MANDATE: 2

CONTRIBUTION PREPARATION AND PLANNING OF DATASET AND PRE TRAINED MODEL

Keywords QuestionAnsweringModel, NLP, IPC, LEGAL-BERT, ENCODING, TOKENIZATION

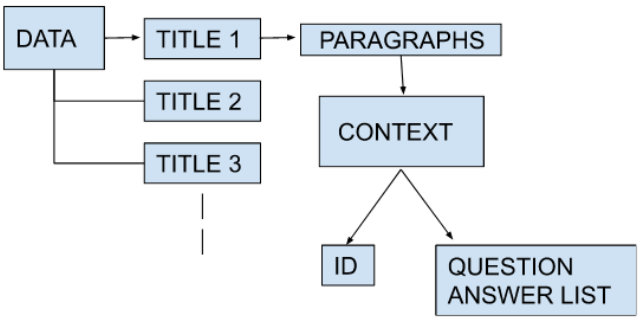
1 DATASET SELECTION:

IPC (Indian Penal Codes) data is available online on Government Site in .pdf format: "https://legislative.gov.in".

But pre trained models like BERT takes data in a particular format.

1.1 HUGGING FACE TRANSFORMERS:

BERT/DistilBERT accepts data in json format:




```
{'title': 'S. 302 Punishment for murder',
 'paragraphs': [{'context': 'Punishment for murder', 'qas': []},
  {'context': 'Whoever commits murder shall be punished with death, or imprisonment fo
r life, and shall also be liable to fine.',
   'qas': [{'question': 'what is the punishment for murder?',
     'id': '62db2313-6a67-45a0-bf35-1c314b71fd9b',
     'answers': [{'answer_start': 32,
       'text': 'punished with death, or imprisonment for life, and shall also be liabl
e to fine.'}]}]}]}
```


For QuestionAnswering Model, the dataset needs to be converted in the above format to feed pretrained model BERT or DistilBERT.

We can do this using HAYSTACK ANNOTATION TOOL.

1.2 HAYSTACK ANNOTATION TOOL:

We can upload our data on this tool and using it we can select a particular text and add question answer sets. This tool then, will automatically format our data as required.

2 PRE-TRAINED MODEL:

2.1 DistilBERT

DistilBERT is smaller but faster version of BERT with maintaining similar accuracy. But it may not perform as BERT in complex tasks.

2.2 LEGAL-BERT

LEGAL-BERT has been trained on large corpus of legal texts. It will help in understanding legal terminology better.

Reference: https://huggingface.co/nlpaueb/legal-bert-base-uncased

2.2.1 MODEL

modelName = "nlpaueb/legal-bert-base-uncased"
tokenizer = AutoTokenizer.from_pretrained(modelName)

3 DATA PRE-PROCESSING:

3.0.1 ADDING ANSWER's ENDING INDEX

In our train dataset we have start index for answer in context but don't have the end index. For each answer, from answer text length we can add start index to it and get end index.

3.0.2 DECONSTRUCTION

Some words like don't, won't, can't, shouldn't: if broken down gets converted to: don, t, can, t, shouldn, t which are not meaningfull.

So, using Regex: These words are replaced by:

don't : do not,

won't : will not,

can't : can not, etc.

Example: Regex for won't is:

sentence = re.sub(r"won\'t", "will not", sentence)

3.0.3 TOKENIZATION

It will break the text on the basis of model's vocabulary.

Since, model does not understand text we need to convert each token into numbers. The token will be mapped to corresponding number allocated in model's vocabulary.

modelName = "nlpaueb/legal-bert-base-uncased" tokenizer = AutoTokenizer.from_pretrained(modelName)

```
train_encodings = tokenizer(train_contexts, train_questions, truncation=True, p
val_encodings = tokenizer(val_contexts, val_questions, truncation=True, padding
print(train_encodings)
```

```
{'input_ids': [[101, 511, 212, 464, 210, 821, 210, 207, 473, 102, 1067, 223, 248, 198,
210, 12690, 102, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
```

4 FINE TUNING:

1. Extending DataSet to create labels for a particular question answer pair.

2. Comparing those question answer pair's tags if their the label intersection with current question is zero then assigning negative score to them else positive score.

4. A try to maintain context between various question answer pairs, when the user will not be satisfied with the response we can present these to them.

Taking reference from:

"https://arxiv.org/ftp/arxiv/papers/1901/1901.08746.pdf"

5 NOTEBOOK:

LINK TO .ipynb file:

"https://drive.google.com/file/d/1AEelr52KUehI7VRWo08 ToMVAaG0ZYNPY/view?usp=share$_{l}ink$"

References

[1] Government of India. "https://legislative.gov.in", Indian Penal Code and laws[Dataset]

[2] "https://github.com/re-search/DocProductstart-of-content", DocProduct [Context Approach]