Yelp Review Analysis

Submitted By:

Amanpreet Kaur

Fall 2018

LING 571 (Computational Corpus Linguistic)

# Table of Content

[**1 Introduction**](#_j44uvtbdlybk) **2**

[**2 Problem Statements**](#_a5agaccmwsce) **3**

[**3 Dataset**](#_k8dpfyhu4se1) **3**

[**4 Methodology**](#_eg05murdr8en) **4**

[**5 Resources**](#_8dmfl7v81m3e) **5**

[**6 Detailed Methodology**](#_f873vxp1p91b) **5**

[6.1 Corpus Generation](#_abr9qyb6hgh9) 5

[6.2 Data cleaning](#_2jifjty927ta) 7

[6.3 Linguistic Analysis](#_jqosrcgvadx9) 7

[1. Lexical Diversity](#_o1j8evebptjs) 7

[2. Words analysis](#_k0sqaxssd1lg) 8

[3. Word Frequency Analysis (with stop words)](#_rtkqas56hjq8) 9

[4. Word Frequency Analysis (without stop words)](#_vg5y7vj04rn5) 9

[5. Complex word count:](#_czyts5n7vve9) 9

[6. Part of Speech Tagging](#_4z2cknf7lkcq) 10

[6.4 Aspect Identification](#_6u32khowr2h5) 11

[6.5 Entity Identification](#_2o3swjo15met) 14

[**7 Result**](#_p29vel9yol6p) **16**

[**8 Future work**](#_avkrn06m18ph) **17**

[**9 Reference**](#_mb02xxmg7ts7) **18**

# 

# 

# 

# 1 Introduction

Internet has become a worldwide mean of sharing reviews about business. This is a crowd-sourced phenomenon of generating a fair and honest review of any business, crowdsourcing also ensure that the reviews are free of any bias. Today people have been travelling across the country and across the globe. They are more relying on the reviews for these services than ever before. Be it choosing a restaurant for dinner, bar for drinks, or a doctor for medical services. Therefore reviews for any business are very important. In the recent years,

the researchers have shown increasing interest to analyze these reviews. I have chosen to analyze the Yelp reviews for different businesses across the United States which comprise of cosmopolitan demography. I have chosen the states of CA(California), NY (New York), FL (Florida), CO (Colorado), and IL (Illinois) for my project. These states have major tourist attractions that attract tourists across globe and large metropolitans that invite people across country for job and livelihood. This mean that there will be people speaking different languages, there will be native as well as non-native speakers of English. I am interested to analyze the reviews by this assorted group of people and identify interesting patterns.

# 2 Problem Statements

In this project I am analyzing the reviews for their linguistic features. As mentioned earlier I am considering 5 states, the dataset has reviews spread from 2004 to 2016 for these states. I created corpus for each state by aggregating reviews for that state for that year. The reason for creating separate corpus for each year is that the demography of a state tends to change from year to year as people migrate in and out of a state. There are three broad targets :

1. Basic linguistic analysis of the corpus eg. lexical diversity, unique words, verbosity, most frequently used part of speech
2. Identifying major aspects that have been talked about for the year
3. Identifying major entities in the reviews for the year

# 3 Dataset

The data has been taken from Yelp reviews dataset hosted on Kaggle at -

<https://www.kaggle.com/yelp-dataset/yelp-dataset>. This dataset is a subset of Yelp's businesses, reviews, and user data. It was originally put together for the Yelp Dataset Challenge which is a chance for students to conduct research or analysis on Yelp's data and share their discoveries. This dataset contains information about 188593 businesses across 11 metropolitan areas in four countries. The dataset also contains 5996996 reviews that have been collected between 2004 and 2016. As mentioned above I will be only considering 5 states across the United States. There are different type of datasets such as business, checkin, tip etc. For the purpose of my project I will be considering only two datasets - Reviews Dataset, Business dataset. The reason I am only considering these two datasets is that I want to analyze reviews given by users to businesses. Here I have described the schema of the dataset -

**Reviews dataset - yelp\_academic\_dataset\_review.json**

|  |
| --- |
| business\_id:Apn5Q\_b6Nz61Tq4XzPdf9A (alphanum) name:Minhas Micro Brewery (string) Neighborhood: (string) address:1314 44 Avenue NE (alphanum) city:Calgary (String) state:AB (String) postal\_code:T2E 6L6 (alphanum) Latitude:51.0918130155 (float) Longitude:-114.031674872 (float) Stars:4 (int) Review\_count:24 (int) Is\_open:1 (int) attributes:{} 13 items categories:Tours, Breweries, Pizza, Restaurants, Food, Hotels & Travel (string) hours:{} 6 items |

**Business dataset - yelp\_academic\_dataset\_business.json**

|  |
| --- |
| business\_id:Apn5Q\_b6Nz61Tq4XzPdf9A (alphanum) name:Minhas Micro Brewery (string) Neighborhood: (string) address:1314 44 Avenue NE (alphanum) city:Calgary (String) state:AB (String) postal\_code:T2E 6L6 (alphanum) Latitude:51.0918130155 (float) Longitude:-114.031674872 (float) Stars:4 (int) Review\_count:24 (int) Is\_open:1 (int) attributes:{} 13 items categories:Tours, Breweries, Pizza, Restaurants, Food, Hotels & Travel (string) hours:{} 6 items |

# 4 Methodology

This system can be divided into five major components:

1. Corpus Generation: The data from Kaggle is pretty clean and structured but isn’t in the format required by this project. For the purpose of this project we need to a create text corpus for each state from the reviews. Also, for each state I needed to create separate corpus for each year. Also, there were many unwanted columns that needed to be dropped before analysis.
2. Data cleaning: As we are dealing with text data in the project I had to clean the data. This included dropping null values, normalizing text data by lowercasing the words, removing punctuations and digits from reviews, and getting rid of stop-words.
3. Linguistic Analysis: I perform different linguistic analysis on the data. To perform this analysis I had to perform some preprocessing such as tokenization, POS tagging.
4. Aspect Identification: Both aspect identification and entity detection are part of information retrieval and text mining. These both follow common pipeline in the initial state which include tokenization, POS tagging. For aspect identification I had to perform relation-detection and Noun Phrase Chunking.
5. Entity Identification: After preprocessing the data, entity identification is done leveraging Spacy library (discussed below). For each corpus I identify entities in the corpus, also I identify count of entities in each category.

# 5 Resources

The project has been developed purely on Python heavily using following libraries -

1. Pandas: Python Data Analysis Library
2. NLTK: NLTK is a leading platform for building Python programs to work with human language data.
3. SpaCy: SpaCy is a free open-source library for Natural Language Processing in Python. It features NER, POS tagging, dependency parsing, word vectors and more.

# 6 Detailed Methodology

## 6.1 Corpus Generation

For corpus generation I took following steps:

*Step 1: Load the two dataset into pandas and take necessary columns:*

|  |
| --- |
| Columns in Business data - ['address', 'attributes', 'business\_id', 'categories', 'city', 'hours', 'is\_open', 'latitude', 'longitude', 'name', 'neighborhood', 'postal\_code', 'review\_count', 'stars', 'state'] Columns in Reviews data - ['business\_id', 'cool', 'date', 'funny', 'review\_id', 'stars', 'text', 'useful', 'user\_id'] |

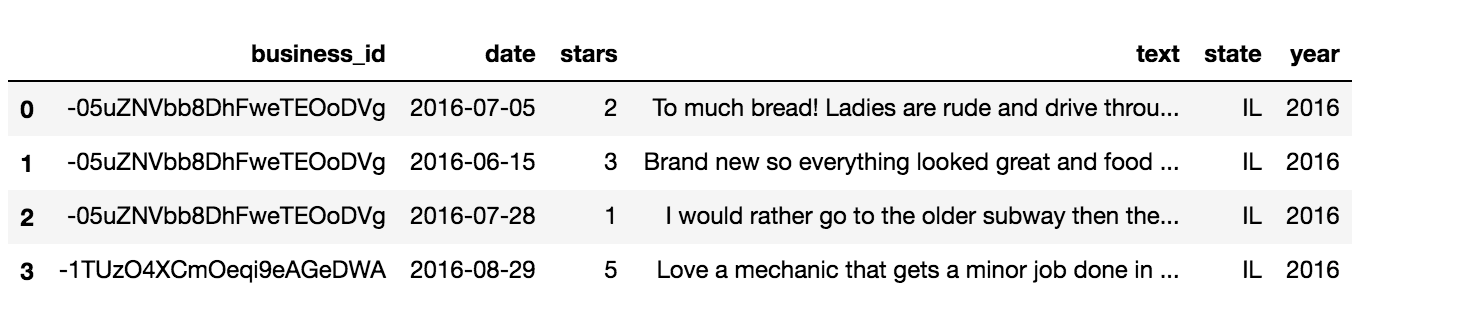
For the purpose of the project I am only considering following columns -

|  |
| --- |
| For Business data - ['business\_id','state'] For Reviews data - ['business\_id','date', 'stars', 'text'] |

Step 2: Filtering reviews data to include records for only the following states - CA(California), NY (New York), FL (Florida), CO (Colorado), IL (Illinois)

Step 3: Merging the two datasets i.e. business and reviews together where key will be business\_id. Resulting columns - ['business\_id', ‘state’, 'date', 'stars', 'text'].

Step 4: Added new column ‘year’ from existing date column. This column will be used to generate corpus for each individual year for the state.



Step 5: Once the new column has been added I am generating the text corpus for users, here I have put the code for that

|  |
| --- |
| def combine\_text(data):  text=[]  data['text'].apply(lambda x: text.append(x))  return " ".join(text)  def gen\_cleaned\_data(df):  cleaned\_data=list()  # For each state combine reviews for each given year and collect in a dataframe  for s in list(df.state.unique()):  years=df[df.state==s].year.unique()  for y in years:  reviews=combine\_text(df[(df.state==s) & (df.year==y)])  cleaned\_data.append({"state":s, "year":y, "review":reviews})  return cleaned\_data |

Note: The corpus generation task was performed after the data cleaning has been performed to get rid of punctuations and

## 6.2 Data cleaning

Data cleaning step involved cleaning the data and converting it in a format that can be utilized in services down the line. Following steps were taken during cleaning the data:

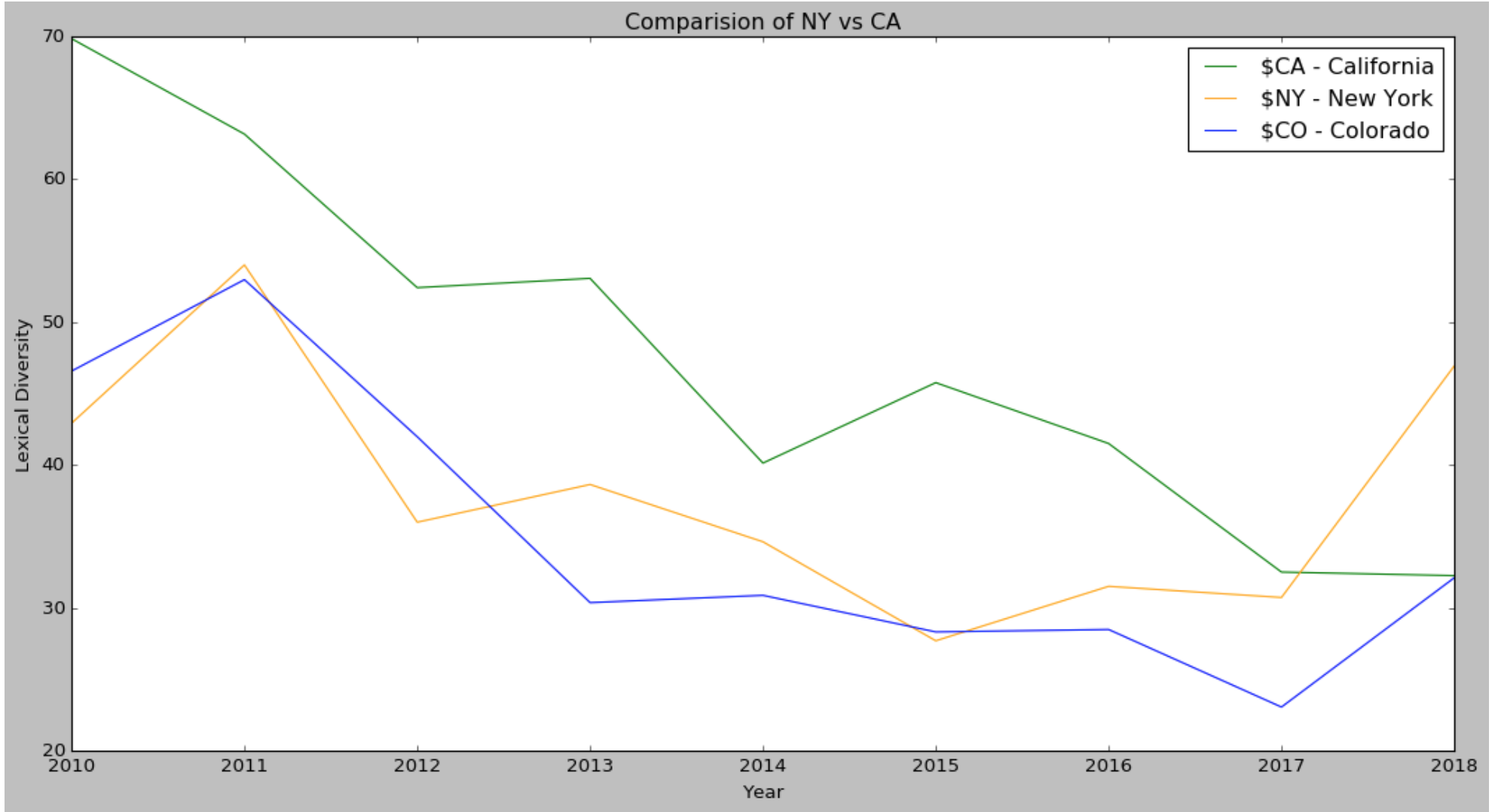
1. Removing Null characters
2. Removing punctuations and numbers from text reviews
3. Lower casing words
4. Tokenize the reviews

## 6.3 Linguistic Analysis

Linguistic researcher can do several analysis with a text corpus. But I am interested to identify usage pattern of english words in the reviews. Please note that there are certain words that are non-english and I haven’t removed them. I have performed following analysis:

### **1. Lexical Diversity**

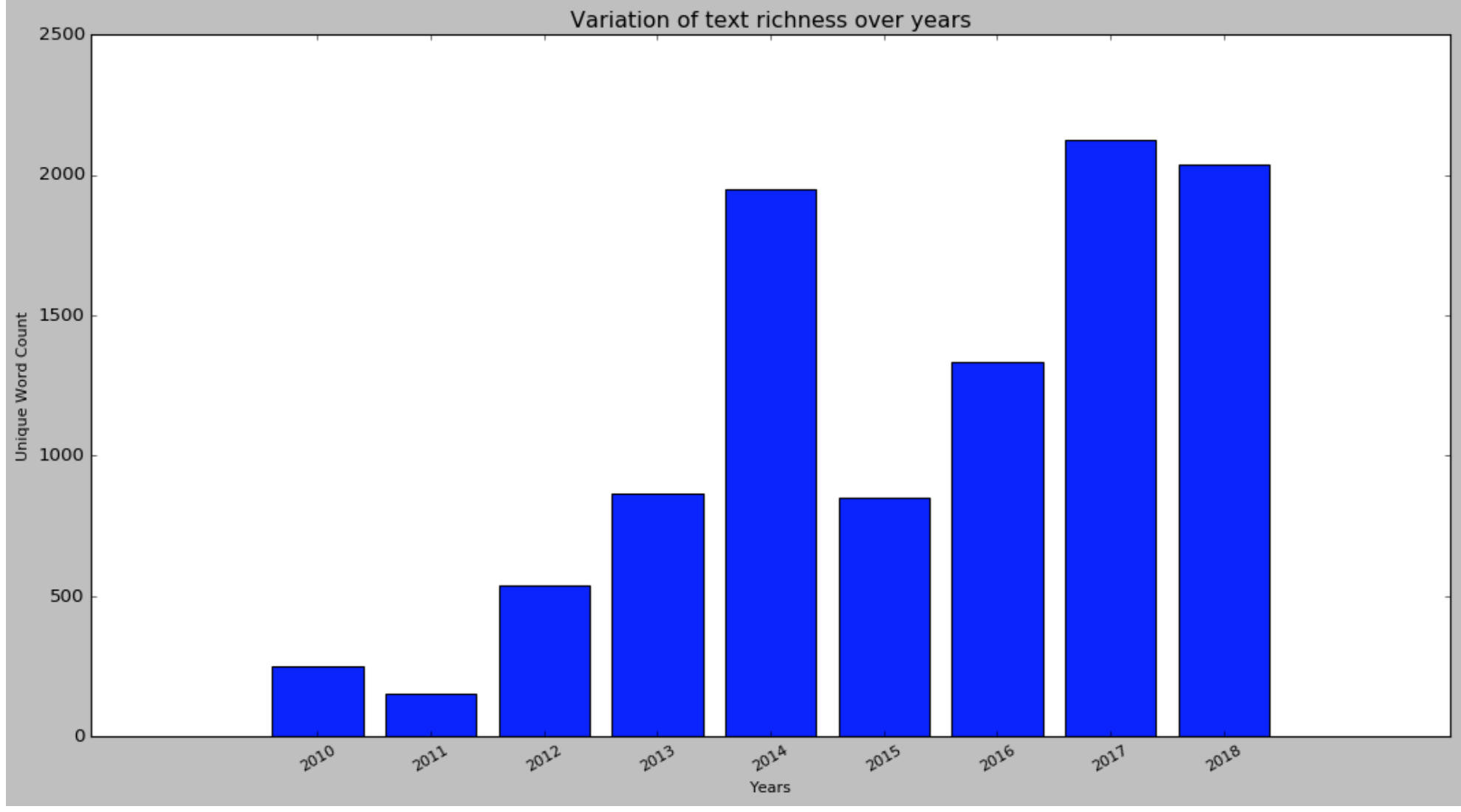
Lexical diversity is a measure of how many different words that are used in a text corpus.



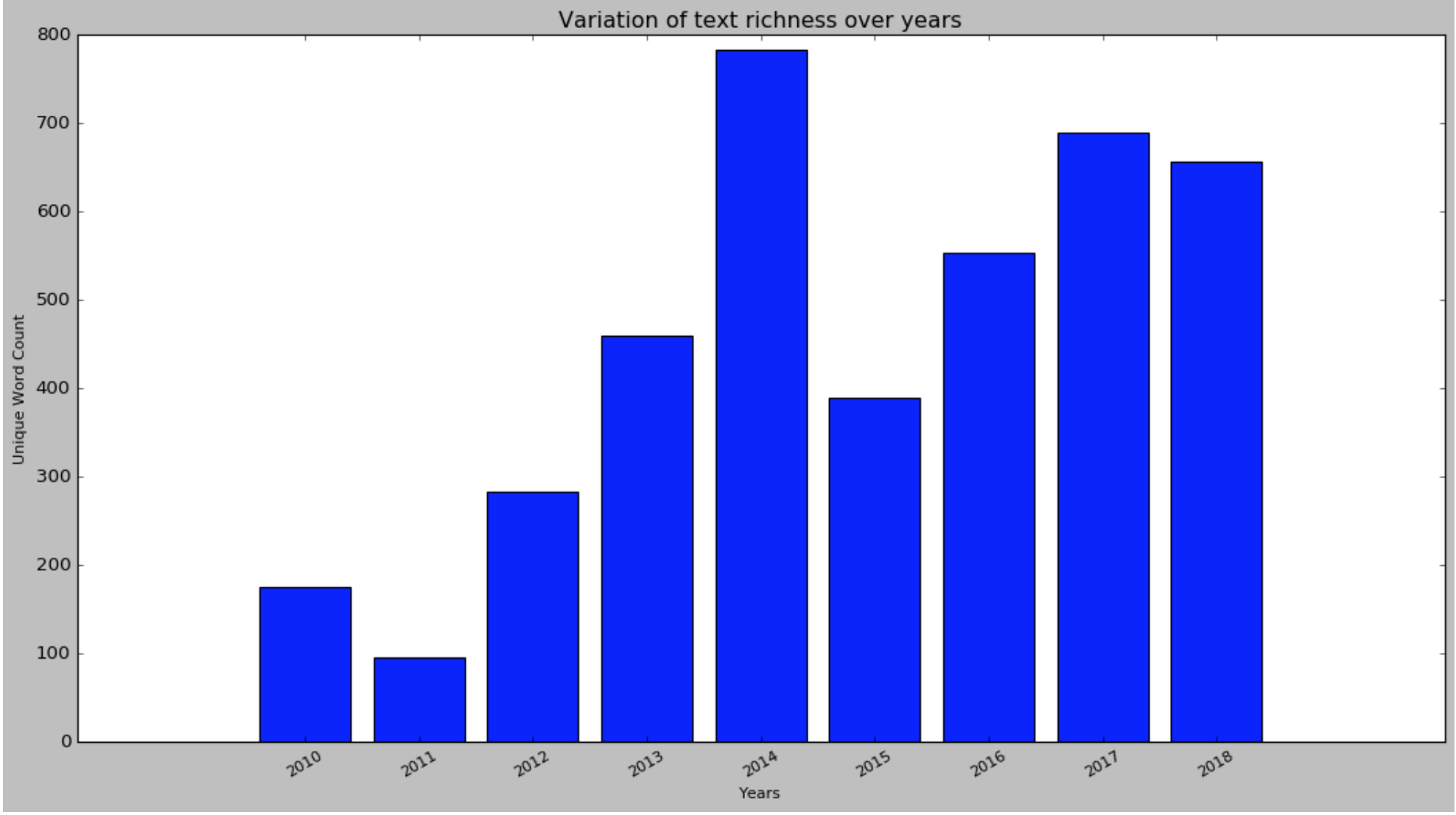
In the graph above I am comparing the variation of lexical diversity(%) over the years between CA and NY. Lexical diversity has consistently decrease over the years for CA but have some ups for NY.

### **2. Words analysis**

Following graphs display variation of unique words for reviews from california corpus for each year from 2010 to 2018. It can be clearly seen that for year 2014 although there was not lot of text but there were many unique words, whereas for year 2017 and 2018 this is not the case.



|  |
| --- |
| *This graph shows the total word count for CA over the years* |



*This graph shows the total unique word count for CA over the years*

### **3. Word Frequency Analysis** (with stop words)

Here I try to find most frequent words with stop words included and then remove the stop words

### **4. Word Frequency Analysis** (without stop words)

I want to identify what are top 15 frequent words in each corpu. Below is a word cloud from 2018 corpus of CA.

|  |
| --- |
| year - **2010**  words - [('ist', 6), ('die', 6), ('man', 6), ('zu', 6), ('es', 5)] year - **2011**  words - [('patricia', 5), ('body', 4), ('massages', 2), ('therapists', 2), ('found', 2)] year - **2012**  words - [('rockabilly', 7), ('vegas', 6), ('people', 6), ('weekend', 5), ('las', 5)] year - **2013**  words - [('e', 9), ('year', 8), ('show', 7), ('di', 7), ('car', 6)] year - **2014**  words - [('one', 13), ('make', 11), ('class', 9), ('us', 9), ('vanessa', 7)] year - **2015**  words - [('time', 4), ('event', 4), ('nothing', 4), ('pay', 4), ('arch', 4)] year - **2016**  words - [('hair', 21), ('kassandra', 14), ('always', 9), ('house', 7), ('amazing', 7)] year - **2017**  words - [('rogers', 18), ('dr', 17), ('back', 16), ('surgery', 14), ('business', 12)] year - **2018**  words - [('dr', 22), ('rogers', 18), ('pain', 17), ('back', 16), ('years', 15)] |

### **5. Complex word count:**

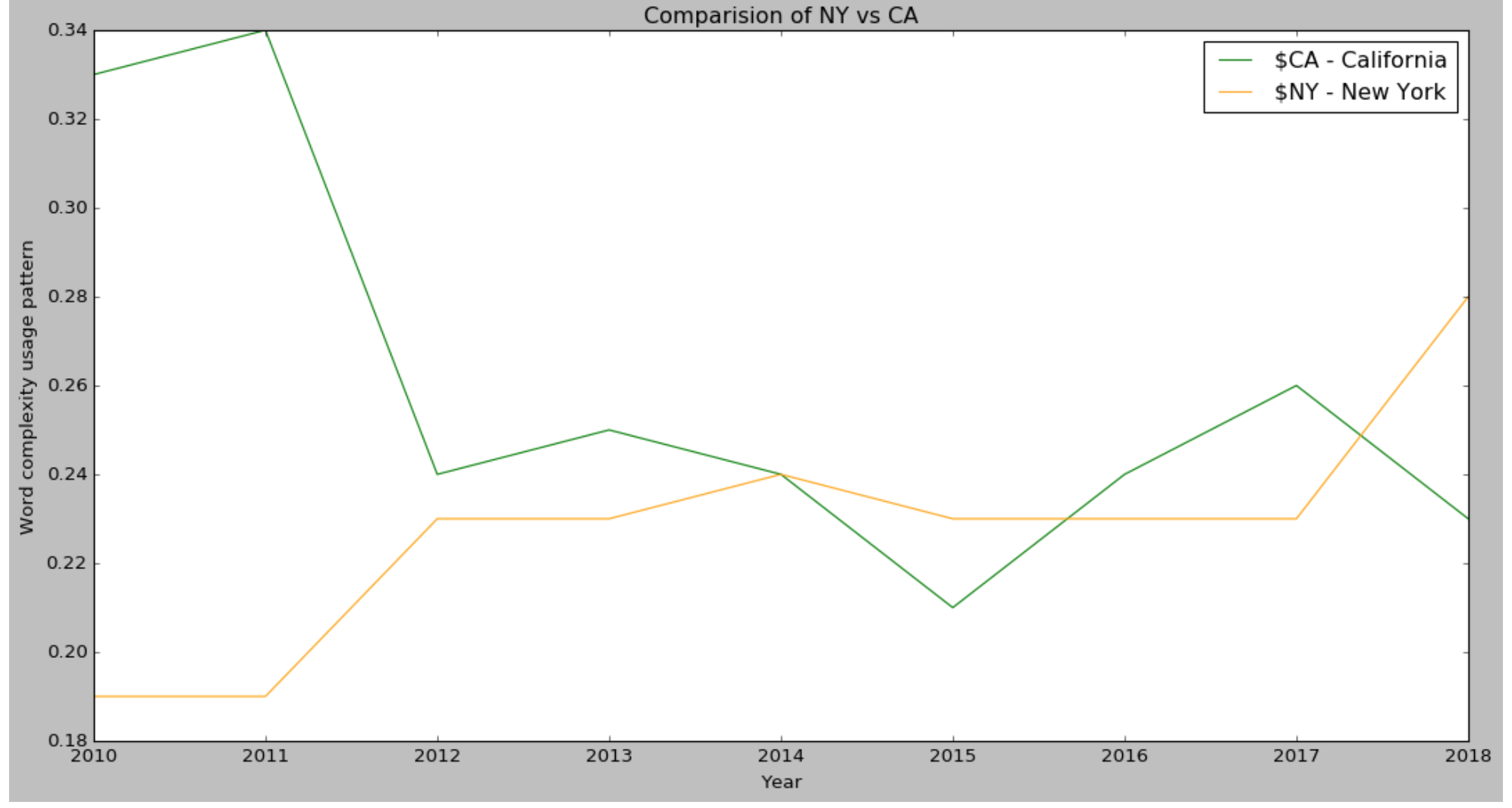
Next I look into usage of complex words over the years. I the following table I show usage of complex words over the years in CA. It can be clearly observed that relative lesser usage of complex words in recent years.

|  |
| --- |
| year - 2010 , all words - 252 , words - 84.0 , %age- 0.33 year - 2011 , all words - 152 , words - 52.0 , %age- 0.34 year - 2012 , all words - 540 , words - 128.0 , %age- 0.24 year - 2013 , all words - 867 , words - 215.0 , %age- 0.25 year - 2014 , all words - 1951 , words - 473.0 , %age- 0.24 year - 2015 , all words - 850 , words - 179.0 , %age- 0.21 year - 2016 , all words - 1335 , words - 325.0 , %age- 0.24 year - 2017 , all words - 2123 , words - 557.0 , %age- 0.26 year - 2018 , all words - 2037 , words - 474.0 , %age- 0.23 |

Following is the pattern for New York

|  |
| --- |
| year - 2009 , all words - 526 , words - 126 , %age- 0.24 year - 2010 , all words - 685 , words - 128 , %age- 0.19 year - 2011 , all words - 413 , words - 78 , %age- 0.19 year - 2012 , all words - 2295 , words - 533 , %age- 0.23 year - 2013 , all words - 2019 , words - 469 , %age- 0.23 year - 2014 , all words - 2065 , words - 498 , %age- 0.24 year - 2015 , all words - 3471 , words - 814 , %age- 0.23 year - 2016 , all words - 2879 , words - 673 , %age- 0.23 year - 2017 , all words - 3363 , words - 780 , %age- 0.23 year - 2018 , all words - 912 , words - 255 , %age- 0.28 |

Following graph shows the pattern of usage of the complex words. Usage of complex words have been on a decrease in California whereas there has been an steady increase in NY.



### **6. Part of Speech Tagging**

In this section I am inspecting the part of speech that are most frequent in the corpus. There are several 30 POS in NLTK but I have tried to show top 5 POS in each corpus. From below results it is clear that NN (Singular noun) is most popular word followed by JJ (Adjective) and IN(preposition). It is obvious as people tend to talk about a subject (eg. restaurant or business) and review them.

Top POS for the CA -

|  |
| --- |
| [[('NN', 152), ('JJ', 43), ('VBP', 18), ('NNS', 11), ('IN', 6)],  [('NN', 29), ('JJ', 19), ('IN', 16), ('NNS', 14), ('DT', 14)],  [('NN', 110), ('DT', 64), ('JJ', 56), ('IN', 49), ('NNS', 40)],  [('NN', 287), ('JJ', 81), ('DT', 76), ('IN', 76), ('NNS', 51)],  [('NN', 409), ('JJ', 178), ('IN', 174), ('DT', 144), ('PRP', 124)],  [('NN', 167), ('DT', 92), ('IN', 82), ('JJ', 67), ('RB', 50)],  [('NN', 327), ('JJ', 102), ('IN', 94), ('DT', 88), ('RB', 85)],  [('NN', 395), ('IN', 228), ('JJ', 175), ('DT', 160), ('PRP', 139)],  [('NN', 370), ('IN', 204), ('JJ', 170), ('RB', 143), ('DT', 142)]] |

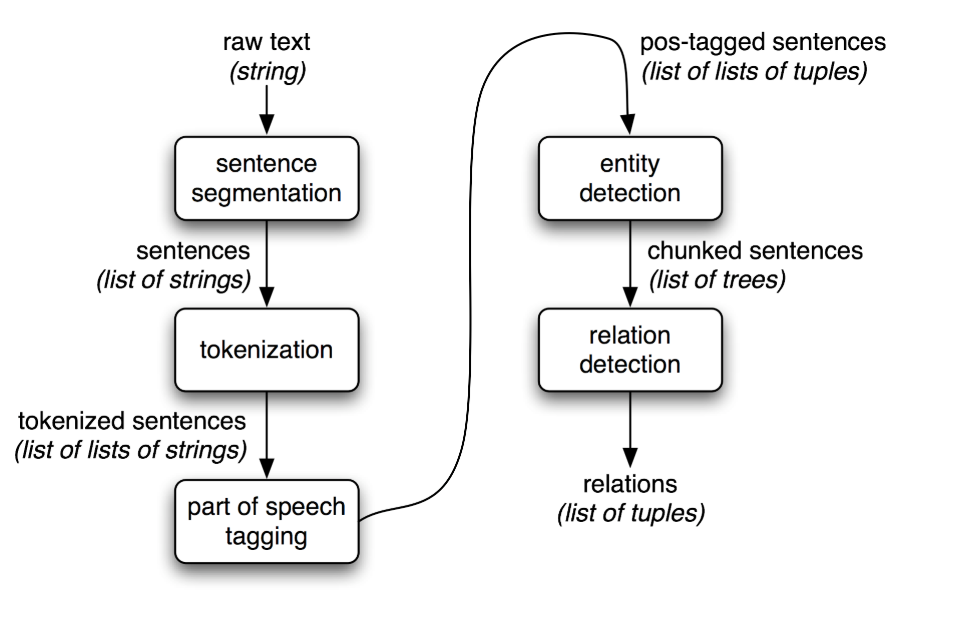
Top POS for the NY -

|  |
| --- |
| [[('NN', 152), ('JJ', 43), ('VBP', 18), ('NNS', 11), ('IN', 6)],  [('NN', 29), ('JJ', 19), ('IN', 16), ('NNS', 14), ('DT', 14)],  [('NN', 110), ('DT', 64), ('JJ', 56), ('IN', 49), ('NNS', 40)],  [('NN', 287), ('JJ', 81), ('DT', 76), ('IN', 76), ('NNS', 51)],  [('NN', 409), ('JJ', 178), ('IN', 174), ('DT', 144), ('PRP', 124)],  [('NN', 167), ('DT', 92), ('IN', 82), ('JJ', 67), ('RB', 50)],  [('NN', 327), ('JJ', 102), ('IN', 94), ('DT', 88), ('RB', 85)],  [('NN', 395), ('IN', 228), ('JJ', 175), ('DT', 160), ('PRP', 139)],  [('NN', 370), ('IN', 204), ('JJ', 170), ('RB', 143), ('DT', 142)]] |

## 6.4 Aspect Identification

I have the following pipeline for Information Extraction

* Sentence segmentation and tokenization, this part was completed during data cleaning steps itself.
* Part of speech tagging of each token in the sentence, this part was done during linguistic analysis.
* Relationship detection, this will be performed with help of Noun Phrase chunking

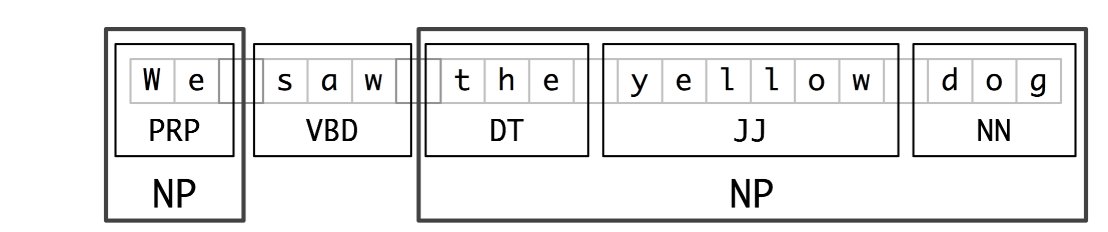


After segmentation and POS tagging, I got a list of tuples containing the individual words in the sentence and their associated part-of-speech. Now, from the POS tag I know that Nouns will the entity. Now to identify the relation I need to implement noun phrase chunking to identify named entities using a regular expression consisting of rules that indicate how sentences should be chunked. Noun phrase chunking, or NP-chunking, where we search for chunks corresponding to individual noun phrases.

|  |
| --- |
| **pattern = 'NP: {<DT>?<JJ>\*<NN>}'**  My chunk pattern consists of one rule, that a noun phrase, NP, should be formed whenever the chunker finds an optional determiner, DT, followed by any number of adjectives, JJ, and then a noun, NN. |

Using this pattern I created a chunk parser and put my sentences through it -

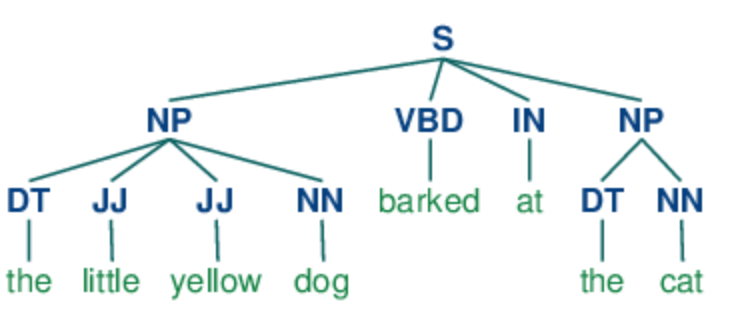
|  |
| --- |
| Example:  sentence = [("the", "DT"), ("little", "JJ"), ("yellow", "JJ"), ("dog", "NN"), ("barked", "VBD"), ("at", "IN"), ("the", "DT"), ("cat", "NN")]  grammar = "NP: {<DT>?<JJ>\*<NN>}" chunk\_parser = nltk.RegexpParser(grammar) tree=chunk\_parser.parse(txt\_pos) return pick\_noun\_phrase(tree) |



*Noun Phrase chunks of the sentence*

Result: The noun Noun Phrase describe the subject. Aspect identification was performed by using NLTK library

|  |
| --- |
| (S  **(NP the/DT little/JJ yellow/JJ dog/NN)**  barked/VBD  at/IN  **(NP the/DT cat/NN))** |



*Tree representing the parsing of the sentence and splitting into NP*

|  |
| --- |
| Following is the list of aspects generated for the NY Corpus for year 2010:  ['a large selection', 'cheap food', 'place', 'the place', 'no problem', 'basic iceberg', 'half', 'the manicotti', 'a good place', 'salad', 'restaurant', 'disgusting soup', 'bowl', 'patio', 'a light lunch', 'a friend', 'pizza', 'life', 'marinara', 'friend', 'a plastic', 'gobs', 'vegetarian pizza', 'average frozen kind', 'coffee', 'someone', 'a split', 'the dressing', 'the soup', 'foot', 'the split', 'the decor', 'roomy fresh modern hip', 'own coffee', 'nice outdoor', 'nothing', 'good i', 'the first visit', 'a sit', 'garlic bread', 'chicken', 'champlain', 'caprese panini', 'a great place', 'silverware', 'yummy sandwich', 'a bad place', 'that fresh hot', 'the kitchen', 'flour', 'the house', 'young clueless', 'free wi', 'a short break', 'sandwich', 'every last crumb', 'spectacular', 'soup', 'loose leaf tea', 'the guy', 'a lot', 'onion', 'sauce', 'dinner', 'potato', 'something', 'homemade', 'combo', 'lunch', 'a steal', 'point', 'cooking', 'a local bakery', 'the whole thing', 'the baked ziti', 'a killer', 'subsandwich', 'the food', 'no creativity', 'flour isn', 'food', 'container', 't feel', 'another day', 'this place', 'the bbq', 'split', 'a half foot', 'the salad', 'pasta', 'a vegetarian pizza']  We can see that the reviews correspond to restaurants in the corpus and describe food and features of the business. |

## 6.5 Entity Identification

Entity identification task is performed using SpaCy library. One of the nice things about Spacy is that we only need to apply nlp once, the entire background pipeline will return the objects. Following are some common entities in text corpus

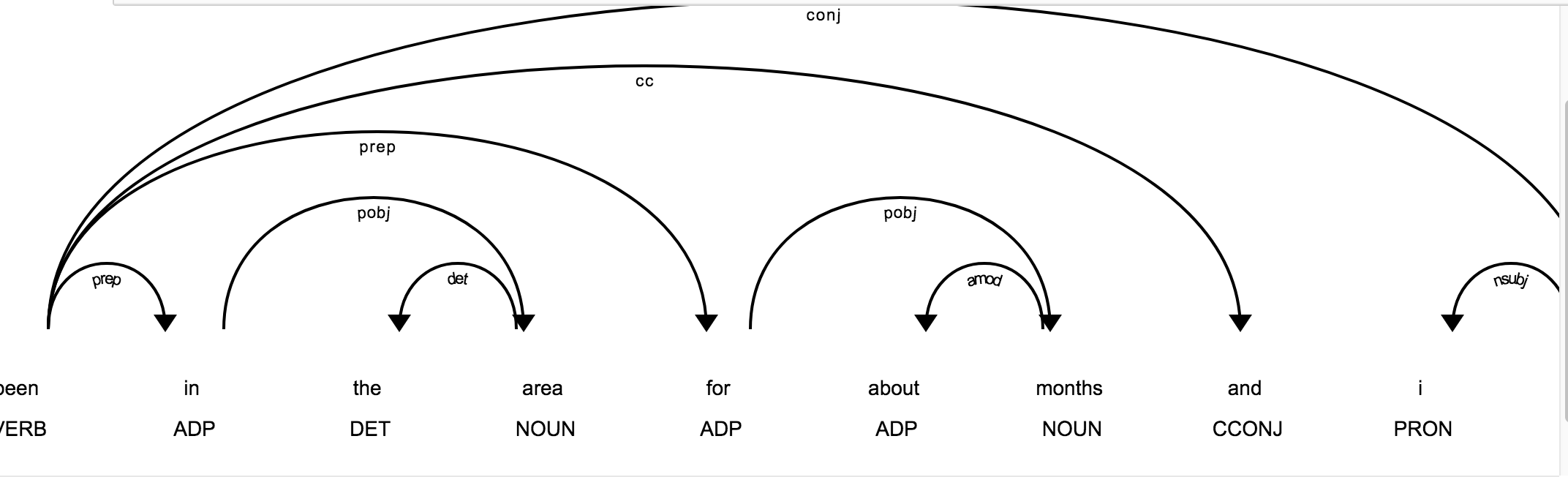
* PERSON People, including fictional.
* NORP Nationalities or religious or political groups.
* FAC Buildings, airports, highways, bridges, etc.
* ORG Companies, agencies, institutions, etc.
* GPE Countries, cities, states.
* LOC Non-GPE locations, mountain ranges, bodies of water.
* PRODUCT Objects, vehicles, foods, etc. (Not services.)

Here is the code to extract the entities from

|  |
| --- |
| def gen\_entity(rev):   entity=[]  count=dict()  # document in spacy format  doc=nlp(" ".join(rev))  # looking through the corpus  for X in doc.ents:  # if X.label\_ in entities:  entity.append((X.label\_,X.text))  if X.label\_ in count.keys():  count[X.label\_]+=1  else:  count[X.label\_]=1  # generate the counter  return (entity,count)  def get\_entity(row):  entity,count=gen\_entity(row['review\_token'])  row['entity']=entity  row['entity\_count']=count  return row |

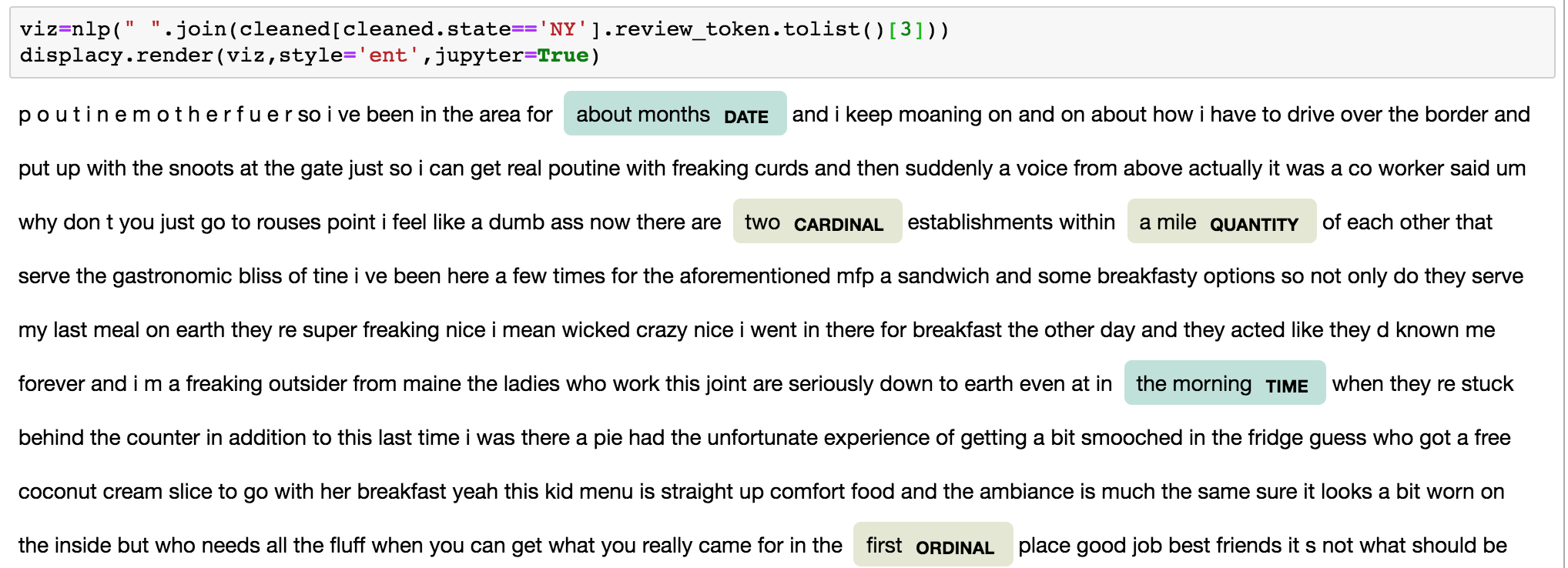
Here I have used the SpaCy library for entity identification. SpaCy model identifies a variety of named and numeric entities, including companies, locations, organizations and products. As the model is statical it doesn’t always work perfect. In the first fig SpaCy represents the relationship between different words based on the part of speech.

|  |
| --- |
| # visualizing entity relations displacy.render(nlp(" ".join(cleaned[cleaned.state=='NY'].review\_token.tolist()[3][20:30])), style='dep', jupyter = True, options = {'distance': 120}) |



*In task 2 I extracted the aspects considering Noun Phrases. If we want to identify other phrases such as Verb Phrases or Clause this visualization is very helpful.*

|  |
| --- |
| # visualizing entities in the sentence viz=nlp(" ".join(cleaned[cleaned.state=='NY'].review\_token.tolist()[3])) displacy.render(viz,style='ent',jupyter=True) |

****

*In the above figure spacy has highlighted certain entities eg, quantity, cardinal time etc.*

# 7 Result

The objective of the project is to analyze the reviews and the changing trends in the language and its usage. As mentioned earlier there has been change in the demography of these states in last one decade. Here are some of the major findings from the project. The success of the project will be measured against the achievement of the following targets -

1. Lexical diversity of the reviews have been on a decline. This can be attributed either to use of simpler words by reviewers or can be attributed to shorter reviews.
2. Usage of unique words has been on a steady increase. As I haven’t checked the individual words this can be attributed to two things - use of better vocabulary or usage or short form of words. Common example of these can be seen today eg.

|  |
| --- |
| and = nd, help = hlp, good = gud, want = wnt. |

All these words become unique words and can cause increase in unique words.

1. Usage of complex words ( words with length > 5) has been on a constant decline. This can be attributed to usage of shortening of complex words by users. NY, CA, and CO all see a slow decline
2. Regarding parts of speech, NN(Noun Singular abstract) nouns are most common nouns in the review corpus. This comes obvious as a review talks about a Proper noun (NNP) and noun (NN) associated with it. Below is the example of a review along with its tags. It is followed by JJ and IN.

|  |
| --- |
| "My first visit here to meet friends for lunch. Nice ambience in downtown Sunnyvale and seems like a popular lunch hangout place. Food is Pan Asian cuisine sort of like Indo-Chinese." |

|  |
| --- |
| [('My', 'PRP$'), (**'first'**, 'JJ'), (**'visit'**, 'NN'), ('here', 'RB'), ('to', 'TO'), ('meet', 'VB'), ('friends', 'NNS'), ('for', 'IN'), (**'lunch'**, 'NN'), ('.', '.'), ('Nice', 'NNP'), (**'ambience'**, 'NN'), ('in', 'IN'), (**'downtown'**, 'JJ'), ('Sunnyvale', 'NNP'), ('and', 'CC'), ('seems', 'VBZ'), ('like', 'IN'), ('a', 'DT'), (**'popular'**, 'JJ'), (**'lunch'**, 'NN'), (**'hangout'**, 'NN'), (**'place'**, 'NN'), ('.', '.'), ('Food', 'NNP'), ('is', 'VBZ'), (**'Pan'**, 'NNP'), ('Asian', 'NNP'), (**'cuisine'**, 'NN'), (**'sort'**, 'NN'), ('of', 'IN'), ('like', 'IN'), ('Indo-Chinese', 'NNP'), ('.', '.')] |

5. Part of speech tags were helpful for generating the Noun Phrase chunking rule.

# 8 Future work

I have completed this project as part of my final project toward LING . I have been able to achieve some early results but there are many more things to analyze in the corpus. Some of the challenges I have faced are:

1. Some words in the business reviews were from foreign languages. As I haven’t removed these from the corpus they impact my analysis.
2. Many of the words written in reviews are shortened form, I haven’t been able to handle those words.
3. People often use smiley icons to represent their emotions, as I have removed punctuations during cleaning of my text corpus.

These are some of the challenges that I faced while working on this project but these can be something I can incorporate in future expansion. I have generated a text corpus for reviews that can be used by other linguistic researchers in future that I plan to host online. Along with text corpus this dataset also holds analysis done by me. In future I can look into patterns such as what % of corpus consist of shortened english word, if there exist non-native words. It would also be an interesting analysis to see what is spoken about certain ubiquitous items such as beer, pizza among restaurant reviews.

# 9 Reference

1. Dataset: <https://www.kaggle.com/yelp-dataset/yelp-dataset>
2. Spacy : <https://spacy.io/usage/processing-pipelines>
3. Nltk : <https://www.nltk.org/book/>
4. Named Entity Recognition with NLTK and SpaCy, Susan Li <https://towardsdatascience.com/named-entity-recognition-with-nltk-and-spacy-8c4a7d88e7da>
5. “A Statistical Model for Multilingual Entity Detection and Tracking. <http://www.aclweb.org/anthology/N04-1001>”, R. Florian, H. Hassan , A. Ittycheriah, H. Jing N. Kambhatla, X. Luo, N. Nicolov, and S. Rouko
6. “Chapter 8. Entity Detection and Tracking”, Xiaoqiang Luo and Imed Zitouni <https://www.oreilly.com/library/view/multilingual-natural-language/9780137047833/ch08.html>
7. “Tokenization and Parts of Speech(POS) Tagging in Python’s NLTK library”, Gianpaul Rachiele <https://medium.com/@gianpaul.r/tokenization-and-parts-of-speech-pos-tagging-in-pythons-nltk-library-2d30f70af13b>