score on legal-NER dataset

dslim/bert-base-NER

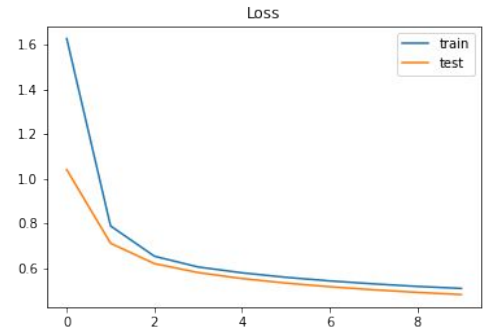
GMB dataset result

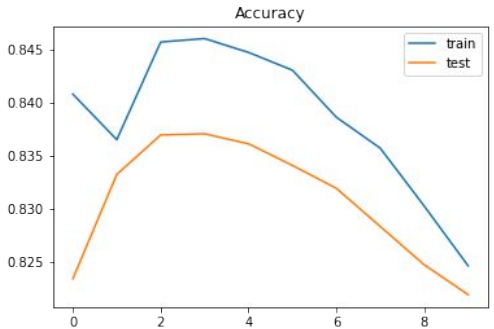


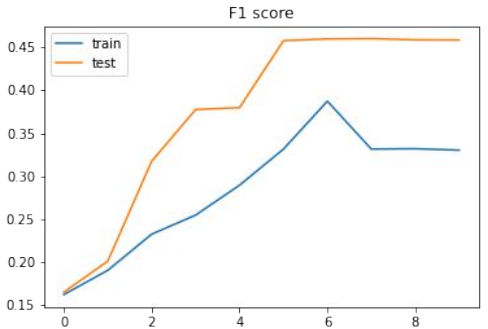




legal-NER dataset result







DeBERTa

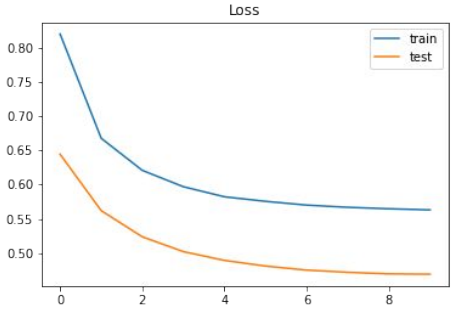
“DeBERTa: Decoding-enhanced BERT with Disentangled Attention” model builds on RoBERTa with disentangled attention and enhanced mask decoder training with half of the data used in RoBERTa.

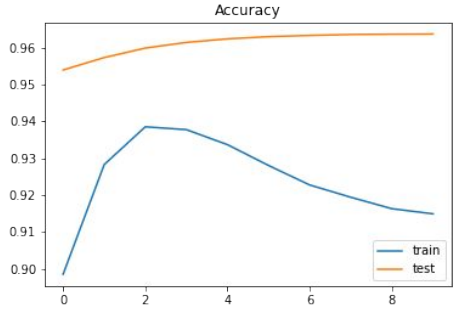
For DeBERTa model, we tried fine-tuning microsoft/deberta-base model on our datasets along with the same optimizer hyperparameters from bert-base-ner model and got:

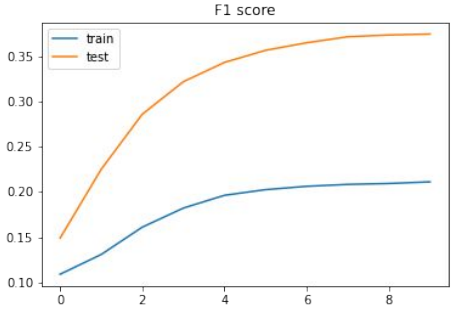
- 96.5% accuracy, 37.5% f1 score on GMB dataset
- 94.3% accuracy, 28.1% f1 score on legal-NER dataset

DeBERTa

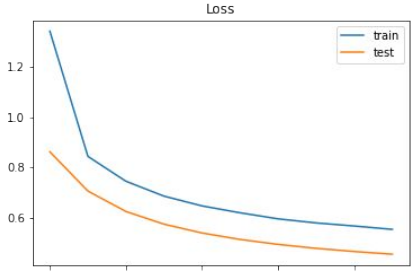
GMB dataset result

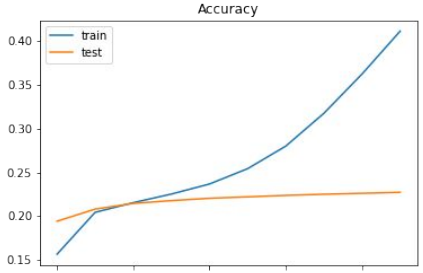


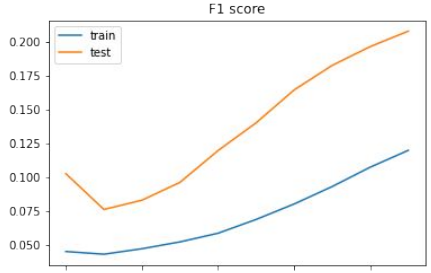




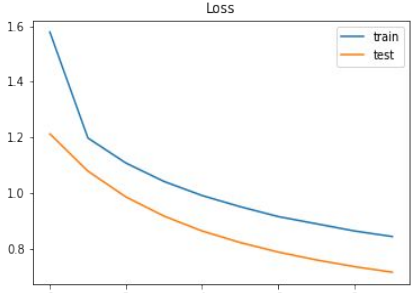
legal-NER dataset result (9 class)

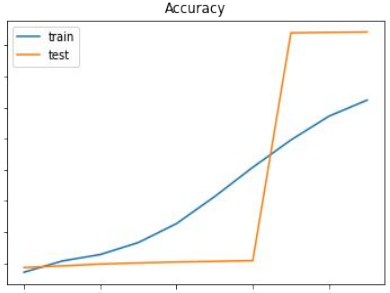


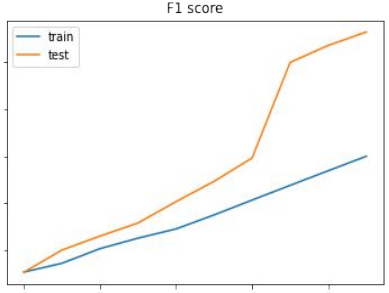




legal-NER dataset result (14 class)







DistilBERT model

- **Data Preparation and Preprocessing**:
- Renamed columns: Original columns were "sentence_idx", "word", and "tag".
- Converted labels to uppercase: Ensured uniformity in label format.
- Train-test split: 80% of the data used for training, 20% for testing.
- **Model Training**:
- Trained on 80% of the data.
- Number of training epochs: 8
- Total training time: Approximately 16 minutes and 37 seconds

Model Evaluation:

- Evaluated on 20% of the data.
- Evaluation loss: 0.6627
- Precision: 0.31996
- Recall: 0.23690
- F1-score: 0.27223

RoBERTa

“RoBERTa: A Robustly Optimized BERT Pretraining Approach” model builds on BERT and modifies key hyperparameters, removing the next-sentence pretraining objective and training with much larger mini-batches and learning rates.

For RoBERTa model, we tried fine-tuning microsoft/codebert-base model on our datasets along with the same optimizer hyperparameters from bert-base-ner model.

Unfortunately, due to the large training parameters and computational requirements for such model, we couldn’t successfully run it in time.

However, benchmarks on several datasets suggest a similar accuracy as DeBERTa when given the right parameters and time to train.

Conclusion

Through this project, we evaluated different models such as BiLSTM and BERT based large language models for the task of Named Entity Recognition using Groninger Meaning Bank dataset for all models as well as Legal dataset of Indian court judgement cases for a couple of the models.

From our results:

- the LSTM model performs best on the GMB dataset with an F1 score **0.992** and accuracy of **0.99**

For the Legal NER dataset:

- the BERT base model performs the best with an F1 score of **0.459** and an accuracy of **0.837**.

Throughout this project, our learnings were multifaceted. We first recapped and learned, at a deep level, how Named Entity Recognition can assist with a variety of tasks by saving people time and effort parsing large amounts of text information and highlighting critical superficial information.

Furthermore, we implemented multiple models to perform NER on multiple datasets, with an increased focus on the GMB dataset. From our experimentation and results, we were able to not only understand our models' performance and benefits, but also visualize how our models performed NER on example data to solidify our understanding of the benefits of NER, specifically in the way we implemented our NER approach. Therefore, we can confidently state that we have achieved a high level of understanding of NER and its broad, time-saving, and useful applications.

References

- https://www.amygb.ai/blog/what-is-named-entity-recognition-in-nlp
- Pengcheng He and Xiaodong Liu and Jianfeng Gao and Weizhu Chen (2021) - “DeBERTa: Decoding-enhanced BERT with Disentangled Attention”
- https://huggingface.co/dslim/bert-base-NER
- https://www.kaggle.com/datasets/abhinavwalia95/entity-annotated-corpus
- https://developer.ibm.com/exchanges/data/all/groningen-meaning-bank/
- Kalamkar et al. (2022) - “Named Entity Recognition in Indian court judgments”