Product Review Data Analysis and Processing

Report for CZ4045 – Natural Language Processing

Genevieve Lam Wen Qi
C150110@e.ntu.edu.sg

Kuan Ji Jie
C150151@e.ntu.edu.sg

Lim Ming Wei
mlim045@e.ntu.edu.sg

Tan Hao Hao
tanh0207@e.ntu.edu.sg

INTRODUCTION

This report consists of 4 main sections. In the first section, we show our results for data analysis on the product review data given. In the second section, we demonstrate our implementation of noun phrase summarizer. In the third section, we discuss on our heuristic-based approach on implementing a sentiment detector. In the fourth section, we show an application on the product review data using topic modelling.

1 Dataset Analysis

1.1 Popular Products and Frequent Reviewers

In this part, we identify the top-10 products that attract the most number of reviews, and the top-10 reviewers who have contributed most number of reviews.

The CellPhoneReview.json is first passed into a pandas dataframe object. Using the pandas dataframe functions, the same product id / user id will first be grouped together and then count the number of reviews for each group. The result is sorted based on the review count in descending order. From the JSON data given, 'asin' is used as the product id and 'reviewerID' is the user id.

The results are shown below:

Top 10 Products with Most Reviews		
	Product ID	Review Count
1	B005SUHPO6	836
2	B0042FV2SI	690
3	B008OHNZI0	657
4	B009RXU59C	634
5	B000S5Q9CA	627
6	B008DJIIG8	510
7	B0090YGJ4I	448
8	B009A5204K	434
9	B00BT7RAPG	431
10	B0015RB39O	424

Top 10 Reviewers with Most Reviews		
	User ID	Review Count
1	A2NYK9KWFMJV4Y	152
2	A22CW0ZHY3NJH8	138
3	A1EVV74UQYVKRY	137
4	A1ODOGXEYECQQ8	133
5	A2NOW4U7W3F7RI	132
6	A36K2N527TXXJN	124
7	A1UQBFCERIP7VJ	112
8	A1E1LEVQ9VQNK	109
9	A18U49406IPPIJ	109
10	AYB4ELCS5AM8P	107

1.2 Sentence Segmentation

The sentence tokenizer in NLTK uses a pre-trained Punkt sentence tokenizer. Punkt is expected to recognize that the periods in 'Mr.' or 'Ms.' do not mark sentence boundaries. However, as Punkt is pre-trained on some corpora, it is domain dependent and hence might not function well when trying to tokenize more creatively designed reviews.

From Figure 1 it shows that a large majority of reviews have about 0 - 50 sentences. This is evident from the peak at the start of the graph. The distribution is shown in the graph is reasonable as most reviewers tend to keep their reviews short yet informative, writing just enough to describe the product without it being lengthy.

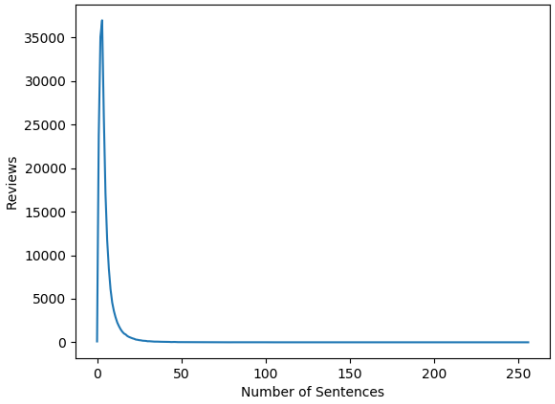

Figure 1: Reviews against number of sentences

From the 5 random samples that were taken, it shows that the sentence segmentation is rather accurate. In item 1 of Appendix, the review is shown first followed by the list structure which contains the segmented sentences together with the number of sentences identified.

1.3 Tokenization and Stemming

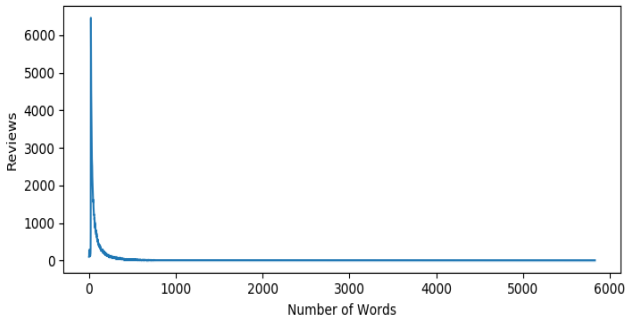

Figure 2.1: Reviews against number of words with stemming

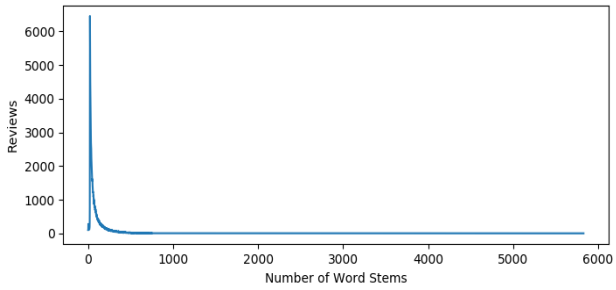

Figure 2.2: Reviews against number of words without stemming

Comparing both graphs in Figure 2.1 and Figure 2.2, both are identical and this result is expected as stemming does not reduce the number of words appearing in a review, it only replaces each word with its respective morpheme. It is also expected that after stemming the total number of unique words will decrease.

Top 20 Words without Stemming		
	Word	Count
1	phone	174345
2	case	144658
3	n’t	116215
4	‘s	97796
5	one	85413
6	like	71795
7	great	65970
8	use	60771
9	screen	59487
10	would	58738
11	good	57855
12	battery	57135
13	well	49465
14	iphone	47732
15	get	46324
16	charge	44390
17	charger	38170
18	really	37971
19	product	37683
20	also	36167

Top 20 Words with Stemming		
	Word	Count
1	phone	189479
2	case	163276
3	use	116695
4	n't	116215
5	's	97796
6	charg	91180
7	one	90926
8	like	79625
9	work	75515
10	great	66009
11	batteri	65076
12	get	61102
13	screen	61067
14	would	58738
15	good	58073
16	look	51808
17	fit	49914
18	iphon	49900
19	well	49476
20	time	46971

From the above results, stemming causes the counts for each word to be increased. Previously the count of the word ‘case’ was recorded at 144658 while after stemming it has increased to 163276. This is because words such as ‘cases’ and ‘casing’ has the same stem ‘case’, thereby increasing its count.

The stop words removed are taken from those specified in the NLTK library. The list of stop words are can be found in item 3 of Appendix.

1.4 POS Tagging

The NLTK library uses the Penn Treebank POS tag set as shown in item 4 of Appendix.

Tagging results are by the POS Tagger are fairly accurate and are shown in item 2 of Appendix. One instance of a wrongly assigned POS tag is that ‘5staramerica’ should be tagged with ‘NNP’ as it is a name of a company but it was tagged as ‘CD’. This might be because of the digit ‘5’ which confuses the tagger to assign it a ‘CD’ tag. Another instance is the inconsistency in tagging the abbreviation ‘usb’ within the same sentence which is sometimes assigned a ‘JJ’ tag while it should have been ‘NN’.

2 Development of a Noun Phrase Summarizer

The method used to extract noun phrases is called chunking. Chunking is a process of extracting phrases from unstructured text. In chunking, it takes into consideration a number of words instead of just one. An example would be to use phrases such as “South

Africa" as a single word instead of 'South' and 'Africa' as separate words.

Chunking works on top of POS tagging, hence POS tags are taken as input and chunks are provided as output. Similar to POS tags, there are a standard set of chunk tags like Noun Phrase (NP). Chunking is used for extract information from text such as location tagging and named entity extraction (NME), which is a common research area in NLP.

Here, we implement a noun phrase summarizer to identify and extract noun phrases from each review. In our context, a noun phrase are words that consist of the following structure: One or more adjective and a noun or more , an adverb a verb and one or more singular proper noun(s), a verb and one or more singular proper noun(s), or lastly an adverb and one or more singular proper noun(s), proper noun and noun. The regular expression to use for extraction is as below:

NP : {<JJ>*<NN>}
NPs : {<RB.?>*<VB.?>*<NNP>+ <NN>?}

```
grammar = ('''

    NPs:{<RB.?>*<VB.?>*<NNP>+}

    NP: {<JJ>*<NN>} # NP

    ''')
chunkParser = nltk.RegexpParser(grammar)

better_tree=[] # PatternTagger
products[product_review3,product_review2,product_review1]

for counter,product in enumerate(products):
       better_tree=[]
       for texts in product_review3:
           tokens = nltk.word_tokenize(texts)
           words = [word for word in tokens if word.isalpha()]
       #     print(tagged_word)
           pattern_tagger = PatternTagger()
           extractor = ConllExtractor()
           tagged_word_2 = TextBlob(' '.join(words), pos_tagger=pattern_tagger,
                                    , np_extractor=extractor)
           better_tree.append(chunkParser.parse(tagged_word_2.tags))

       np_trees=[]

       for tree in better_tree:
           for subtree in tree.subtrees(filter=lambda t: t.label() == 'NP'
                                        or t.label() == 'NPs' ):
               if len(subtree)>1:
                   np_trees.append(subtree)
```

Figure 3: Code snippet for noun phrase summarizer

Single word nouns are removed after extraction. We did not take in phrases that comprises of a singular noun (e.g. phone), or a determiner and a noun (e.g. a phone) as noun phrases, because these phrases are not meaningful for our analysis.

The results for top noun phrases is showed in the table below:

Top Noun Phrases		
Rank	**Noun phrases**	**Count**
1	Same time	3635
2	Long time	2209
3	micro USB	2088
4	Good quality	2010
5	Great product	1923
6	Little bit	1877
7	Great case	1783
8	First time	1743
9	Great price	1704
10	High quality	1663
11	Full charge	1624
12	Hard plastic	1609
13	New phone	1557
14	External batteries	1346
15	Good case	1324
16	Good product	1307
17	Good protection	1223
18	Big deal	1207
19	Nice case	1157
20	Other end	1064

The three popular products with the largest number of reviews are B005SUHPO6, B0042FV2SI and B008OHNZI0. The following are the top 10 noun phrases for each product:

Top Noun Phrases for B005SUHPO6		
Rank	**Noun phrases**	**Count**
1	OtterBox Defender	24
2	Hard plastic	23
3	Great case	21
4	Great product	20
5	Great protection	18
6	Protective case	14
7	Good case	12
8	Outer rubber	12
9	Hard case	12
10	Clear plastic	10

Top Noun Phrases for B0042FV2SI		
Rank	**Noun phrases**	**Count**
1	Great product	10
2	Other screen	10
3	Great price	10

4	Good screen	8
5	First time	7
6	Good quality	6
7	Long time	6
8	Big deal	5
9	Little bit	5
10	New phone	5

Top Noun Phrases for B008OHNZI0		
Rank	**Noun phrases**	**Count**
1	Great product	20
2	Other screen	20
3	First time	18
4	Great screen	14
5	High quality	11
6	Good quality	10
7	Good product	9
8	Good screen	8
9	Clear screen	7
10	Long time	6

The 5 sentences randomly chosen are as follows. The ground truth, wrong and missed annotation are shown.

1) *When there is no outlets, or chargers nearby its Powerbear to the rescue! Ordered one for my husband, and myself. Great purchase!!*
   I. Great purchase

2) *it worked for the first week then it only charge my phone to 20%. it is a waste of money.*
   I. first week

3) *Good case, solid build. Protects phone all around with good access to buttons. Battery charges with full battery lasts me a full day. I usually leave my house around 7am and return at 10pm. I'm glad that it lasts from start to end. 5/5",*
   I. Good case
   II. solid build
   III. solid build protects – wrong annotation
   IV. Good access
   V. Full battery
   VI. Full day

4) *Just what I needed. I needed a phone case for myself and my two sons, but I also needed new replacement batteries. Now this isn't the case, since I got both in one. Awesome thanks A+.*
   I. Phone case – missed annotation
   II. New replacement

5) *This is a fantastic case. Very stylish and protects my phone. Easy access to all buttons and features, without any loss of phone reception. But most importantly, it double power, just as promised. Great buy*
   I. Fantastic case
   II. Easy access
   III. Double power
   IV. Great buy – missed annotation
   V. promise great – wrong annotation

From the results above, we can see that there are 2 false positives (promise great, solid build protects) as well as 2 false negatives (phone case, great buy).

Recall = true positive / (false negative+ true positive)

= 11 / (11 + 2)

= 0.845

Precision = true positive / (true positive + false positive)

= 11 / (11 + 2)

= 0.845

F1 score = 2 * (precision * recall) / (precision + recall)

= 0.845

Errors made by our summarizer are mainly due to (1) error in POS tagging made by our tagger, and (2) removal of punctuations, because our summarizer could account for sentence boundaries when the full stops are removed.

3 Sentiment Word Detection

In determining the sentiment of a word, sentences are first tokenized, tagged and stemmed. Tokenizing is done using NLTK's TreebankTokenizer and stemmed with NLTK's SnowballStemmer. Tagging is applied to tokens first to remove words that do not potentially hold sentiment; tags such as nouns, pronouns, modals and preposition were removed before stemming. The averaged perceptron tagger in NLTK is used without training.

Tokens are grouped into word stems and then the frequency is counted according to the type (reviewText and summary) and rating. A score is then applied to determine the overall sentiment of the word, and 20 words with the heaviest positive and negative sentiments are tabulated.

Before determining the sentiment score of each word, certain heuristics are considered:

1. 5-star and 1-star ratings carry more weight than 4-star and 2-star respectively. (Score with respect to rating is not linear)
2. Although a word may appear equally for positive and negative reviews, it may still be calculated as a positive or negative word if there are far more of one type of review over the other.
3. Words that occur too frequently may not hold high significance.

```
def calculate_freq_set(tset,inv_bank):
	freq_dataset = {}
	key_errors = 0
	for iter_string in tqdm(tset):
		item = iter_string[0]
		rating = iter_string[1]
		token = iter_string[2]

		stem = inv_bank[token]
		if stem not in freq_dataset.keys():
			generate_word_key_dict(stem,freq_dataset,
				items_to_process = items_to_process)
		try:
			freq_dataset[stem][item][rating] += 1
		except KeyError:
			key_errors+=1

	if key_errors >0 :
		print('%d key errors exist, check code and data')
	return freq_dataset
```

Figure 4: Code snippet for calculating word frequency

```
def compute_score(word,item_type, rating, count):
	f,g,h,i = 1,1,1,1
	#string type weight
	f = string_type_weights[item_type]

	#basic sentiment weight
	try:
		g = rating_score[float(rating)]
	except KeyError:
		g = rating_score[rating]

	#inverse count weight/ inverse document frequency
	h = word_inverse_score[word]

	#distribution by score weight
	# i = word_rating_distribution[word][rating]

	return f*g*h*i*count
```

Figure 5: Code snippet for compute score

Words are then scored according to their stems with the following equation:

$$\sum_{r=1}^{5} w_r t \frac{\left[(1+\ln(f))\left(ln\frac{N}{f}\right)\right]}{\ln(N)}$$

where w_r = adjusted weight with respect(w.r.t.) to rating
r = rating of review where the word is found
t = rating type weight ('reviewText' or 'summary')
f = term frequency
N = total number of words in dataset

3.1 Equation Components

W_r

The adjusted weight given to a word depending on its rating. The weights were first defined with a default value : 2.5 for 5-star, 1.5 for 4-star, 0.5 for 3-star, -1.5 for 2-star and -2.5 for 1-star.

A 3-star rating is given a nominal value of 0.5 as it is thought that they carry a slightly positive sentiment, also, giving a weight of 0 would eliminates them from calculation.

To obtain the adjustment for the weights, the normalized mean of words in the dataset are first calculated:

$$\frac{\sum r f_r}{\sum r_{max} f_r}$$

where r = rating
f_r = frequency w.r.t. rating
r_{max} = maximum rating

Given a scenario where a word occurs once across all ratings, the normalized mean score of this word should approximate to 0.04.

The weight w_r is then adjusted by the difference in the normalized mean and target mean of 0.04.

For this dataset, the normalized weight obtained was 5-star = 1.1, 4-star = 0.66, 3-star = 0.22, 2-star = -2.34, 1-star = -3.9.

```
document_norm_mean: 0.6
goal_norm_mean: 0.040000000000000036


{5: 1.1, 4: 0.6600000000000001, 3:
0.2200000000000003, 2: -2.34, 1:
-3.900000000000000004}
```

Figure 6: Normalized weights for each rating

As negative sentiments are weighted more heavily than positive sentiments, it can be deduced that there are much more words in positive ratings as compared to negative ones.

t

As the summary is the first part of the review that is read, it makes sense that reviewers would put words that relay the intended sentiment here. Hence, words found in the summary are given a higher weight.

We give a weight of 2.5 for words in the summary and 1.0 for words in the review text.

$$\frac{\left[(1+\ln(f))\left(ln\frac{N}{\bar{f}}\right)\right]}{\ln(N)}$$

This is a form of term frequency – inverse document frequency (tf-idf) which penalizes words that occur too frequently. The result is normalized by ln(N) to prevent it from being too large.

```
def normalize_rating_weights(array,score_key_name = 'overall'):
        #mean when equal distribution of comments over ratings
        max = max_overall

        wvalues = []
        for key,value in rating_score.items():
                wvalues.append(value)
        wmax = npmax(wvalues)
        norm_wvalues = [i/wmax for i in wvalues]
        goal_mean = mean(norm_wvalues)

        word_count_by_rating = calculate_word_count_by_rating(array)
        values = []
        norm_values = []
        for rating,count in word_count_by_rating.items():
                f = float(rating)*count
                values.append(f)
                norm_values.append(f/(count*max))

        # max = np.max(values)
        # norm_values = [i/max for i in values]
        norm_mean = mean(norm_values)

        alpha = norm_mean - goal_mean
        for rating,weight in rating_score.items():
                if (weight >0):
                        if(alpha>0):
                                rating_score[rating] = (1-abs(alpha))*weight
                        else:
                                rating_score[rating] = (1+abs(alpha))*weight
                elif(weight<0):
                        if(alpha>0):
                                rating_score[rating] = (1+abs(alpha))*weight
                        else:
                                rating_score[rating] = (1-abs(alpha))*weight
```

Figure 7: Code snippet for normalizing rating weights

```
for word,count in word_occurences.items():
        tf = 1+log(count)
        idf = log(total_words/count)
        #normalized tf-idf
        word_inverse_score[word] = (tf*idf)/log(total_words)
```

Figure 8: Normalization of frequency using tf-idf

The results of the analysis (after stemming) are listed as below:

Top 20 Positive Sentiment Words		
	Word	**Score**
1	great	364040
2	good	171847
3	love	171285
4	use	164739
5	charg	123026
6	nice	118705
7	easi	84906.7
8	best	82853.4
9	need	74174.1
10	perfect	73076
11	littl	68379
12	look	65356.3
13	protect	63209.6
14	usb	60949.5
15	recommend	55427.3
16	excel	54394.5
17	work	53387.5
18	awesom	41684.5
19	keep	39884.9
20	want	39181.3

Top 20 Negative Sentiment Words		
	Word	**Score**
1	poor	-31183.3
2	return	-29984.2
3	disappoint	-22437.1
4	broke	-20482.7
5	cheap	-19067.4
6	horribl	-18537
7	terribl	-18366.8
8	bad	-18289.8
9	wast	-15643.3
10	stop	-13372.2
11	pay	-10794
12	worst	-10390.2
13	send	-8922.71
14	defect	-8774.63
15	useless	-7904.36
16	fell	-6386.65

17	sent	-6064.02
18	broken	-5970.47
19	fail	-5752.77
20	wors	-4903.03

While most of the negative sentiment words made logical sense. Some of the positive sentiment words did not. Words like 'use' was derived from a mix of neutral words such as 'using' or 'used' and positive words such as 'useful'. Despite the removal of nouns and stopwords, words such as 'usb' made it through the filters and obtained a high score. Without training the perceptron tagger used for tagging, the tagger could not recognize the word and tagged it as an adjective. ('JJ')

A list of stems in the top 20 list and their constituent tokens can be found in results.txt together with the source code.

4 Application

4.1 Aspect Extraction using Topic Modelling

As an application based on the dataset, we implement aspect extraction on the product reviews using topic-modelling techniques. The purpose of aspect extraction is to:

(1) Find out which aspects are covered by the review;

(2) Evaluate the performance of each aspect based on the ratings given in the reviews.

Topic modelling applies statistical modelling on a collection of documents in order to organize and summarize them based on their implied topics. Here, we choose Latent Dirichlet Allocation (LDA) as our topic model.

LDA is an unsupervised, generative, probabilistic method for topic modelling on a corpus. It assumes that each document can be represented as a probabilistic distribution over latent topics, and each latent topic in the LDA model is also represented as a probabilistic distribution over words. The words with highest probabilities in each topic usually give a good idea of what the topic is summarizing.

LDA assumes that documents are generated in the following sequence:

(1) Decide on the number of words the document have;

(2) Choose a probability distribution of the topic mixture of the document;

(3) Generate each word in the document by first picking a topic according to the distribution chosen in (2), and using that topic to generate the word.

With this assumption, the model learns by first randomly assign each word in the document to one of the topics. With this assignment, we can have the topic representations of all documents and the word distributions of all topics at the same time. We can further improve this assignment by improving two probabilities, (i) P(word | topic) for each word and topic, and (ii) P(topic | document) for each topic and document. This can be done by methods such as Gibbs sampling, Expectation-Maximization algorithm (EM) or variational Bayes inference.

The steps for topic modelling are as follow:

(1) Extract term frequency from given data.

(2) Construct the LDA model and train it.

(3) Extract 5 topics from the dataset with 6 top words each.

(4) Assign a topic for each review.

(5) Evaluate the assignment by randomly sample 20 reviews from the dataset and manually annotate their topics.

(6) With the ground truth, calculate precision, recall and F1 score.

(7) Find the average rating for each topic for performance analysis.

```
tf_vectorizer = CountVectorizer(max_df=0.95, min_df=2, max_features=no_features,
                                stop_words='english')
tf = tf_vectorizer.fit_transform(documents)
tf_feature_names = tf_vectorizer.get_feature_names()

no_topics = 5

# Run LDA
lda = LatentDirichletAllocation(n_topics=no_topics, max_iter=5, learning_method='online',
                                learning_offset=50., random_state=0, verbose=1).fit(tf)

no_top_words = 6

lda_topics = get_topics(lda, tf_feature_names, no_top_words)
w = lda.transform(tf)           # topic of each document
h = lda.components              # rank for each word for a topic
```

Figure 9: Code snippet for implementing LDA

4.2 Results

Below shows the topics modelled by LDA. We can see that the segregation of topic is working well, except for topic 3 which does not sound meaningful. As a post-process step, we remove topic 3 and for those documents originally assigned to this topic, we assign the topic of the second highest probability to the document.

```
Topic 0:
phone sound bluetooth speaker use good
Topic 1:
case phone screen like protector iphone
Topic 2:
great product phone price good just
Topic 3:
phone 34 use just like don
Topic 4:
battery charge charger charging usb power
```

We can see that Topic 0 is related to the audio aspect of the product, for example on Bluetooth speakers and sound quality. Topic 1 focuses on phone cases and screen protectors. Topic 2 comments on the general aspects of a phone, its price and whether it is good to use. Topic 4 is related to the charging aspects, eg. battery charger and power consumption of the phone.

Below shows a bar chart on the distribution of reviews across each aspect, and their average ratings.

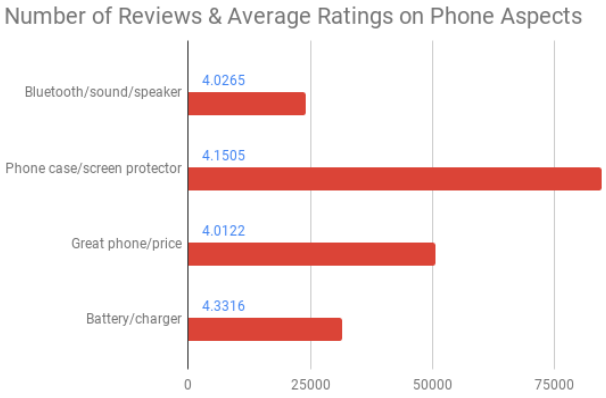

Figure 10: Number of reviews and average ratings on phone aspects

Overall, we can conclude that none of the aspects are underperforming, as the ratings are all higher than 4. Based on the chart, the most number of reviews are about phone cases and screen protectors, and reviews about Bluetooth speakers and sound are of the least amount. The best performing aspect is on battery/charger, which as an average rating of 4.3316, and the aspect with the lowest rating among all is about the general attributes of the phone, which is rated 4.0122 on average.

4.3 Evaluation

We manually annotate 20 review texts by assigning them to one of the 4 topics that the LDA model has generated. The confusion matrix is as below:

	Actual_0	Actual_1	Actual_2	Actual_4	Total
0	3	0	1	0	4
1	0	6	0	0	6
2	0	3	2	1	6
4	0	0	0	4	4
Total	3	9	3	5	

The precision, recall and F1 score are as below:

Topic	Precision	Recall	F1_score
0	0.75	1	0.857
1	1	0.66	0.8
2	0.33	0.66	0.44
4	1	0.8	0.88

From the results, we can see that all topics have an F1 score above 0.8 except Topic 2. The reason is that Topic 2 is vague by itself and any review could have contained words such as “great”, “phone” and “price”. Hence, it is not a good topic to be used.

CONCLUSION

In this report, we -

1. first showed our results for data analysis in terms of Popular Products and Frequent Reviewer, Sentence Segmentation, Tokenization and Stemming and POS Tagging,
2. demonstrated our implementation of our noun phrase summarizer and discussed the results of the noun phrases,
3. discussed on our heuristic-based approach on implementing a sentiment detector, and
4. experimented on a possible application approach using topic modelling for aspect extraction.

REFERENCES

[1] Python NLTK, https://www.nltk.org/.
[2] TextBlob: Simplified Text Processing, https://textblob.readthedocs.io/en/dev/.
[3] Jocelyn D'Souza. 2018. Learning POS Tagging & Chunking in NLP. https://medium.com/greyatom/learning-pos-tagging-chunking-in-nlp-85f7f811a8cb
[4] Susan Li. 2018. Topic Modeling and Latent Dirichlet Allocation (LDA) in Python. https://towardsdatascience.com/topic-modeling-and-latent-dirichlet-allocation-in-python-9bf156893c24

APPENDIX

1. 5 Random Samples for Sentence Segmentation

ive bought 200 of these for resale at venezuela, let me start off by saying the seller is honest, the product is good and the packaging is really good and better than i expected, of the 200 hundred only one came doa, thats a 0.5% of failure rate which i'm sure the seller would replace if i asked him to, but its cheaper for me to just trow it away than it is to ship it back and get it replaced,it can charge an iphone 5 around 120- 130% on a full charge and an htc one (m7) around 100%, the led light is superb.would recommend to anybody with a smartphone.

"[""ive bought 200 of these for resale at venezuela, let me start off by saying the seller is honest, the product is good and the packaging is really good and better than i expected, of the 200 hundred only one came doa, thats a 0.5% of failure rate which i'm sure the seller would replace if i asked him to, but its cheaper for me to just trow it away than it is to ship it back and get it replaced,it can charge an iphone 5 around 120- 130% on a full charge and an htc one (m7) around 100%, the led light is superb.would recommend to anybody with a smartphone.""]"

Sentence count: 1

bought this for my husband since the last one he had fell off repeatedly almost losing the phone altogether. this one has a very stiff clip on the back, which is what we wanted to keep it from falling off his belt. and, it has 2 leather loops on either side of the center clip so you can slide the belt through all 3. it's not coming off, no way. the loops however are not really wide. will accommodate more of a slender width belt. my husband wears work jeans everyday with a wider belt, so he only uses the clip. so far it works fine. the flap is magnetized and is pretty secure. he likes it.

"['bought this for my husband since the last one he had fell off repeatedly almost losing the phone altogether.', 'this one has a very stiff clip on the back, which is what we wanted to keep it from falling off his belt.', ""and, it has 2 leather loops on either side of the center clip so you can slide the belt through all 3. it's not coming off, no way."", 'the loops however are not really wide.', 'will accommodate more of a slender width belt.', 'my husband wears work jeans everyday with a wider belt, so he only uses the clip.', 'so far it works fine.', 'the flap is magnetized and is pretty secure.', 'he likes it.']"

Sentence count: 9

hey ordered these for my friends and they love them. the only downside is when they take it off there phone, sometimes they forget to put it back on...oops

['hey ordered these for my friends and they love them.', 'the only downside is when they take it off there phone, sometimes they forget to put it back on...oops']

Sentence count: 2

i upgraded to this phone from the samsung galaxy victory. it's definitely is an upgrade. its a lot faster with downloading apps and going on the internet. it has a very nice speaker that is clear and can be turned up pretty loudly for a phone which is nice for listening to music or speaker phone calls. its size, weight, and how thin it is, is very nice as well. it also comes with a nice headset. i would definitely recommend this phone.

"['i upgraded to this phone from the samsung galaxy victory.', ""it's definitely is an upgrade."", 'its a lot faster with downloading apps and going on the internet.', 'it has a very nice speaker that is clear and can be turned up pretty loudly for a phone which is nice for listening to music or speaker phone calls.', 'its size, weight, and how thin it is, is very nice as well.', 'it also comes with a nice headset.', 'i would definitely recommend this phone.']"

Sentence count: 7

pretty cool device and inexpensive, i'm using it with a car that has bluetooth, but its voice only. i have the nokia lumia 810 windows phone, so now i can connect to my car to make and receive calls, and simultaneously connect to this device to play music. it will turn off once not connected for a while.

"[""pretty cool device and inexpensive, i'm using it with a car that has bluetooth, but its voice only."", 'i have the nokia lumia 810 windows phone, so now i can connect to my car to make and receive calls, and simultaneously connect to this device to play music.', 'it will turn off once not connected for a while.']"

Sentence count: 3

2. 5 Random Samples for POS Tagging (wrong tags are highlighted)

[('broke', 'NN'), ('as', 'RB'), ('soon', 'RB'), ('as', 'IN'), ('i', 'NN'), ('put', 'VBD'), ('my', 'PRP$'), ('phone', 'NN'), ('in', 'IN'), ('it', 'PRP'), ('very', 'RB'), ('disappointed', 'JJ'), ('but', 'CC'), ('it', 'PRP'), ("'s", 'VBZ'), ('hello', 'JJ'), ('kitty', 'FW'), ('so', 'IN'), ('i', 'NN'), ('would', 'MD'), ('still', 'RB'), ('recommend', 'VB'), ('it', 'PRP'), ('to', 'TO'), ('people', 'NNS'), ('the', 'DT'), ('price', 'NN'), ('is', 'VBZ'), ('amazing', 'VBG')]

[('four', 'CD'), ('and', 'CC'), ('an', 'DT'), ('half', 'JJ'), ('successful', 'JJ'), ('stars', 'VBZ'), ('the', 'DT'), ('5staramerica', 'CD'), ('3100', 'CD'), ('mah', 'NN'), ('portable', 'JJ'), ('charger', 'NN'), ('is', 'VBZ'), ('a', 'DT'), ('versatile', 'JJ'), ('all-around', 'JJ'), ('34', 'CD'), ('quick', 'JJ'), ('34', 'CD'), ('charger', 'NN'), ('for', 'IN'), ('electronic', 'JJ'), ('portable', 'JJ'), ('gadgets', 'NNS'), ('of', 'IN'), ('all', 'DT'), ('sorts', 'NNS'), ('cell', 'VBP'), ('phones', 'NNS'), ('reading', 'VBG'), ('tablets', 'NNS'), ('ipods', 'NNS'), ('etc', 'VBP'), ('ipads', 'NNS'), ('and', 'CC'), ('the', 'DT'), ('like', 'JJ'), ('are', 'VBP'), ('not', 'RB'), ('on', 'IN'), ('their', 'PRP$'), ('list', 'NN'), ('of', 'IN'), ('chargeable', 'JJ'), ('items', 'NNS'), ('but', 'CC'), ('it', 'PRP'), ('charges', 'VBZ'), ('them', 'PRP'), ('too', 'RB'), ('at', 'IN'), ('a', 'DT'), ('slower', 'JJR'), ('rate', 'NN'), ('i', 'NN'), ('received', 'VBD'), ('one', 'CD'), ('of', 'IN'), ('these', 'DT'), ('items', 'NNS'), ('for', 'IN'), ('testing', 'VBG'), ('and', 'CC'), ('it', 'PRP'), ('is', 'VBZ'), ('a', 'DT'), ('sleek', 'JJ'), ('attractive-looking', 'NN'), ('charger', 'NN'), ('in', 'IN'), ('a', 'DT'), ('black', 'JJ'), ('and', 'CC'), ('purple', 'JJ'), ('cover', 'NN'), ('with', 'IN'), ('the', 'DT'), ('length', 'NN'), ('and', 'CC'), ('width', 'JJ'), ('dimensions', 'NNS'), ('of', 'IN'), ('a', 'DT'), ('cell', 'NN'), ('phone', 'NN'), ('but', 'CC'), ('clocking', 'VBG'), ('in', 'IN'), ('at', 'IN'), ('3', 'CD'), ('oz', 'NNS'), ('compared', 'VBN'), ('to', 'TO'), ('an', 'DT'), ('iphone', 'NN'), ('of', 'IN'), ('4.0', 'CD'), ('oz.there', 'EX'), ('is', 'VBZ'), ('no', 'DT'), ('on/off', 'NN'), ('switch', 'NN'), ('there', 'EX'), ('is', 'VBZ'), ('a', 'DT'), ('5v/1.5a', 'CD'), ('advertised', 'JJ'), ('input', 'NN'), ('port', 'NN'), ('to', 'TO'), ('charge', 'VB'), ('the', 'DT'), ('unit', 'NN'), ('with', 'IN'), ('a', 'DT'), ('matching', 'JJ'), ('purple-color', 'JJ'), ('mini-usb', 'JJ'), ('cable', 'NN'), ('provided', 'VBN'), ('for', 'IN'), ('input', 'NN'), ('charging', 'NN'), ('and', 'CC'), ('it', 'PRP'), ('has', 'VBZ'), ('three', 'CD'), ('lights', 'NNS'), ('which', 'WDT'), ('indicate', 'VBP'), ('the', 'DT'), ('level', 'NN'), ('of', 'IN'), ('charging', 'VBG'), ('the', 'DT'), ('output', 'NN'), ('port', 'NN'), ('is', 'VBZ'), ('rated', 'VBN'), ('by', 'IN'), ('5staramerica', 'CD'), ('as', 'IN'), ('5v/2.1', 'CD'), ('amp', 'IN'), ('no', 'DT'), ('output', 'NN'), ('cable', 'NN'), ('is', 'VBZ'), ('provided', 'VBN'), ('i', 'JJ'), ('began', 'VBD'), ('charging', 'VBG'), ('a', 'DT'), ('power-depleted', 'JJ'), ('iphone', 'NN'), ('5', 'CD'), ('and', 'CC'), ('it', 'PRP'), ("'awakened", 'VBD'), ('within', 'IN'), ('5', 'CD'), ('minutes', 'NNS'), ('charged', 'VBN'), ('at', 'IN'), ('up', 'IN'), ('to', 'TO'), ('1', 'CD'), ('per', 'IN'), ('minute', 'NN'), ('at', 'IN'), ('times', 'NNS'), ('fully', 'RB'), ('charging', 'VBG'), ('the', 'DT'), ('phone', 'NN'), ('in', 'IN'), ('just', 'RB'), ('under', 'IN'), ('2', 'CD'), ('hours', 'NNS'), ('with', 'IN'), ('2', 'CD'), ('lights', 'NNS'), ('still', 'RB'), ('showing', 'VBG'), ('i', 'JJ'), ('re-charged', 'VBD'), ('the', 'DT'), ('5staramerica', 'CD'), ('unit', 'NN'), ('and', 'CC'), ('charged', 'VBD'), ('another', 'DT'), ('iphone', 'NN'), ('with', 'IN'), ('the', 'DT'), ('same', 'JJ'), ('results', 'NNS'), ('and', 'CC'), ('with', 'IN'), ('the', 'DT'), ('remainder', 'NN'), ('of', 'IN'), ('that', 'DT'), ('power', 'NN'), ('i', 'NN'), ('was', 'VBD'), ('able', 'JJ'), ('to', 'TO'), ('charge', 'VB'), ('an', 'DT'), ('iphone', 'NN'), ('4', 'CD'), ('which', 'WDT'), ("'awakened", 'VBD'), ('in', 'IN'), ('7', 'CD'), ('minutes', 'NNS'), ('up', 'RB'), ('to', 'TO'), ('63', 'CD'), ('per', 'IN'), ('cent', 'NN'), ('capacity', 'NN'), ('before', 'IN'), ('the', 'DT'), ('3100', 'CD'), ('mah', 'NN'), ('charger', 'NN'), ('was', 'VBD'), ('depleted', 'VBN'), ('the', 'DT'), ('5staramerica', 'CD'), ('unit', 'NN'), ('itself', 'PRP'), ('re-charges', 'JJ'), ('best', 'RBS'), ('using', 'VBG'), ('a', 'DT'), ('wall', 'NN'), ('plug', 'NN'), ('not', 'RB'), ('provided', 'VBN'), ('over', 'IN'), ('the', 'DT'), ('usp', 'JJ'), ('plug', 'NN'), ('in', 'IN'), ('7', 'CD'), ('hours', 'NNS'), ('in', 'IN'), ('addition', 'NN'), ('i', 'NN'), ('was', 'VBD'), ('able', 'JJ'), ('to', 'TO'), ("'charge", 'VB'), ('the', 'DT'), ('charger', 'NN'), ('while', 'IN'), ('charging', 'VBG'), ('another', 'DT'), ('item', 'NN'), ('34', 'CD'), ('pass', 'NN'), ('thru', 'NN'), ('34', 'CD'), ('charging', 'NN'), ('but', 'CC'), ('at', 'IN'), ('a', 'DT'), ('slower', 'JJR'), ('rate', 'NN'), ('than', 'IN'), ('when', 'WRB'), ('only', 'RB'), ('charging', 'VBG'), ('using', 'VBG'), ('a', 'DT'), ('fully', 'RB'), ('charged', 'VBN'), ('5staramerica', 'CD'), ('unit', 'NN'), ('this', 'DT'), ('is', 'VBZ'), ('an', 'DT'), ('excellent', 'JJ'), ('quick-charge', 'NN'), ('unit', 'NN'), ('that', 'IN'), ('when', 'WRB'), ('fully', 'RB'), ('charged', 'VBN'), ('can', 'MD'), ('provide', 'VB'), ('a', 'DT'), ('full', 'JJ'), ('iphone', 'NN'), ('charge', 'NN'), ('within', 'IN'), ('2', 'CD'), ('hours', 'NNS'), ('with', 'IN'), ('enough', 'JJ'), ('power', 'NN'), ('for', 'IN'), ('partially', 'RB'), ('charging', 'VBG'), ('another', 'DT'), ('of', 'IN'), ('the', 'DT'), ('portable', 'JJ'), ('items', 'NNS'), ('listed', 'VBN'), ('in', 'IN'), ('its', 'PRP$'), ('specifications', 'NNS'), ('this', 'DT'), ('item', 'NN'), ('was', 'VBD'), ('provided', 'VBN'), ('to', 'TO'), ('me', 'PRP'), ('for', 'IN'), ('testing', 'VBG'), ('highly', 'RB'), ('recommended', 'VBN'), ('four', 'CD'), ('and', 'CC'), ('a', 'DT'), ('half', 'JJ'), ('effective', 'JJ'), ('stars', 'NNS'), ('charger', 'VBP'), ('micro', 'JJ'), ('usb', 'JJ'), ('cable', 'NN'), ('no', 'DT'), ('detailed', 'JJ'), ('quick-start', 'JJ'), ('instructions',

'NNS'), ('but', 'CC'), ('its', 'PRP$'), ('use', 'NN'), ('will', 'MD'), ('be', 'VB'), ('intuitive', 'JJ'), ('for', 'IN'), ('many', 'JJ'), ('buyers', 'NNS'), ('with', 'IN'), ('its', 'PRP$'), ('34', 'CD'), ('in', 'IN'), ('34', 'CD'), ('and', 'CC'), ('34', 'CD'), ('out', 'IN'), ('34', 'CD'), ('ports', 'NNS'), ('the', 'DT'), ('website', 'NN'), ('appears', 'VBZ'), ('to', 'TO'), ('be', 'VB'), ('under', 'IN'), ('construction', 'NN'), ('9999days', 'CD'), ('warranty', 'NN'), ('made', 'VBN'), ('in', 'IN'), ('china', 'NN')]
[('as', 'IN'), ('others', 'NNS'), ('mention', 'VBP'), ('this', 'DT'), ('battery', 'NN'), ('pack', 'NN'), ('does', 'VBZ'), ("n't", 'RB'), ('charge', 'VB'), ('in', 'IN'), ('a', 'DT'), ('matter', 'NN'), ('of', 'IN'), ('seconds', 'NNS'), ('however', 'RB'), ('in', 'IN'), ('my', 'PRP$'), ('case', 'NN'), ('it', 'PRP'), ('only', 'RB'), ('took', 'VBD'), ('thinking', 'VBG'), ('ahead', 'RB'), ('a', 'DT'), ('bit', 'NN'), ('on', 'IN'), ('a', 'DT'), ('recent', 'JJ'), ('trip', 'NN'), ('to', 'TO'), ('ny', 'VB'), ('after', 'IN'), ('making', 'VBG'), ('calls', 'NNS'), ('pulling', 'VBG'), ('up', 'RP'), ('maps', 'NNS'), ('checking', 'VBG'), ('yelp', 'NN'), ('reviews', 'NNS'), ('etc', 'VBP'), ('my', 'PRP$'), ('phone', 'NN'), ('went', 'VBD'), ('dead', 'JJ'), ('in', 'IN'), ('record', 'NN'), ('time', 'NN'), ('and', 'CC'), ('of', 'IN'), ('course', 'NN'), ('in', 'IN'), ('the', 'DT'), ('middle', 'NN'), ('of', 'IN'), ('a', 'DT'), ('call', 'NN'), ('having', 'VBG'), ('charged', 'VBN'), ('my', 'PRP$'), ('motorola', 'JJ'), ('battery', 'NN'), ('pack', 'NN'), ('the', 'DT'), ('previous', 'JJ'), ('night', 'NN'), ('i', 'NN'), ('was', 'VBD'), ('quickly', 'RB'), ('able', 'JJ'), ('to', 'TO'), ('connect', 'VB'), ('again', 'RB'), ('to', 'TO'), ('the', 'DT'), ('world', 'NN'), ('the', 'DT'), ('built', 'VBN'), ('in', 'IN'), ('cord', 'NN'), ('connected', 'VBN'), ('phone', 'NN'), ('and', 'CC'), ('battery', 'NN'), ('pack', 'NN'), ('in', 'IN'), ('just', 'RB'), ('seconds', 'VBZ'), ('the', 'DT'), ('phone', 'NN'), ('was', 'VBD'), ('a', 'DT'), ('little', 'RB'), ('more', 'RBR'), ('cumbersome', 'JJ'), ('to', 'TO'), ('use', 'VB'), ('with', 'IN'), ('the', 'DT'), ('pack', 'NN'), ('attached', 'VBD'), ('but', 'CC'), ('it', 'PRP'), ('was', 'VBD'), ('certainly', 'RB'), ('preferable', 'JJ'), ('to', 'TO'), ('no', 'DT'), ('phone', 'NN'), ('at', 'IN'), ('all', 'DT'), ('interestingly', 'RB'), ('the', 'DT'), ('pack', 'NN'), ('proved', 'VBD'), ('to', 'TO'), ('increase', 'VB'), ('the', 'DT'), ('amount', 'NN'), ('of', 'IN'), ('time', 'NN'), ('available', 'JJ'), ('by', 'IN'), ('a', 'DT'), ('multiple', 'NN'), ('of', 'IN'), ('3', 'CD'), ('compared', 'VBN'), ('to', 'TO'), ('my', 'PRP$'), ('phone', 'NN'), ("'s", 'POS'), ('stock', 'NN'), ('battery', 'NN'), ('back', 'RB'), ('at', 'IN'), ('the', 'DT'), ('hotel', 'NN'), ('i', 'NN'), ('left', 'VBD'), ('the', 'DT'), ('2', 'CD'), ('devices', 'NNS'), ('connected', 'VBN'), ('and', 'CC'), ('plugged', 'VBN'), ('in', 'IN'), ('only', 'RB'), ('the', 'DT'), ('battery', 'NN'), ('pack', 'NN'), ('to', 'TO'), ('the', 'DT'), ('wall', 'NN'), ('outlet', 'NN'), ('i', 'NN'), ('was', 'VBD'), ('able', 'JJ'), ('to', 'TO'), ('charge', 'VB'), ('both', 'DT'), ('devices', 'NNS'), ('at', 'IN'), ('once', 'RB'), ('even', 'RB'), ('while', 'IN'), ('periodically', 'RB'), ('using', 'VBG'), ('the', 'DT'), ('phone', 'NN'), ('besides', 'IN'), ('the', 'DT'), ('phone', 'NN'), ('i', 'NN'), ("'ve", 'VBP'), ('since', 'IN'), ('used', 'VBN'), ('the', 'DT'), ('pack', 'NN'), ('to', 'TO'), ('power', 'NN'), ('charge', 'NN'), ('my', 'PRP$'), ('kindle', 'JJ'), ('fire', 'NN'), ('when', 'WRB'), ('it', 'PRP'), ('went', 'VBD'), ('dead', 'JJ'), ('while', 'IN'), ('out', 'RB'), ('and', 'CC'), ('about', 'IN'), ('for', 'IN'), ('me', 'PRP'), ('the', 'DT'), ('key', 'NN'), ('has', 'VBZ'), ('been', 'VBN'), ('getting', 'VBG'), ('in', 'IN'), ('the', 'DT'), ('habit', 'NN'), ('of', 'IN'), ('charging', 'VBG'), ('up', 'RP'), ('the', 'DT'), ('pack', 'NN'), ('daily', 'RB'), ('as', 'IN'), ('i', 'NN'), ('do', 'VBP'), ('my', 'PRP$'), ('phone', 'NN'), ('with', 'IN'), ('a', 'DT'), ('not', 'RB'), ('supplied', 'VBN'), ('mini', 'NN'), ('usb', 'NN'), ('to', 'TO'), ('regular', 'JJ'), ('usb', 'JJ'), ('connection', 'NN'), ('cord', 'NN'), ('i', 'NN'), ('can', 'MD'), ('also', 'RB'), ('charge', 'VB'), ('the', 'DT'), ('pack', 'NN'), ('from', 'IN'), ('my', 'PRP$'), ('laptop', 'JJ'), ('since', 'IN'), ('this', 'DT'), ('was', 'VBD'), ('cheaper', 'JJR'), ('than', 'IN'), ('a', 'DT'), ('2nd', 'CD'), ('phone', 'NN'), ('battery', 'NN'), ('lasts', 'VBZ'), ('longer', 'JJR'), ('than', 'IN'), ('my', 'PRP$'), ('regular', 'JJ'), ('phone', 'NN'), ('battery', 'NN'), ('and', 'CC'), ('can', 'MD'), ('be', 'VB'), ('used', 'VBN'), ('for', 'IN'), ('more', 'JJR'), ('than', 'IN'), ('just', 'RB'), ('phone', 'NN'), ('battery', 'NN'), ('backup', 'NN'), ('buying', 'VBG'), ('the', 'DT'), ('pack', 'NN'), ('made', 'VBD'), ('more', 'JJR'), ('sense', 'NN')] [('i', 'RB'), ('love', 'VBP'), ('this', 'DT'), ('case', 'NN'), ('so', 'RB'), ('girly', 'RB'), ('adorable', 'JJ'), ('very', 'RB'), ('protective', 'JJ'), ('it', 'PRP'), ("'s", 'VBZ'), ('definitely', 'RB'), ('a', 'DT'), ('great', 'JJ'), ('case', 'NN'), ('i', 'NN'), ('was', 'VBD'), ('super', 'JJ'), ('pleased', 'JJ'), ('to', 'TO'), ('see', 'VB'), ('that', 'IN'), ('it', 'PRP'), ('arrived', 'VBD'), ('weeks', 'NNS'), ('ahead', 'RB'), ('of', 'IN'), ('time', 'NN'), ('i', 'JJ'), ("'m", 'VBP'), ('very', 'RB'), ('happy', 'JJ'), ('about', 'IN'), ('that', 'DT'), ('great', 'JJ'), ('product', 'NN')]
[('the', 'DT'), ('size', 'NN'), ('the', 'DT'), ('price', 'NN'), ('the', 'DT'), ('sound', 'NN'), ('...', ':'), ('this', 'DT'), ('august', 'JJ'), ('ms425', 'NN'), ('bluetooth', 'NN'), ('speaker', 'NN'), ('was', 'VBD'), ('just', 'RB'), ('what', 'WP'), ('i', 'NN'), ('was', 'VBD'), ('looking', 'VBG'), ('for', 'IN'), ('is', 'VBZ'), ('it', 'PRP'), ('a', 'DT'), ('bose', 'NN'), ('...', ':'), ('no', 'DT'), ('but', 'CC'), ('i', 'RB'), ('needed', 'VBD'), ('a', 'DT'), ('bluetooth', 'NN'), ('speaker', 'NN'), ('for', 'IN'), ('my', 'PRP$'), ('kindle', 'JJ'), ('fire', 'NN'), ('to', 'TO'), ('have', 'VB'), ('a', 'DT'), ('little', 'JJ'), ('louder', 'NN'), ('sound', 'NN'), ('for', 'IN'), ('music', 'NN'), ('and', 'CC'), ('movies', 'NNS'), ('without', 'IN'), ('breaking', 'VBG'), ('the', 'DT'), ('bank', 'NN'), ('the', 'DT'), ('sound', 'NN'), ('from', 'IN'), ('this', 'DT'), ('little', 'JJ'), ('speaker', 'NN'), ('is', 'VBZ'), ('very', 'RB'), ('good', 'JJ'), ('remember', 'NN'), ('there', 'EX'), ('is', 'VBZ'), ('a', 'DT'), ('break', 'NN'), ('in', 'IN'), ('period', 'NN'), ('the', 'DT'), ('first', 'JJ'), ('1/2', 'CD'), ('hour', 'NN'), ('of', 'IN'), ('use', 'NN'), ('to', 'TO'), ('calibrate', 'VB'), ('the', 'DT'), ('speaker', 'NN'), ('enjoy', 'NN')]

3. Stop Word List

Stop Words				
i	what	a	to	all

me	which	an	from	any
my	who	the	up	both
myself	whom	and	down	each
we	this	but	in	few
our	that	if	out	more
ours	these	or	on	most
ourselves	those	because	off	other
you	am	as	over	some
your	is	until	under	such
yours	are	while	again	no
yourself	was	of	further	nor
yourselves	were	at	then	not
he	be	by	once	only
him	been	for	here	own
his	being	with	there	same
himself	have	about	when	so
she	has	against	where	than
her	had	between	why	too
hers	having	into	how	very
herself	do	through		s
it	does	during		t
its	did	before		can
itself	doing	after		will
they		above		just
them		below		don
their				should
theirs				now
themselves				

4. Penn Treebank Tagset

Tag	Description	Example	Tag	Description	Example
CC	coordin. conjunction	*and, but, or*	SYM	symbol	*+,%, &*
CD	cardinal number	*one, two, three*	TO	"to"	*to*
DT	determiner	*a, the*	UH	interjection	*ah, oops*
EX	existential 'there'	*there*	VB	verb, base form	*eat*
FW	foreign word	*mea culpa*	VBD	verb, past tense	*ate*
IN	preposition/sub-conj	*of, in, by*	VBG	verb, gerund	*eating*
JJ	adjective	*yellow*	VBN	verb, past participle	*eaten*
JJR	adj., comparative	*bigger*	VBP	verb, non-3sg pres	*eat*
JJS	adj., superlative	*wildest*	VBZ	verb, 3sg pres	*eats*
LS	list item marker	*1, 2, One*	WDT	wh-determiner	*which, that*
MD	modal	*can, should*	WP	wh-pronoun	*what, who*
NN	noun, sing. or mass	*llama*	WP$	possessive wh-	*whose*
NNS	noun, plural	*llamas*	WRB	wh-adverb	*how, where*
NNP	proper noun, singular	*IBM*	$	dollar sign	*$*
NNPS	proper noun, plural	*Carolinas*	#	pound sign	*#*
PDT	predeterminer	*all, both*	"	left quote	' or "
POS	possessive ending	*'s*	"	right quote	' or "
PRP	personal pronoun	*I, you, he*	(	left parenthesis	[, (, {, <
PRP$	possessive pronoun	*your, one's*	)	right parenthesis	],), }, >
RB	adverb	*quickly, never*	,	comma	,
RBR	adverb, comparative	*faster*	.	sentence-final punc	. ! ?
RBS	adverb, superlative	*fastest*	:	mid-sentence punc	: ; ... – -
RP	particle	*up, off*			