PROJECT TITLE:

Natural Language Processing and Named Entity Recognition Project

Made By:

Diya Bansal
1st Year CS, Cornell University, U.S.A.
db688@cornell.edu
+1-607-262-6514

Duration of Internship: June 1, 2022 – July 1, 2022

Under the Guidance of:

Sandeep Kumar
Apoorv Gehlot
Manu Raj Hada

Contents

Conclusion

References

Problem Statement:

- To get acquainted with basic Python tools, the Pandas library, and the Numpy library.
- To create, manage, and use Jira, Confluence, and BitBucket accounts.

Week 1: Introduction

Week 1 involved an introduction to the basics of python and some corresponding deliverables. In addition, I read up on the Pandas and Numpy libraries of python and completed some practice coding exercises as deliverables for them too.

• Python basics

Python basics involved covering the fundamental tools of python and understanding how to code in the language as well as understanding some simple but important coding features such as functions, input/output, exceptions, and object-oriented programming.

• Pandas library

The pandas library is a library that is extensively used to store, sort, manipulate and analyze data with. I completed an introductory course to pandas on Kaggle as well as some practice exercises in Python in which I primarily learnt how to use and analyze data that is stored in DataFrames.

• Numpy library

The numpy library is extensively used to perform mathematical operations on data stored in arrays in Python. I completed some readings on the numpy library and then did some practice exercises which involved performing operations such as indexing, slicing, sorting, searching, statistical functions, and so forth on data.

Problem Statement:

- To analyze a section of the ‘NIIT Technologies Annual Financial Report 2018-2019’ and perform Named Entity Recognition (NER) on it using different Python packages and APIs.
- To extract various types of entities from the report and confidently identify them.

Week 2 and 3: Named Entity Recognition (NER)

NER is a particular type of natural language processing (NLP), which is a subfield of artificial intelligence.

• Introduction

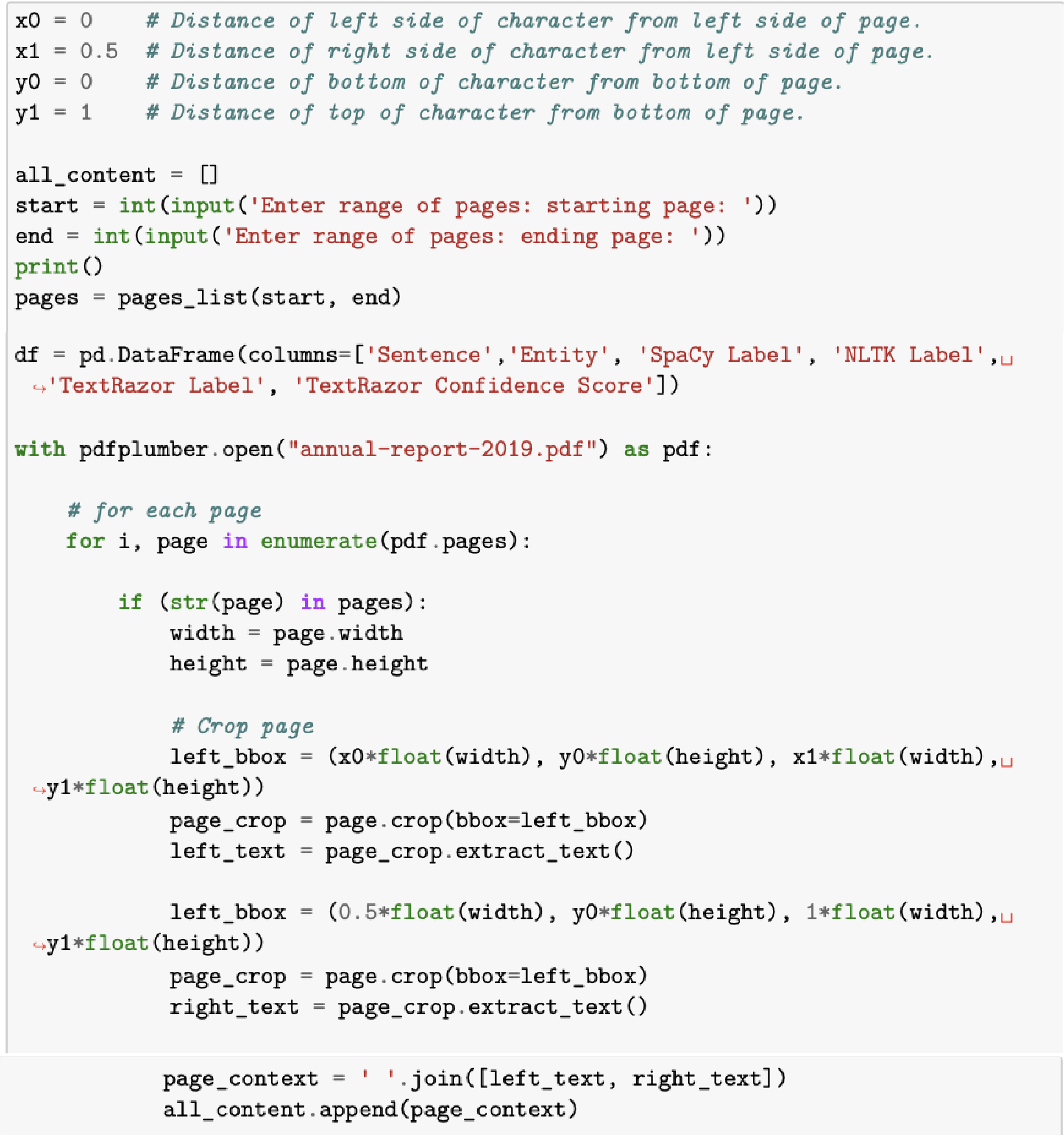
NER is the process of identifying and extracting key information (entities) from text. An entity can be any singular word or sequence of words that falls into some predetermined category. The process of NER involved detecting a named entity and then categorizing it.

• Libraries Used

1. PDFPlumber

PDFPlumber is a Python library that ‘plumbs’ a PDF and extracts detailed information from it. I used PDFPlumber to extract text from the annual report in order to analyze the text for entities.

Using PDFPlumber to extract text from PDF:

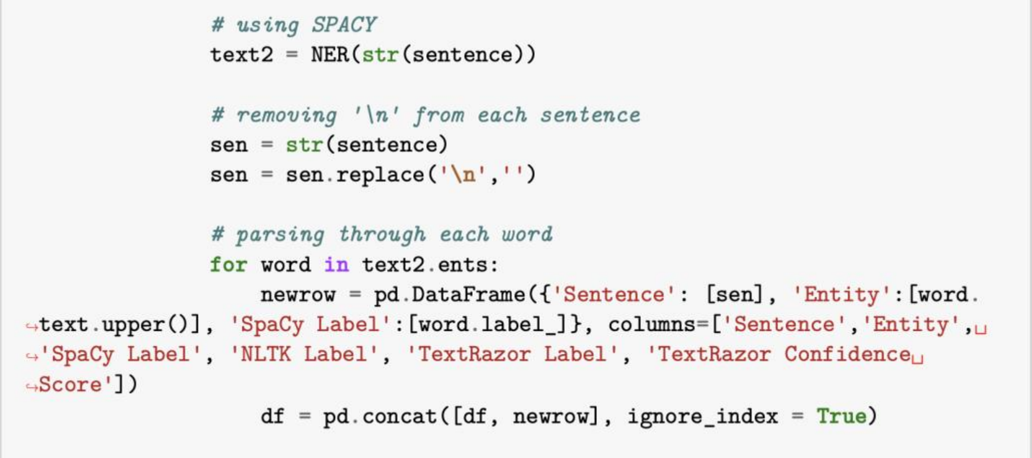


```
x0 = 0    # Distance of left side of character from left side of page.
x1 = 0.5  # Distance of right side of character from left side of page.
y0 = 0    # Distance of bottom of character from bottom of page.
y1 = 1    # Distance of top of character from bottom of page.

all_content = []
start = int(input('Enter range of pages: starting page: '))
end = int(input('Enter range of pages: ending page: '))
print()
pages = pages_list(start, end)

df = pd.DataFrame(columns=['Sentence','Entity', 'SpaCy Label', 'NLTK Label',␣
 ↪'TextRazor Label', 'TextRazor Confidence Score'])

with pdfplumber.open("annual-report-2019.pdf") as pdf:

    # for each page
    for i, page in enumerate(pdf.pages):

        if (str(page) in pages):
            width = page.width
            height = page.height

            # Crop page
            left_bbox = (x0*float(width), y0*float(height), x1*float(width),␣
 ↪y1*float(height))
            page_crop = page.crop(bbox=left_bbox)
            left_text = page_crop.extract_text()

            left_bbox = (0.5*float(width), y0*float(height), 1*float(width),␣
 ↪y1*float(height))
            page_crop = page.crop(bbox=left_bbox)
            right_text = page_crop.extract_text()

            page_context = ' '.join([left_text, right_text])
            all_content.append(page_context)
```

2. SpaCy

SpaCy is a free open-source software library for advanced natural language processing in Python. It includes features that implement NER, POS-tagging, and more. I used SpaCy to extract entities and POS-tags from the text and store the same in a DataFrame. Here is a screenshot of my code:

Using SpaCy to identify entities and tags:

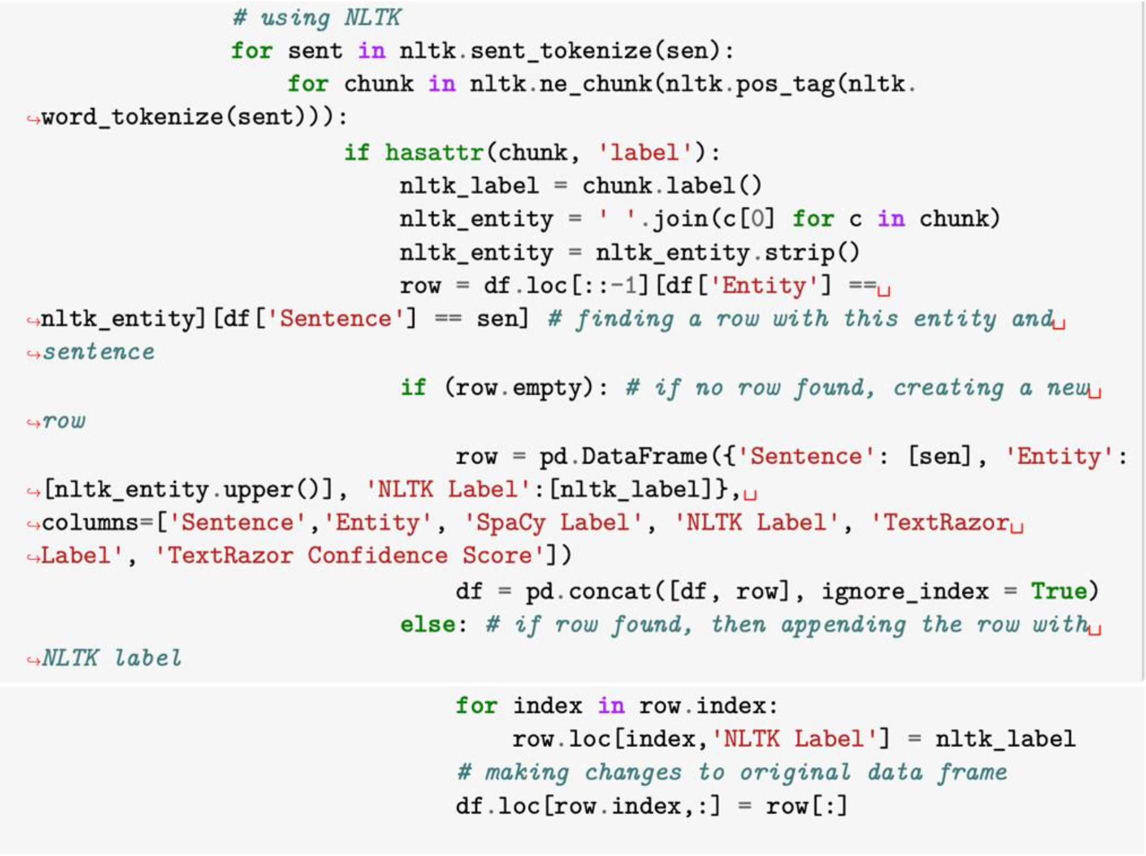


```
                # using SPACY
                text2 = NER(str(sentence))

                # removing '\n' from each sentence
                sen = str(sentence)
                sen = sen.replace('\n','')

                # parsing through each word
                for word in text2.ents:
                    newrow = pd.DataFrame({'Sentence': [sen], 'Entity':[word.
↪text.upper()], 'SpaCy Label':[word.label_]}, columns=['Sentence','Entity',␣
↪'SpaCy Label', 'NLTK Label', 'TextRazor Label', 'TextRazor Confidence␣
↪Score'])
                    df = pd.concat([df, newrow], ignore_index = True)
```

3. NLTK

The NLTK library stands for ‘Natural Language Tool Kit’. It is a Python package that is used for natural and statistical language processing. It is one of the most powerful NLP libraries and is used to make machines understand human language. I used NLTK to perform NER analysis on the text as well and find entities and their tags. Here is a screenshot of my code:

Using NLTK to identify entities and tags:

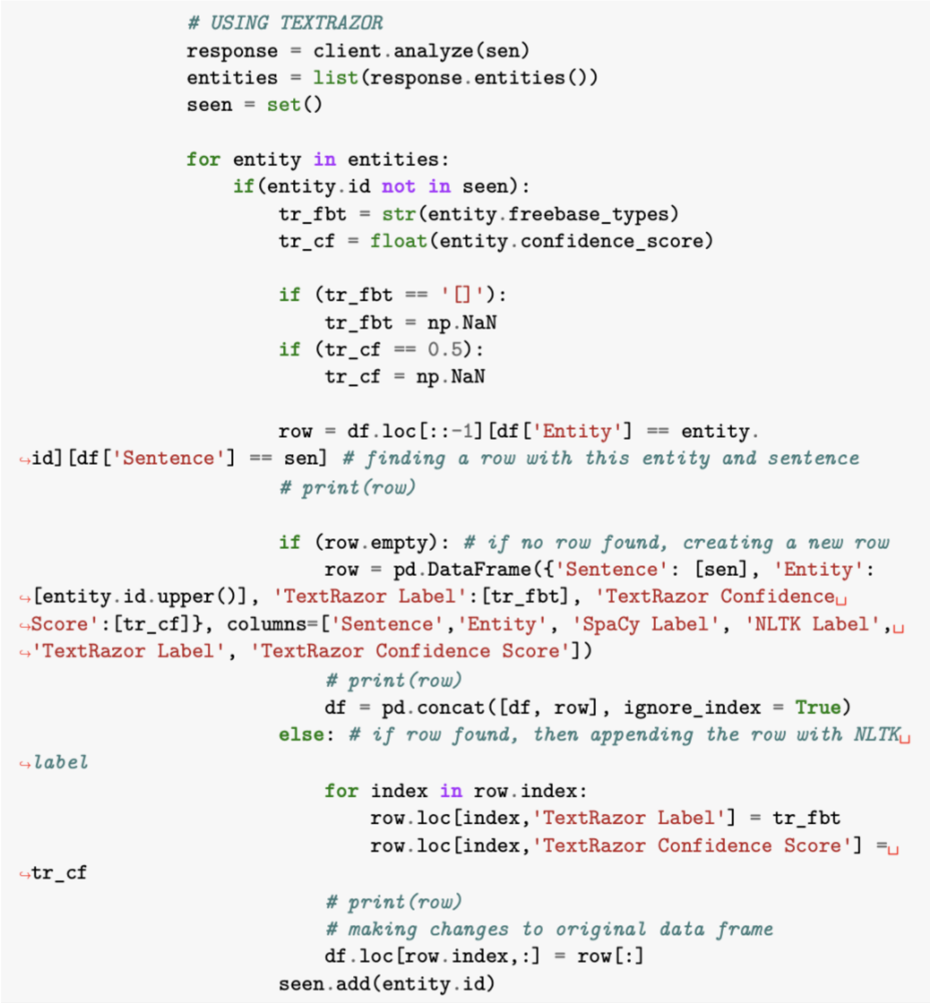


```
                # using NLTK
                for sent in nltk.sent_tokenize(sen):
                    for chunk in nltk.ne_chunk(nltk.pos_tag(nltk.
↪word_tokenize(sent))):
                        if hasattr(chunk, 'label'):
                            nltk_label = chunk.label()
                            nltk_entity = ' '.join(c[0] for c in chunk)
                            nltk_entity = nltk_entity.strip()
                            row = df.loc[::-1][df['Entity'] ==␣
↪nltk_entity][df['Sentence'] == sen] # finding a row with this entity and␣
↪sentence
                            if (row.empty): # if no row found, creating a new␣
↪row
                                row = pd.DataFrame({'Sentence': [sen], 'Entity':
↪[nltk_entity.upper()], 'NLTK Label':[nltk_label]},␣
↪columns=['Sentence','Entity', 'SpaCy Label', 'NLTK Label', 'TextRazor␣
↪Label', 'TextRazor Confidence Score'])
                                df = pd.concat([df, row], ignore_index = True)
                            else: # if row found, then appending the row with␣
↪NLTK label
                                for index in row.index:
                                    row.loc[index,'NLTK Label'] = nltk_label
                                # making changes to original data frame
                                df.loc[row.index,:] = row[:]
```

4. TextRazor API

TextRazor is a natural language processing application programming interface that offers a comprehensive and integrated analysis of texts. Its features include identifying entities along with a long list of synonymous categories that the entity belongs to as well as a confidence score. I use TextRazor to find the above same and store it in a DataFrame as well.

Using TextRazor to identify entities, tags, and a confidence score:



```
                # USING TEXTRAZOR
                response = client.analyze(sen)
                entities = list(response.entities())
                seen = set()

                for entity in entities:
                    if(entity.id not in seen):
                        tr_fbt = str(entity.freebase_types)
                        tr_cf = float(entity.confidence_score)

                        if (tr_fbt == '[]'):
                            tr_fbt = np.NaN
                        if (tr_cf == 0.5):
                            tr_cf = np.NaN

                        row = df.loc[::-1][df['Entity'] == entity.
↪id][df['Sentence'] == sen] # finding a row with this entity and sentence
                        # print(row)

                        if (row.empty): # if no row found, creating a new row
                            row = pd.DataFrame({'Sentence': [sen], 'Entity':
↪[entity.id.upper()], 'TextRazor Label':[tr_fbt], 'TextRazor Confidence␣
↪Score':[tr_cf]}, columns=['Sentence','Entity', 'SpaCy Label', 'NLTK Label',␣
↪'TextRazor Label', 'TextRazor Confidence Score'])
                            # print(row)
                            df = pd.concat([df, row], ignore_index = True)
                        else: # if row found, then appending the row with NLTK␣
↪label
                            for index in row.index:
                                row.loc[index,'TextRazor Label'] = tr_fbt
                                row.loc[index,'TextRazor Confidence Score'] =␣
↪tr_cf
                            # print(row)
                            # making changes to original data frame
                            df.loc[row.index,:] = row[:]
                        seen.add(entity.id)
```

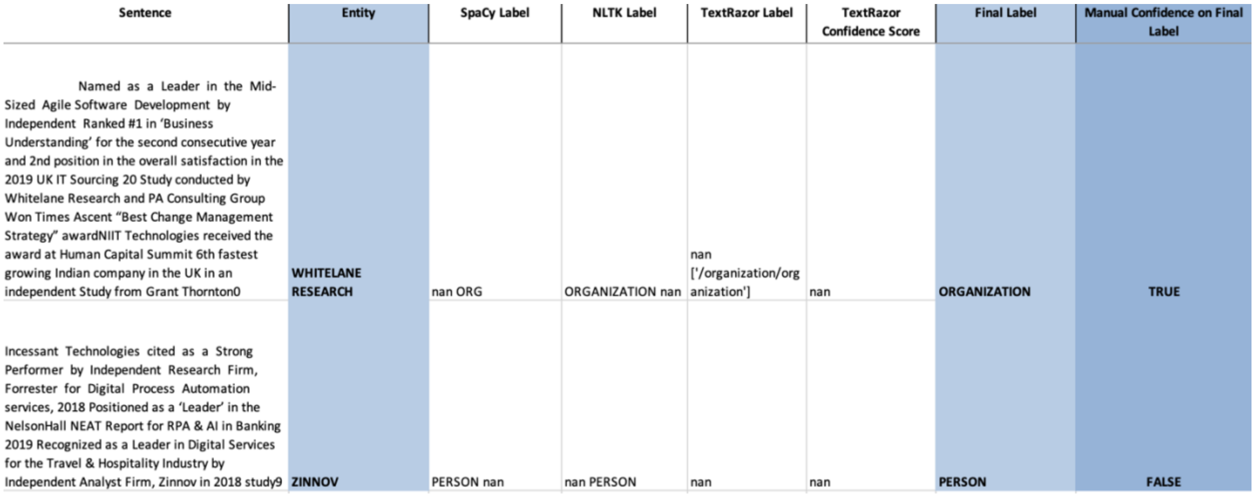
• Categories Explored and Results

After using SpaCy, NLTK, and TextRazor libraries to identify entities and their tags in the text, I stored the data in a DataFrame which I then exported to a Microsoft Excel document.

Each library identifies entities differently. For example, one library identifies a particular phrase as an entity which the other two do not and so on, so forth.

Thus, in order to ascertain whether an identified entity was indeed accurately identified as an entity, I followed a 2/3 identification approach. If two out of the three libraries tagged the same entity in the same sentence with the same/similar tag, then the entity was given the same tag as the 'Final Label'. After running the program and exporting the resultant data (with all the entities that satisfied the 2/3 rule) to another excel sheet, I manually checked whether the final label entities had been correctly or incorrectly identified given the context they were used in in their corresponding sentences. A screenshot of the excel sheet is seen below.

Excel sheet with final entities:



Sentence	Entity	SpaCy Label	NLTK Label	TextRazor Label	TextRazor Confidence Score	Final Label	Manual Confidence on Final Label
Named as a Leader in the Mid-Sized Agile Software Development by Independent Ranked #1 in 'Business Understanding' for the second consecutive year and 2nd position in the overall satisfaction in the 2019 UK IT Sourcing 20 Study conducted by Whitelane Research and PA Consulting Group Won Times Ascent "Best Change Management Strategy" awardNIIT Technologies received the award at Human Capital Summit 6th fastest growing Indian company in the UK in an independent Study from Grant Thornton0	**WHITELANE RESEARCH**	nan ORG	ORGANIZATION nan	nan ['/organization/organization']	nan	**ORGANIZATION**	**TRUE**
Incessant Technologies cited as a Strong Performer by Independent Research Firm, Forrester for Digital Process Automation services, 2018 Positioned as a 'Leader' in the NelsonHall NEAT Report for RPA & AI in Banking 2019 Recognized as a Leader in Digital Services for the Travel & Hospitality Industry by Independent Analyst Firm, Zinnov in 2018 study9	**ZINNOV**	PERSON nan	nan PERSON	nan	nan	**PERSON**	**FALSE**

After doing this, I calculated that my program had an accuracy rate of **53.8%** with this approach. In specific, out of the 26 entities that had been identified and satisfied the 2/3 rule, 14 of them were identified correctly given the context they were used in in their respective sentences.

Furthermore, another noteworthy observation was that the only resultant labels were ORGANIZATION and PERSON. Upon analyzing the labels for each library for all entities, even those which did not satisfy the 2/3 rule, I found out that while SpaCy and TextRazor have a wide variety of labels they assign to identified entities, NLTK is fairly limited and has only a few categories for entities. Moreover, on applying the 2/3 rule, the only entities that satisfied the rule happened to be the ones that were categorized as either an ORGANIZATION and PERSON.

• Results and Discussion

With an accuracy percentage of **53.8%**, we can infer that the model is not very good at identifying entities accurately even though this is a single score that has been calculated after analyzing a 4-page section of the annual report.

Further limitations are related to SpaCy and NLTK packages. NLTK has the following tags: FACILITY, GPE, GSP, LOCATION, ORGANIZATION, PERSON. While SpaCy does have more tags, it still has only 18 tags which makes it still fairly limited in scope. TextRazor has a list of synonymous 'freebase types' instead of a single-word label for its entity tag. This provides for a wider categorization.

Further improvements to the model can be made by giving a higher weightage to TextRazor or by utilizing other packages and APIs that have a wide categorization for labels. This would help reduce discrepancies, identify more entities that could have possibly been missed with my current model, and increase accuracy.

Problem Statement:

- To build a working Machine Learning Model.

Week 4: Machine Learning Model

A machine learning model identifies certain patterns that lie in data and is able to analyze them and learn them. A working model can use this learning to process more data and make predictions based on the information that is provided to it.

• Machine Learning Kaggle Course

In week 4, a completed an introductory machine learning course on Kaggle.com that taught me how to explore data and use it to build a basic machine learning model using a decision tree regressor. During this course, I was also exposed to the potential problems of machine learning models such as underfitting and overfitting.

At the end of this course, I built a working machine learning model that predicts housing prices for users based on the features a user prefers. The user can pick from a list of 79 features.

To build this model, I used a simple decision tree regressor and used the training data provided by Kaggle to train my model. I then applied the model to the test data they provided and predicted house prices.

Conclusion

This is the outcome of performing named entity recognition on a section of the annual financial report of NIIT Technologies for the year 2018-2019. My approach uses named entity recognition to extract various entities and their categories using different libraries before applying a 2/3 rule to confirm that an entity was indeed identified correctly.

Through this internship, I gained in-depth experience and knowledge of data science for the first time on such a level. I was also exposed to the concepts of natural language processing and machine learning on an introductory level. In the one month I spent on this internship, I was able to hone and develop my skills and advance my understanding of NLP, ML, and data science in a way that will prove to be helpful for my career ahead.

During this internship, I also learned many new technologies. While I do have some prior experience with Python, I was working with data science libraries and packages like pandas, SpaCy, NLTK, and TextRazor for the first time. I also learned how to use Jira, Confluence, and BitBucket accounts during this internship.

I would also like to acknowledge and thank my mentors – the team that guided me through this internship – Mr. Sandeep Kumar, Mr. Apoorv Gehlot, and Mr. Manu Raj Hada. Without their support and guidance, I would not have been able to successfully complete this internship. I would like to sincerely thank them for helping me navigate this internship, guiding me through any obstacles I faced, helping me learn many new technologies, and for giving me their invaluable time in completing this internship.

I would also like to thank Metacube Software Pvt Ltd, Jaipur and Mr. Parijat Agarwal who provided me with the opportunity to intern at the company and learn so much from this experience. It was my pleasure to intern and I look forward to having more such opportunities in the future.

References

- https://hackr.io/blog/python-programming-language
- https://www.kaggle.com/code/kashnitsky/topic-1-exploratory-data-analysis-with-pandas
- https://www.oreilly.com/library/view/python-for-data/9781449323592/ch04.html
- https://numpy.org/doc/stable/user/absolute_beginners.html
- https://medium.com/mysuperai/what-is-named-entity-recognition-ner-and-how-can-i-use-it-2b68cf6f545d
- https://monkeylearn.com/blog/what-is-natural-language-processing/
- https://pypi.org/project/pdfplumber/0.1.2/
- https://spacy.io
- https://www.guru99.com/nltk-tutorial.html
- https://www.textrazor.com
- https://www.sas.com/en_ae/insights/analytics/machine-learning.html