ASSIGNMENT: Review Data Analysis and Processing

[AI6122] Text Data Management and Processing - *2021/2022 Semester 1*

Hantao Li and other group members
HLI038@e.ntu.edu.sg

1 Dataset Analysis

We use the *Yelp Open Dataset* to implement the following analysis. [1] Each review is one line in the JSON file, and each review has the following components: review id, user id, business id, stars, date, text, useful, funny, cool.

1.1 Tokenization and Stemming

1.1.1 Business Selecting First, we randomly select two businesses from the whole database as b_1 and b_2, to form the sub-dataset B_1 and B_2. After a brief reading of the extracted merchant reviews, we can find the names and websites of the two merchants. The b_1 we selected is a hamburger store named *Pdx* in Portland [2], while b_2 is a Chinese restaurant named *NingTu* in Vancouver.[3] First, we counted the average scores of the review components of the two businesses and showed it in Table 1 for subsequent discussion and analysis.

TABLE I
AVERAGE PARAMETERS OF SELECTED BUSINESSES

Name	PDX Sliders	NingTu Restaurant
Number of Reviews	866	69
Stars	4.66	3.94
Useful	0.61	1.59
Funny	0.29	1.13
Cool	0.44	1.12

1.1.2 Stemming Implementation In order to facilitate the analysis, we not only exclude stopwords but also delete punctuation in sentences and convert all letters to lowercase before counting the word frequency distributions. The word distributions of the top-15 frequency in the reviews of the two businesses are shown in Fig. 1 and Fig. 2. The core code of 1.1 and 1.2 are shown below together：

```
token_word = word_tokenize(sentence)   # Tokenization
token_words = pos_tag(token_word)      # POS
words_lematizer = []                   # Lemmatize
wordnet_lematizer = WordNetLemmatizer()
for word, tag in token_words:
    if tag.startswith('NN'):
        word_lematizer =  wordnet_lematizer.lemmatize(word, pos='n')
    elif tag.startswith('VB'):
        word_lematizer =  wordnet_lematizer.lemmatize(word, pos='v')
    elif tag.startswith('JJ'):
        word_lematizer =  wordnet_lematizer.lemmatize(word, pos='a')
    elif tag.startswith('R'):
        word_lematizer =  wordnet_lematizer.lemmatize(word, pos='r')
    else:
        word_lematizer =  wordnet_lematizer.lemmatize(word)
    words_lematizer.append(word_lematizer)
```

```
cleaned_words =    # Stopwords, punctuations, convert all letters to lowercase
      [word for word in words_lematizer if word not in stopwords.words('english')]
characters = [',', '.','DBSCAN', ':', ';', '?', '(', ')', '[', ']', '&', '!', '*', '@', '#', '$', '%','-','...','^',{',}']
words_list = [word for word in cleaned_words if word not in characters]
words_lists = [x.lower() for x in words_list
stemmer = SnowballStemmer("english", ignore_stopwords=True)   # Stemming
words_stemmer = [stemmer.stem(token_word) for token_word in words_lists]
filtered_words = [word for word in words_stemmer if word not in ['n\'t', '\'s']]
freq = FreqDist(filtered_words)    #Plot
freq.plot(15,cumulative=False)]
```

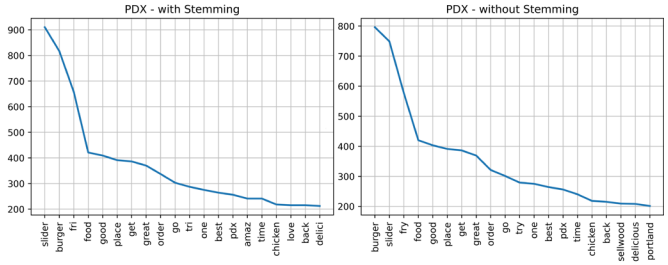

Figure 1: Word frequency distribution of b_1 (PDX Sliders) with and without Stemming.

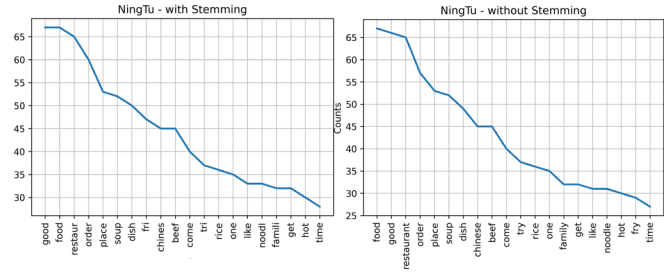

Figure 2: Word frequency distribution of b_2 (NingTu restaurant) with and without Stemming.

1.1.3 Discussion Firstly, analyzing the word distribution before and after Stemming in B_1 separately, we can observe the following characteristics:

1. B_1 is a business that sells hamburgers; most of their dishes are hamburgers or slider (a small hamburger). [4] Therefore, *'burger'* and *'slider'* appear much more frequently than any other words in reviews.
2. Although *'fri'* appears more frequently than *'fry'* due to the change of tense before and after stemming. Since fry is a cooking method directly corresponding to *'burger'* and *'slider'*, its frequency is stable in the third place.

3. According to follow-up observations (1 and 2), it can be noticed that after stemming, the *'sliders'* and *'slider'* are converged, making the number of *'slider'* become more than *'burger.'*
4. Adjectives such as *'good'*, *'great'*, and *'best'* that express simple and direct praise appear a lot both before and after stemming, and their ranking is almost identical.
5. *'amaz-'*, *'delici-'*, and other words expressing praise, but usually utilized with complex morphological changes, their ranking changes significantly before and after stemming.
6. Most verbs in the table are simple verbs that can form many phrases, such as *'get'*, *'go'*, and *'try.'* It is challenging for us to distinguish its specific meaning in the text.

Secondly, by comprehensively observing the result figures of B_1 and B_2, we can find the following characteristics:

1. Since both two businesses are restaurants, *'good'*, *'food'*, *'order'*, and *'place'* appear many times.
2. There is no dish name with a particular quantity advantage similar to B_1 because B_2, as a typical Chinese restaurant, provides a wide variety of food categories. Thus, it is challenging to find a dish name or cooking method with undeniable advantages.
3. Despite the above analysis, the word *'fry'* still appears as the top word, for this cooking method is more prevalent in Chinese food. In addition, words such as *'noodle'*, *'soup'*, *'rice'*, *'hot'*, and *'beef'* also appear in the distribution. Viewing these words alone, we cannot suppose the type of restaurant; however, only Chinese restaurants can have these high-frequency words simultaneously.

Finally, based on the above analysis, we can draw the following conclusions:

1. From the nouns appearing in the review, we can analyze the business type. In these two examples, high-frequency words such as *'food'*, *'order'*, and *'place'* indicate both businesses are restaurants.
2. Whether there are nouns with a specific high frequency in the reviews is directly related to the merchant's characteristics. When a merchant focuses on a specific commodity, such as sliders, a noun with prominent advantages will appear in the review.
3. From the word frequency distribution, whether through the stemming or not, there are always simple verbs with high frequency, such as *'go'*, *'try'*, *'come'*, *'get.'* Nevertheless, while there has no specific context and phrase, these words have no worth in word frequency analysis.
4. After stemming, we can count the number of noun occurrences more accurately to avoid statistical errors. Stemming can combine the singular and plural nouns, avoiding unfair statistics between countable and uncountable nouns.
5. After stemming, the number of the original form of a specific verb increases, while its frequency ranking does not change significantly.
6. Simple adjectives, such as *'good'* and *'great'*, little variation will occur before and after stemming. However, for more complex adjectives, such as *'amazing'*, we are more likely to utilize its variations, such as *'amazement'* and *'amaze'*, the frequency of its root *'amaz-'* will rise significantly after stemming.

1.2 POS Tagging

1.2.1 POS Tagging We select five complete sentences in B_1 and then implement the POS Tagging on them using two methods. The first method we use is the pos_tag() function in the NLTK library [5]; The second way is using the POS function contained in the CoreNLP, released by Stanford University. [6] In order to make the selected sentences reasonably examine the results of POS Tagging, we adopt the following strategies when selecting five sentences: Firstly, select 3 sentences with increasing grammatical complexity and length while having the correct grammar. Secondly, select 1 sentence expressed with style on the internet. Thirdly, select 1 sentence with grammatical problems.

TABLE II
POS TAGGING RESULTS*

If	*you*	*think*	*you*	*have already tried*	*the best burger,*	*you*	*are*	*mistaken.*				
IN	PRP	VBP	PRP	VBP	RB	VBN	DT	JJS	NN	PRP	VBP	**VBN**
IN	PRP	VBP	PRP	VBP	RB	VBN	DT	JJS	NN	PRP	VBP	**JJ**
Two	*of*	*my*	*favorites*	*are*	*the Hawthorne and the Steel --- I*				*'m*			
CD	IN	PRP$	NNS	VBP	DT	NNP	CC	DT	**NNP**	PRP	VBP	
CD	IN	PRP$	NNS	VBP	DT	NNP	CC	DT	**NN**	PRP	VBP	
salivating	*just*	*thinking*	*about*	*them!*								
VBG	RB	VBG	IN	PRP								
VBG	RB	VBG	IN	PRP								
I	*hope*	*to*	*have*	*PDX Sliders*	*cater*	*every*	*remaining*	*significant*	*event*	*in*		
PRP	VBP	TO	VB	**NNP**	**NNP**	**NN**	DT	VBG	JJ	NN	IN	
PRP	VBP	TO	VB	**NN**	**NNS**	**VBP**	DT	VBG	JJ	NN	IN	
my	*life,*	*up*	*to,*	*and*	*including,*	*my*	*funeral*	*(it*	*'s*	*the*	*only*	*way*
PRP$	NN	**RB**	**TO**	CC	VBG	PRP$	**JJ**	PRP	VBZ	DT	JJ	NN
PRP$	NN	**IN**	**IN**	CC	VBG	PRP$	**NN**	PRP	VBZ	DT	JJ	NN
people	*will*	*be*	*able*	*to*	*swallow*	*my*	*demise)...*					
NNS	MD	VB	JJ	TO	VB	PRP$	NN					
NNS	MD	VB	JJ	TO	VB	PRP$	NN					
Omg	*their*	*fries*	*are*	*THEEE BEST!*								
NNP	PRP$	NNS	VBP	**NNP**	**NNP**							
NNP	PRP$	NNS	VBP	**JJ**	**JJS**							
Our	*go*	*to*	*on date night*	*and*	*when*	*friend*	*come*	*in*	*to town.*			
PRP$	NN	**TO**	IN	NN	NN	CC	WRB	NN	**VBP**	IN	**TO**	NN
PRP$	NN	**IN**	IN	NN	NN	CC	WRB	NN	**VBN**	IN	**IN**	NN

*Both of the two POS method use the *Penn Treebank* site as the POS tag set. [7] Due to the length, we will not show the specific meaning of each label in the report.

In the above content, we present the selected five sentences and the results of POS Tagging in Table 2. NLTK is represented in blue, while CoreNLP is represented in green. The difference between them is highlighted in bold.

1.2.2 Discussion We can notice the following characteristics from the POS results listed in Table 2:

1. In most cases, the accuracy of POS results is acceptable. Where having discrepancies between the two methods, the results of CoreNLP are correct in most cases compared with the NLTK.
2. With the increasing complexity of sentence patterns, the probability of disagreements between the two methods will also increase. In simple sentences, POS results are often correct and the same.
3. NLTK may misjudge some complex expression methods. For example, when *'mistaken'* has both adjective and verb forms, NLTK confuses passive verbs and adjectives.
4. POS does not perform well to determine nouns and proper nouns, which is understandable, while even humans cannot distinguish NN and NNP properly without knowing in advance.
5. When there is a networked informal expression in the sentence, such as *'THEEE BEST'*, the outcome of POS is inferior. The *'THEEE'* as a determiner is not correctly pointed out under both POS. NLTK even put the two words together as a proper noun.
6. When there is a syntax error in the original sentence, neither POS method can tag it well. The reason is apparent that it is confusing for even humans to tag grammatically wrong sentences.

Consequently, we can draw the following conclusions for the above characteristics:

1. Most labeling results of NLTK and CoreNLP are accurate. In comparison, CoreNLP is better than NLTK in dealing with complicated and error-prone parts.
2. The two methods have a poor judgment for nouns and proper nouns.
3. For prevalent internet words, neither method can identify them.
4. For wrong syntax, the two methods cannot distinguish them.
5. For simple sentences without complex sentence patterns, the result of POS can be considered correct; however, when POS is performed on complex sentences, there will be an uncertainty of the result. We cannot assume that the result of POS will be accurate.
6. Similarly, when we are not convinced that there are no prevalent internet words or wrong syntax in the text, the result of POS cannot be considered correct without inspection.

CONTRIBUTIONS

Hantao Li: Tokenization and Stemming; POS Tagging; Consolidated Report; Reformatted Report and README.
Other group members.

REFERENCES

[1] https://www.yelp.com/dataset
[2] https://www.yelp.com/biz/pdx-sliders-portland?osq=pdx
[3] https://www.yelp.ca/biz/ningtu-restaurant-vancouver
[4] https://www.pdxsliders.com/menu
[5] http://www.nltk.org/
[6] https://stanfordnlp.github.io/CoreNLP/
[7] Beatrice Santorini. 1990. Part-of-Speech Tagging Guidelines for the Penn Treebank Project (3rd Revision, 2nd printing).