Financial Report Analysis using NLP

Made by Anant Mehta
Mentor: Apoorv Gehlot
(Technical Lead - Metacube)

Metacube Software,
Jaipur, Rajasthan, India

1 Summary

The internship project involved working on natural language processing techniques. The first task was to successfully analyse the 'Financial Report Section' of the provided document. Named Entity Recognition tokens were extracted using four different libraries and the results were merged in one CSV file. Code was written to automate the pipeline of reading a .pdf file and extracting the entities and making a new CSV containing the tokens. Second task involved making the financial section shorter by summarizing using extraction. The third task was to build a QA system specific for the financial document. The client/investor can search for any information through the system.
Skills: Python, NumPy, Pandas, Flair, Transformers, NLP, File Handling.

2 Training Period

Multiple training sessions were organized by the company to enhance practical knowledge and deliver information regarding industry level procedures and SOPs. Some of them are discussed below:

- Git and Github
- Atlassian Products
- Object Oriented Paradigm
- Exception Handling
- Basic Data Science Packages

3 Task 1: Required NER Extraction from Financial Document

- Company Report was provided in the form of unprocessed pdf file. From which 'task' was to extract out the financial part. Using PyPDF 2 and Fitz, was successfully able to read the file, and convert it to text.
- While converting to text, garbage symbols were encountered and spaces were not correctly reflected. So used 'UTF-8' encoding along with 'unicode' library to remove these symbols. Extra spaces was also removed using different modules. Below is the code snippet for the task.

```
def garbage_value():
    from unidecode import unidecode
    output = []
    main_text=[]
    for page in doc:
        output += page.get_text("blocks")
    previous_block_id = 0
    for block in output:
        if block[6] == 0: # We only take the text
            if previous_block_id != block[5]:
                print("\n")
                plain_text = unidecode(block[4])
                main_text.append(plain_text)
```

- After this, the lines with no text and only special characters were removed. Thereafter, all the sentences obtained were parsed and culminated in a single Python list. This list could be further used as an input to various NLP models. Used '.strip()' and '.split()' function to do the above task.
- **Spacy**: SpaCy is known for its speed and efficiency, that being the reason it was first used[1]. Categories of NER labels explored were:

```
spacy_cols={ 'DATE':[], 'EVENT':[], 'GPE':[],
             'LOC':[], 'MONEY':[], 'ORG':[],
             'PERSON':[], 'TIME':[] }
```

- **NLTK**: It offers various NER algorithms, such as rule-based, dictionary-based, and machine learning-based approaches, allowing flexibility in choosing the most suitable method[2]. Categories of NER labels explored were:

```
spacy_cols={ 'DATE':[], 'EVENT':[], 'GPE':[],
             'LOC':[], 'MONEY':[], 'ORG':[],
             'PERSON':[], 'TIME':[] }
```

- **BERT**: Bidirectional Transformer was also tried for the given task[3]. Categories of NER labels explored were:

```
Bert_cols={ 'B-MISC':[], 'I-MISC':[], 'B-PER':[], 'I-PER':[],
            'B-ORG':[], 'I-ORG':[], 'B-LOC':[], 'I-LOC':[] }
```

- **Flair**: Flair is a cutting-edge NLP package with NER models that have already been trained. It works with several languages and gives contextual string embeddings, which make NER forecasts more accurate[4]. Categories of NER labels explored were:

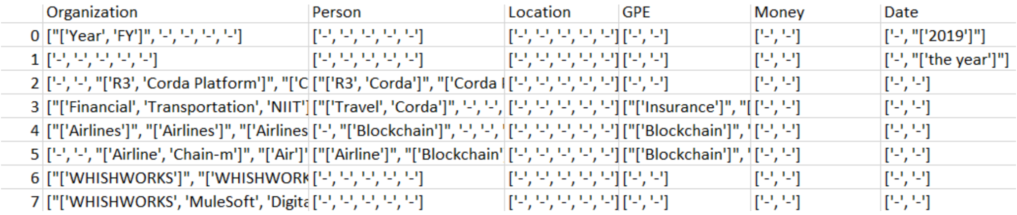
```
flair_cols={ 'PER':[], 'MISC':[], 'LOC':[], 'ORG':[] }
```

- **Results**:
  The data from the all the libraries was taken collected and four different dataframes were created. Then these dataframes were merged into a single one using 'pd.concat()' function.

To do this first same entities were renamed under a single heading e.g., 'ORG' and 'ORGANIZATION' were put under one column 'org'. After creating different series for each heading, all were merged using the below code. Below is the code snippet:

```
def merge(df_org, df_per,df_loc,df_gpe,df_money,df_date):
    df_final=pd.concat([df_org, df_per,df_loc,df_gpe,
    df_money,df_date],axis = 1)
    return df_final
```

- **Table Generated**:
  First few rows of the culminated table are shown below. This table is stored in the form of a csv.



	Organization	Person	Location	GPE	Money	Date
0	["['Year', 'FY']", '-', '-', '-', '-']	['-', '-', '-', '-', '-']	['-', '-', '-', '-', '-']	['-', '-']	['-', '-']	['-', "['2019']"]
1	['-', '-', '-', '-', '-']	['-', '-', '-', '-', '-']	['-', '-', '-', '-', '-']	['-', '-']	['-', '-']	['-', "['the year']"]
2	['-', '-', "['R3', 'Corda Platform']", "['C	["['R3', 'Corda']", "['Corda	['-', '-', '-', '-', '-']	['-', '-']	['-', '-']	['-', '-']
3	["['Financial', 'Transportation', 'NIIT']	["['Travel', 'Corda']", '-', '-',	['-', '-', '-', '-', '-']	["['Insurance']", "[	['-', '-']	['-', '-']
4	["['Airlines']", "['Airlines']", "['Airlines	['-', "['Blockchain']", '-', '-', '	['-', '-', '-', '-', '-']	["['Blockchain']", '	['-', '-']	['-', '-']
5	['-', '-', "['Airline', 'Chain-m']", "['Air']'	["['Airline']", "['Blockchain'	['-', '-', '-', '-', '-']	["['Blockchain']", '	['-', '-']	['-', '-']
6	["['WHISHWORKS']", "['WHISHWORK	['-', '-', '-', '-', '-']	['-', '-', '-', '-', '-']	['-', '-']	['-', '-']	['-', '-']
7	["['WHISHWORKS', 'MuleSoft', 'Digita	['-', '-', '-', '-', '-']	['-', '-', '-', '-', '-']	['-', '-']	['-', '-']	['-', '-']

Fig. 1. Merged CSV with 6 labels

- **Use Cases**:
  Financial Report Analysis can be easily done by investors. Moreover, the extracted entities can serve as key instances/events in the corresponding year. Another usage is for ATS Resume scanners, to extract organizations, institutions and numbers from the resume.
- **Issues faced**:
  BERT model is constrained by its training dataset of entity-tagged news articles from a particular time period. Thus, it may not be applicable to all use cases across all domains. In addition, the model occasionally labels subword tokens as entities, which may necessitate post-processing of the results. For example,

```
Sample word := Airplane
BERT tokenization,
1. B-ORG := Air
2. I-ORG := ##line
Others,
ORG := Airplane
```

So, due to inconsistency in the results, the NER approach with BERT could not be carried forward. Task was well achieved with other three frameworks.

4 Task 2: Document Summarization (Extractive)

A concise information about the financial document was extracted using summarization. Summarizers enable users to save time by swiftly obtaining a summarized version of a document instead of reading it in its entirety. So, the process of decision making is fastened.

Summarization using Spacy:
For extractive summarization using Spacy, below code was used.

```
def summarize_frequency(text, per):
nlp = spacy.load('en_core_web_sm')
doc= nlp(text)
tokens=[token.text for token in doc]
word_frequencies={}
for word in doc:
    if word.text.lower() not in list(STOP_WORDS):
        if word.text.lower() not in punctuation:
            if word.text not in word_frequencies.keys():
                word_frequencies[word.text] = 1
            else:
                word_frequencies[word.text] += 1
max_frequency=max(word_frequencies.values())



def summarize_sentence_scores(max_frequency,tokens):
    for word in word_frequencies.keys():
    word_frequencies[word]=word_frequencies[word]/max_frequency
sentence_tokens= [sent for sent in doc.sents]
sentence_scores = {}
for sent in sentence_tokens:
 for word in sent:
  if word.text.lower() in word_frequencies.keys():
    if sent not in sentence_scores.keys():
      sentence_scores[sent]=word_frequencies[word.text.lower()]
    else:
      sentence_scores[sent]+=word_frequencies[word.text.lower()]
select_length=int(len(sentence_tokens)*per)
summary=nlargest(select_length,
sentence_scores,key=sentence_scores.get)
final_summary=[word.text for word in summary]
summary=''.join(final_summary)
return summary
```

NIIT Technologies forged new partnerships with leading industry players across its focused industries, celebrated the long-term partnerships with its clients and received industry accolades for the partnerships during the year - key ones include:.Collaboration with Microsoft to drive Cloud led transformation and introduce Cognitive Service Desk Audit, a new Cloud- Based Solution.Vamsi Krishna Rupakula appointed as the EVP & Global Head - Infrastructure Management Services With over 20 years of experience in multiple industries and operating domains, Vamsi brings proven expertise in architecting, delivering and operating large-scale infrastructure and business.process solutions for large multi-national clients.Furthermore, as part of this strategy, your company drove a sharp capability augmentation in the Cloud, Cognitive, Automation, Digital and Data areas and made strategic investments in platforms, products, partnerships and in onboarding top-tier leadership talent.

Summarization using NLTK:

```
def summarize_frequency(text, per):
    stopwords = nltk.corpus.stopwords.words('english')
    tokens=[token.text for token in doc]
    word_frequencies = {}
    for word in nltk.word_tokenize(formatted_article_text):
        if word not in stopwords:
            if word not in word_frequencies.keys():
                word_frequencies[word] = 1
            else:
                word_frequencies[word] += 1
def summarize_sentence_scores(max_frequency, tokens):
    sentence_scores = {}
    for sent in sentence_list:
     for word in nltk.word_tokenize(sent.lower()):
       if word in word_frequencies.keys():
        if len(sent.split('␣')) < 100:
         if sent not in sentence_scores.keys():
            sentence_scores[sent] = word_frequencies[word]
         else:
            sentence_scores[sent] += word_frequencies[word]
    import heapq
    summary_sentences = heapq.nlargest(2,
    sentence_scores, key=
    sentence_scores.get)
    summary = '␣'.join(summary_sentences)
return summary
```

Ltd receiving the award Incessant Technologies and Ruletek, NIIT Technologies companies, were recognized with ”Partner Excellence in Growth and Delivery” award by Pegasystems Inc. the software company empowering digital transformation at the world’s leading enterprises.In order to execute on the company’s strategy and accelerate growth, NIIT Technologies continued to add top-tier leadership talent.Vamsi Krishna Rupakula appointed as the EVP & Global Head - Infrastructure Management Services With over 20 years of experience in multiple industries and operating domains, Vamsi brings proven expertise in architecting, delivering and operating large-scale infrastructure and business.process solutions for large multi-national clients. The company also introduced Exact Max to improve risk assessment at the point of underwriting, respond to events, and streamline management of exposures across the organisation.NIIT Technologies forged new partnerships with leading industry players across its focused industries, celebrated the long-term partnerships with its clients and received industry accolades for the partnerships during the year - key ones include:.Collaboration with Microsoft to drive Cloud led transformation and introduce Cognitive Service Desk Audit, a new Cloud- Based Solution.

5 Task 3: A QA System based on the finanical Report

I used the roberta-base model, fine-tuned using the SQuAD2.0 dataset[5]. For the purpose of answering questions, it has been trained on question-answer pairs, even those that cannot be answered.

All the sentences collectively are termed as ’context’. A context needs to be fed in the pipeline along with the question. Reply is prompted by the model in a json format, along with the first and last pointer location and confidence score. The hyperparameters for the used transformers are shown below in table 1:

Table 1. Hyperparameters

Parameter	**value**
batch_size	96
n_epochs	2
base_LM_model	roberta-base
max_seq_len	386
learning_rate	3e-5
lr_schedule	LinearWarmup
warmup_proportion	0.2
doc_stride	128
max_query_length	64

Code to develop the context based QA system[6] is as follows:

```
def QA(context):
    from transformers import AutoModelForQuestionAnswering, pipeline
    model_name = ”deepset/roberta-base-squad2”
    nlp = pipeline('question-answering', model=model_name,
                      tokenizer=model_name)
    QA_input = {'question': 'Report␣is␣for␣which␣year?',
                  'context': context }
   res = nlp(QA_input)
   return res
```

The results are as follows:

```
{'score': 0.3598778545856476,
 'start': 1882,
 'end': 1886,
 'answer': 'FY19'}
```

Another Question:

```
    QA_input = {'question': 'Company␣came␣up␣with␣which
␣␣␣␣solution␣for␣the␣airline?',
                  'context': context }
   res = nlp(QA_input)
   return res
```

The results for this question are as follows:

```
{  'score': 0.07767104357481003,
   'start': 748,
   'end': 755,
   'answer': 'Chain-m'}
```

The code starts the question-answering process by calling the nlp pipeline with the QA input dictionary as an argument. The pipeline processes the input using the tokenizer and pre-trained model, then produces a dictionary containing the solution. The variable 'res' is given the value of the outcome.
By allowing the investor to get specified information from a large financial document, including annual report and financial statements, QA systems play a vital role in expediting financial analysis and research.

References

1. Bhargav Srinivasa-Desikan. *Natural Language Processing and Computational Linguistics: A practical guide to text analysis with Python, Gensim, spaCy, and Keras.* Packt Publishing Ltd, 2018.
2. Edward Loper and Steven Bird. Nltk: The natural language toolkit. *arXiv preprint cs/0205028*, 2002.

3. Francisca Adoma Acheampong, Henry Nunoo-Mensah, and Wenyu Chen. Transformer models for text-based emotion detection: a review of bert-based approaches. *Artificial Intelligence Review*, pages 1–41, 2021.
4. Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. Flair: An easy-to-use framework for state-of-the-art nlp. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics (demonstrations)*, pages 54–59, 2019.
5. Mark Yatskar. A qualitative comparison of coqa, squad 2.0 and quac. *arXiv preprint arXiv:1809.10735*, 2018.
6. Esra Yıldız, Şerife Saçmacı, Şenol Kartal, and Mustafa Saçmacı. A new chelating reagent and application for coprecipitation of some metals in food samples by faas. *Food chemistry*, 194:143–148, 2016.