CHAPTER 4

DATA COLLECTION AND PROCESSING

4.1 Data Collection

The data were taken from amazon customer's review websites, that is specifically talking about the after-market experience of customer's usage on Samsung S9 smartphone. The data is in form of textual data that is been posted since August 2018 up until now. The web page of amazon customer's review can be seen in Figure 4.1. The comparison dataset that talking about other smartphone brands' customer reviews is downloaded from Kaggle.com in form of CSV sheet dataset.

Figure 4. 1. Amazon Customer Review Website

This research applies the method of scrapping via the language of XPATH in python. XPATH is an expression language that accesses and processes items in XML (Extensible Mark-up Language) by addressing the syntax to access the hierarchy of the preferred accessed system. In this research the scrapping process is conducted using python and the web is accessed using opera browser. The detail of the scrapping process can be seen below.

4.1.1 Defining the Website Structure

Since XPATH method is applied to scrap the amazon website, the website structure can be inspected via developer tools in opera. The customer reviews in website contain some division of classes, such as review author, review tittle, review ratings, date of review, review text and then review helpful. XPATH method tries to analyse the hierarchy of the system by defining the name of the classes in python, which is shown in Figure 4.2.

```
XPATH_REVIEWS = '//div[@data-hook="review"]'
XPATH_REVIEW_RATING = './/i[@data-hook="review-star-rating"]//text()'
XPATH_REVIEW_HEADER = './/a[@data-hook="review-title"]//span[@class]//text()'
XPATH_REVIEW_AUTHOR = './/div[@class="a-profile-content"]//text()'
XPATH_REVIEW_DATE = './/span[@data-hook="review-date"]//text()'
XPATH_REVIEW_BODY = './/span[@data-hook="review-body"]//span[@class]//text()'
XPATH_REVIEW_HELPFUL = './/span[@data-hook="helpful-vote-statement"]//text()'
```

Figure 4. 2. Defining the web structure in python

We can see that for example the defined name for XPATH_REVIEWS variable is '//div [@data-hook=" review"]' in which it can be seen from inspect elements menu in opera for each variable of each desired classes in website that are going to be scrapped. The classes name on each sub-title for the customer reviews in this website can be seen in Figure 4.3, 4.4, 4.5, 4.6, 4.7, and 4.8 respectively.

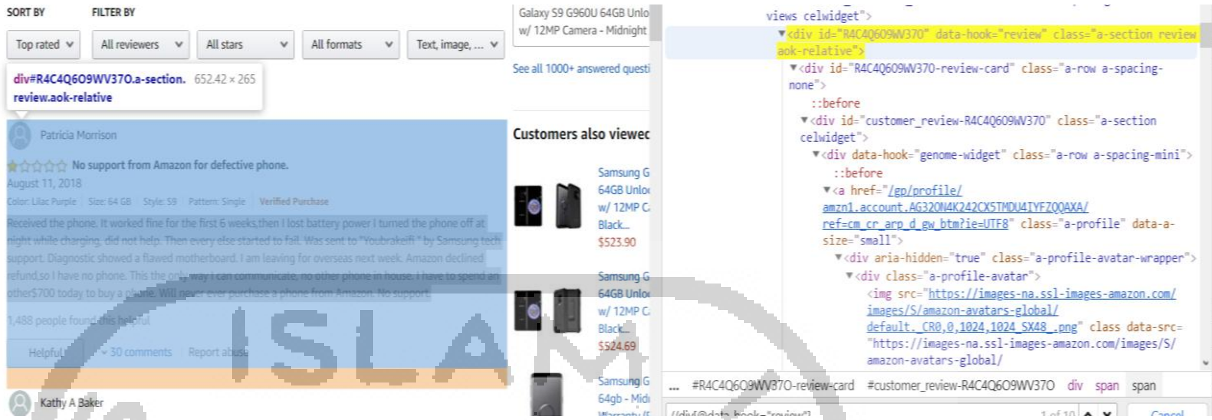

Figure 4. 3. The elements of review body in the website

Figure 4. 4. The elements of review rating in the website

Figure 4. 5. The elements of review title in the website

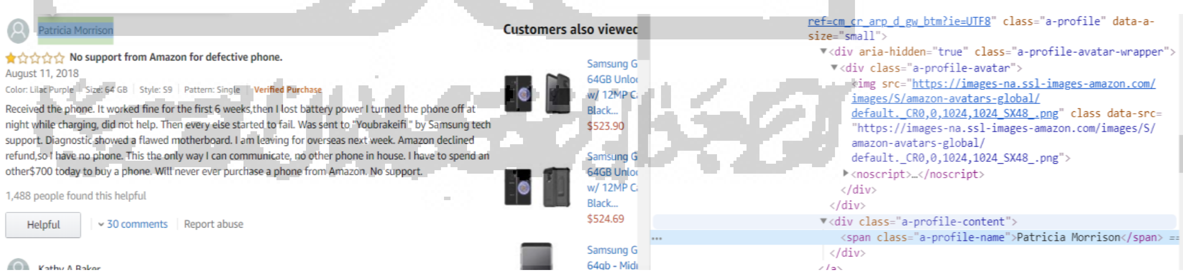

Figure 4. 6. The elements of review author in website

Figure 4. 7. The elements of review date in the website

Figure 4. 8. The elements of review helpful in the website

4.1.1 Website URL and Website' ASIN Input

In XPATH method, the website's ASIN is employed to get inside the hierarchy of the website structure specifically for amazons' websites. ASIN stands for Amazon Standard Identification Number. It's a 10-character alphanumeric unique identifier that is assigned by Amazon.com and its partners. It is used for product-identification within Amazon.com organization. ASINs are only guaranteed unique within a marketplace. So, different national Amazon sites may use different ASINs for the same product. The code to define the website source and its ASIN shown in Figure 4. and 4.10.

```
url = 'https://www.amazon.com/product-reviews/' + asin + '?pageNumber=' + str(p_num) + '&sortBy=recent' + '&ie=UTF8' + '&reviewerType=' + (review)
print(url)
```

Figure 4. 9. URL defining code

```
asins = ['B079H6RLKQ'] #s9
```

Figure 4. 10. Amazon Website's ASIN

4.1.2 The Main Process of Scrapping

After locating the elements of the website, then, the function to navigate the webpages on the current websites is defined. This website contains 2058 reviews and each web page displays 10 reviews. So, the function could be shown below.

```
        p_num += 1
        if p_num == 210 ++ 1:
            break
```

Figure 4. 11. Web page navigation

After the web page is navigated, then, the data-frame is defined to hold the scrapped values using the pandas library in python. The code itself is intact with XPATH function from html library that has been called previously, but it has not been used yet in the mapping process. It is also required to name the column inside the data-frame that is used to store the data, and the codes and function of those processes mentioned above can be seen in Figure 4.12, 4.13, 4.14, and 4.15 respectively.

```
reviews_df = pd.DataFrame()
```

Figure 4. 12. Naming the data-frame

```
raw_review_author = review.xpath(XPATH_REVIEW_AUTHOR)
raw_review_rating = review.xpath(XPATH_REVIEW_RATING)
raw_review_header = review.xpath(XPATH_REVIEW_HEADER)
raw_review_date = review.xpath(XPATH_REVIEW_DATE)
raw_review_body = review.xpath(XPATH_REVIEW_BODY)
raw_review_helpful = review.xpath(XPATH_REVIEW_HELPFUL)
```

Figure 4. 13. Defining the named elements to XPATH

```
'review_text': raw_review_body,
'review_posted_date': raw_review_date,
'review_header': raw_review_header,
'review_rating': raw_review_rating,
'review_helpful': raw_review_helpful,
'review_author': raw_review_author
```

Figure 4. 14. Naming the column in the data-frame to store the data

```
reviews_df = reviews_df.append(review_dict, ignore_index=True)
```

Figure 4. 15. Appending the named column to the data frame

After those processes above are completed, the next process is to join the data-frame function with the main process function of scrapping, that iscalled by using the named function previously in locating process of the website. The complete code can be seen below.

```
reviews_df = scrape_reviews(asins)
```

Figure 4. 16. The function of the main process

Then, the data obtained from this scrapping process on amazon.com website are saved in CSV format. The collected reviews data about Samsung S9 are totaled as 2058 reviews. And for the dataset of other smartphone brands taken from Kaggle.com is numbered up to 1.4 million reviews talking about various smartphone brands. The sample of the review data taken can be seen in table 4.1.

Review author	Review text
Abdullah S	First if all it was NOT new. Second, it came with the previous model charger. Third the internet connection did not work so well, it would keep disconnecting and reconnecting. Very disappointed
Raymond	Good phone. Upgraded from s8 and there is almost no difference. Both great phones
Amber	Came as described, be patient it takes a while to ship
J C Weaver	Bought this from Amazon. Phone reception is terrible. I have to go outside and hunt for a signal to make a call. My Wife's Galaxy S9 has no problem at all. Can use her phone anywhere to make a call. Swap sims and the problem remains the same. Samsung is less than helpful because I didn't buy it from them. Amazon has no customer service. Probably won't buy a phone from Amazon again

Table 4. 1. Reviews data sample

4.2 Data Processing

For data processing, the input of product review is constructed in the form of semi-structured textual data. The process in getting into the data starts with data pre-processing as the cleaning and normalization process of the dataset. Then, the implementation of topic modelling can be carried on after the dataset is normalized and cleaned. The comparison dataset uses downloaded reviews from Kaggle that also undergoes a pre-processing step, the dataset from Kaggle contains customer reviews about various smartphone brands besides Samsung S9 smartphone.

4.2.1 Pre-processing

Pre-processing steps become a crucial factor in this research, a proper pre-processing will give a better insight from the dataset used. The complete and brief description about data pre-processing flow that is used in this research is shown in Figure 4.17.

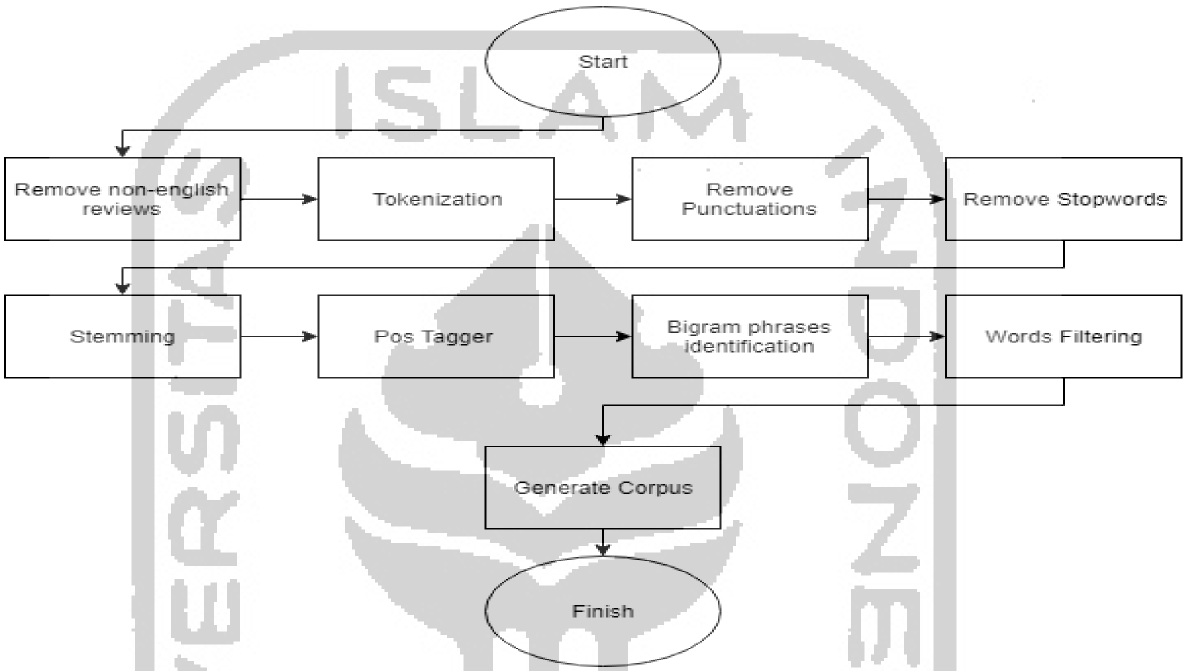

Figure 4. 17. Pre-processing flowchart

a. Remove non-English reviews

From the data obtained from both Amazon or Kaggle, it is found that there are many reviews talking about the phone in various languages. Since this research only applies reviews in English for the analysis process, it will become a noise if there is a non-English review existed in the dataset, which researcher opted those data to be removed. This process resulted several reductions in the dataset, mostly in the other smartphone dataset which is reduced from 1.4 million data to approximately 500.000 reviews. researcher uses pandas library in python to remove non-English words. The brief description of this process can be seen below.

Table 4. 2. Removing non-English reviews step

Input	Process	Output

Dataset in CSV format	• Read csv files using pandas library
	`data = pd.read_csv('s9.csv',sep=',')`
	• Removing non-English words using code below
	`dfx = [df1,df2,df3,df4,df5,df6]`
	`for d in dfx:`
	`    indexNames = d[d['lang'] != "en" ].index`
	`    d.drop(indexNames, inplace =True)s`
		Clean review dataset with non-English words

b. Tokenization

This is a process of dividing a group of words in form of sentence, into its each word state and each word called as tokens. These tokens still have no meaning in terms of insights of the data. But it is a step to make it meaningful for other process after tokenization. Inside this process, a transformation to lowercase words is undergone as well. The sample of input and the output of this process can be seen in Table 4.3.

Table 4. 3. Tokenization result samples

Inputs	Outputs
It is a good buy when I got it. Got the Mophie case right away because I know Samsung batteries	'it', 'is', 'good', 'buy', 'when', 'got', 'it', 'got', 'the', 'mophie', 'case', 'right', 'away', 'because', 'know', 'samsung', 'batteries'
This phone would not connect to service outside my calling area.	this', 'phone', 'would', 'not', 'connect', 'to', 'service', 'outside', 'my', 'calling', 'area'
Phone is great. You can tell it is brand new and not refurbished.	'phone', 'is', 'great', 'you', 'can', 'tell', 'it', 'is', 'brand', 'new', 'and', 'not', 'refurbished'
Haven't even have the phone a year in its already trash	'haven', 'even', 'have', 'the', 'phone', 'year', 'in', 'its', 'already', 'trash'

c. Remove Punctuations

This step undergoes a process of removing punctuations that might disturb the dataset. Punctuation such as, (! ()-[]{};:'"\,<>./?@#$%^&*_~).

d. Stemming

Stemming is a process of reducing a word to its words stem or we can say to the original root forms of the words. It is an important part in this pre-processing step, because such words can give a different interpretation in this research. This process uses gensim library simple pre-processing library. For these three processes above, this research uses the library from gensim, called simple pre-process library that enables these processes in one part. The brief description can be seen below.

Table 4. 4. Tokenizing, Stemming and Removing punctuation process

<table>
<tr><th>Input</th><th>Process</th><th>Output</th></tr>
<tr>
<td>Cleaned review data all in English language, already stored in data frame in pandas</td>
<td>• Implementing stemming, tokenizing and removing punctuations using the code below.
<pre><code>def sent_to_words(sentences):
    for sentence in sentences:
        yield(gensim.utils.simple_preprocess(str(sentence), deacc=True))

data_words = list(sent_to_words(data.extract))</code></pre></td>
<td>• Tokenized words,
• Stemmed words,
• No punctuations.</td>
</tr>
</table>

e. Remove Stop words

This step removes words that are considered as less-meaningful words that might disturb LDA result. This process implements the stop words set downloaded from Spacy library in python. Some less-meaningful words such as, I, this, that, it, is, etc. The sample list of stop words from Spacy can be seen in table 4.2.

Table 4. 5. Sample list of Stop Words from Spacy

Stop Words					
a',	'all',	'among',	'anything',	'became',	'being',
'about',	'almost',	'amongst',	'anyway',	'because',	'below',
'above',	'alone',	'amount',	'anywhere',	'become',	'beside',
'across',	'along',	'an',	'are',	'becomes',	'besides',
'after',	'already',	'and',	'around',	'becoming',	'between',
'afterwards',	'also',	'another',	'as',	'been',	'beyond',
'again',	'although',	'any',	'at',	'before',	'both',
'against',	'always',	'anyhow',	'back',	'beforehand',	'bottom',

Table 4. 6. Stop Words removal process

Input	Process	Output

Clean words from previous process.	• Calling Spacy library for stop words.
	`from spacy.lang.en.stop_words import STOP_WORDS`
	• Input to dataset variable.
	`def remove_stopwords(texts):`
	`    return [[word for word in simple_preprocess(str(doc)) if word not in spacy.lang.en.STOP_WORDS and len(word) > 3] for doc in texts]`
		• Cleaned review dataset

f. Part-of Speech Tagger

This process identifies the part of speech of given words to the syntax in python. It implements the use of Spacy library of Part of speech tagging. This process also crucial, because it defines the words that are used in the LDA process. In applying LDA for finding the topics about the features of the analysed reviews dataset of the phones, this research decides to use only Noun part-of speech, which indicates the features of the phone itself. Such word that is not Noun part-of speech will be removed.

Table 4. 7. POS Tagging process

Input	Process	Output

No common words review dataset.	• Loading spacy library for POS tagging
	`nlp = spacy.load('en_core_web_lg', disable=['parser', 'ner'])`
	• Sorting only Noun part of speech in data set
	`data_lemmatized = lemmatization(data_words_bigrams, allowed_postags=['NOUN'])`
		Review data only with Noun Part-of speech

g. Bigram Phrases Identification

After the data are considered to be clean enough, bigram identification process is done. This process will find the 2 words that mostly occur together in the dataset and lock them in the dictionary that later can give a more meaningful insight towards the LDA analysis. The example of implementation of this result can be seen in Table 4.3.

Table 4. 8. Bigram Samples

Bigram Samples	
wifi_calling	picture_quality
fingerprint_sensor	sim_card
battery_life	fingerprint_scanner

Table 4. 9. Bigram phrases identification step

Input	Process	Output

Clean review dataset	• Declare the function to create bigram
	`def make_bigrams(texts):`
	`    return [bigram_mod[doc] for doc in texts]`
	• The function to create the bigram phrases
		Review with some words bonded together known as bigram phrases

```
#Form Bigrams
data_words_bigrams = make_bigrams(data_words_nostops)
```

h. Words Filtering

This process filtered out words that occur less than 5 times in whole dataset and removing words that occur not more than 30 percent in the whole dictionary. This process is aimed to clean the dictionary used in LDA and avoid words that is less frequently occurs and occurs many times inside the dictionary and gives a maximum result to the LDA. This filtering process might reduce the dataset in numbers.

.

Table 4. 10. Words filtering step

Input	Process	Output

Review dataset equipped with bigram phrases inside the dataset	• Creates the dictionary to use the filter function using gensim corpora library
	`# Create Dictionary`
	`id2word = corpora.Dictionary(data_lemmatized)`
	• Filtering the words, removing the words that are occurring less than 5 and not more than 30% of the whole dataset
	`id2word.filter_extremes(no_below=5, no_above= 0.3)`
		Clean datasets

i. Generate Corpus

This process generates a corpus, or we can say as bag-of words. It is a form of words in vectors and each word is identified as unique tokens that can be analysed by LDA later. Which it enables LDA to identify each word in different meaning. These unique tokens are the input of creating this corpus, that is derived from the cleaned data reviews in the previous process mentioned above. It is also calculating the frequency of unique tokens that later can be accumulated by LDA process in topic modelling.

Table 4. 11. Generating corpus step

<table>
<tr><th>Input</th><th>Process</th><th>Output</th></tr>
<tr><td>Filtered and clean dataset</td><td>• Creating corpus using genism doc2bow library<pre># Term Document Frequency
corpus = [id2word.doc2bow(text) for text in texts]</pre></td><td>Corpus for LDA</td></tr>
</table>

4.2.2 Latent Dirichlet Allocation

This process uses gensim python library for topic modelling, which is LDA library, that contains module to run topic modelling in a given datasets. The result of LDA will show a given list of topics and possible representative words that correlate to the topics and the percentage of words representation to each topic. In this research, LDA process is held 2 times for a different dataset. The first dataset is Samsung S9 reviews from amazon.com and the second dataset is other smartphone brand reviews taken from Kaggle.com.

Table 4. 12. Samsung S9 reviews LDA process

<table>
<tr><th>Input</th><th>Process</th></tr>
<tr><td>• Filtered Words in dictionary
• Corpus</td><td>• Using gensim LDA library<pre># Build LDA model
lda_model = gensim.models.ldamodel.LdaModel(corpus=corpus,
                                            id2word=id2word,
                                            num_topics=8,
                                            update_every=None,
                                            passes=1000,
                                            alpha=0.5,
                                            eta=0.5,
                                            per_word_topics=True)</pre></td></tr>
</table>

Then, the result of this LDA model implementation is called by using pprint library. And the result that is displayed in the program can be seen in Figure 4.18. However, there are many scrambled words inside each topic generated, which then can be analysed the most probable result that we can interpret.

```
[(0,
  '0.076*"device" + 0.050*"samsung" + 0.039*"update" + 0.023*"smartphone" + '
  '0.022*"software" + 0.021*"note" + 0.020*"bixby" + 0.019*"datum" + '
  '0.016*"transfer" + 0.016*"version"'),
 (1,
  '0.121*"work" + 0.094*"verizon" + 0.086*"card" + 0.048*"carrier" + '
  '0.043*"review" + 0.041*"sprint" + 0.037*"network" + 0.037*"store" + '
  '0.027*"model" + 0.018*"factory"'),
 (2,
  '0.069*"battery_life" + 0.063*"iphone" + 0.055*"app" + 0.051*"thing" + '
  '0.041*"picture" + 0.031*"year" + 0.023*"user" + 0.021*"video" + '
  '0.019*"star" + 0.016*"picture_quality"'),
 (3,
  '0.105*"screen" + 0.096*"camera" + 0.074*"galaxy" + 0.063*"battery" + '
  '0.061*"quality" + 0.032*"case" + 0.028*"edge" + 0.026*"samsung" + '
  '0.024*"upgrade" + 0.022*"speaker"'),
 (4,
  '0.274*"love" + 0.070*"deal" + 0.042*"color" + 0.026*"change" + 0.021*"need" '
  '+ 0.017*"scratch" + 0.015*"photo" + 0.014*"gift" + 0.012*"pic" + '
  '0.012*"learning"'),
 (5,
  '0.112*"amazon" + 0.079*"time" + 0.078*"samsung" + 0.075*"problem" + '
  '0.070*"month" + 0.041*"warranty" + 0.029*"week" + 0.023*"return" + '
  '0.021*"charge" + 0.020*"hour"'),
 (6,
  '0.065*"brand" + 0.061*"purchase" + 0.055*"seller" + 0.043*"charger" + '
  '0.038*"money" + 0.031*"day" + 0.030*"condition" + 0.027*"accessory" + '
  '0.025*"item" + 0.023*"order"'),
 (7,
  '0.120*"issue" + 0.095*"price" + 0.095*"feature" + 0.095*"product" + '
  '0.045*"service" + 0.031*"cell" + 0.029*"provider" + 0.022*"wifi_calling" + '
  '0.020*"contact" + 0.020*"call"')]
```

Figure 4. 18. Samsung S9 reviews LDA result

This process also done for the Kansei words comparison dataset, to find the topics about the reviewed features from various smartphone brands. The brief explanation about this step can be seen as follows:

Table 4.13. Other smartphone brand LDA process

Input	Process

• Filtered Words in dictionary
• Corpus
	• Using gensim LDA library
	`# Build LDA model`
	`lda_model = gensim.models.ldamodel.LdaModel(corpus=corpus,`
	`id2word=id2word,`
	`num_topics=10,`
	`update_every=1,`
	`passes=10,`
	`chunksize=500,`
	`alpha='auto',`
	`eta='auto',`
	`per_word_topics=True)`

The next process is the same with the previous dataset conducted previously, which is calling the result with pprint library in python. However, the result also shows some scrambled words inside the result, and researcher must manually assess the LDA result to find the most suitable insight from the collections of words into the topics which later considered as the features of various phone from the comparison dataset. The result of this process can be seen in Figure 4.19.

```
[(0,
  '0.120*"time" + 0.094*"month" + 0.076*"problem" + 0.066*"work" + '
  '0.054*"reception" + 0.046*"text" + 0.035*"service" + 0.032*"verizon" + '
  '0.025*"message" + 0.024*"card"'),
 (1,
  '0.189*"battery" + 0.096*"life" + 0.088*"motorola" + 0.066*"button" + '
  '0.050*"mobile" + 0.044*"menu" + 0.036*"voice" + 0.036*"charge" + '
  '0.027*"hour" + 0.023*"blackberry"'),
 (2,
  '0.065*"speaker" + 0.054*"look" + 0.050*"product" + 0.049*"player" + '
  '0.048*"music" + 0.037*"flip" + 0.032*"volume" + 0.031*"speakerphone" + '
  '0.029*"store" + 0.029*"company"'),
 (3,
  '0.107*"pro" + 0.083*"call" + 0.071*"con" + 0.054*"money" + 0.047*"number" + '
  '0.033*"fact" + 0.032*"replacement" + 0.032*"key" + 0.027*"piece" + '
  '0.024*"today"'),
 (4,
  '0.072*"day" + 0.058*"contract" + 0.055*"thing" + 0.054*"network" + '
  '0.045*"colour" + 0.040*"software" + 0.035*"handset" + 0.033*"charger" + '
  '0.029*"case" + 0.027*"version"'),
 (5,
  '0.164*"feature" + 0.113*"love" + 0.088*"week" + 0.080*"people" + '
  '0.070*"design" + 0.061*"ringtone" + 0.040*"friend" + 0.037*"sound" + '
  '0.025*"area" + 0.021*"style"'),
 (6,
  '0.103*"review" + 0.048*"didn" + 0.048*"couple" + 0.041*"need" + '
  '0.041*"keypad" + 0.040*"razr" + 0.038*"upgrade" + 0.028*"brand" + '
  '0.025*"experience" + 0.024*"system"'),
 (7,
  '0.077*"cell" + 0.062*"model" + 0.061*"color" + 0.060*"display" + '
  '0.058*"game" + 0.051*"function" + 0.046*"signal" + 0.041*"keyboard" + '
  '0.033*"minute" + 0.024*"choice"'),
 (8,
  '0.172*"camera" + 0.125*"quality" + 0.104*"year" + 0.061*"picture" + '
  '0.042*"video" + 0.041*"internet" + 0.034*"user" + 0.029*"reason" + '
  '0.023*"ring_tone" + 0.020*"world"'),
 (9,
  '0.209*"screen" + 0.075*"price" + 0.070*"size" + 0.054*"bluetooth" + '
  '0.044*"hand" + 0.038*"pocket" + 0.035*"device" + 0.027*"talk" + '
  '0.018*"fone" + 0.017*"line"')]
```

Figure 4. 19. Other smartphone dataset LDA result

4.2.3 Topic Interpretation

This is the process of interpreting the LDA process result, in which this research will filter the most meaningful representative words on each topic in the LDA process. Somehow, it will possible that in certain topics will contain a meaningless word, so this research opted to bypass those meaningless words. Later, this LDA process is aimed to find the topics talking about the features of the smartphone that is this object of this research.

a. Samsung S9 review dataset

As we can see in Figure 4.18, 8 topics could be generated for the dataset and the results show some topics about the features popping up with various words that can be neglected. Then, we can interpret it one by one starting from topic 0 which is topic 1. However, researcher decides to look-up directly to the features of the phone, which are:

- Bixby Features
- Battery life
- Picture quality
- Screen
- Camera
- Speaker
- Colour of the phone
- Charger
- Accessory
- Price

As we know that there are some features talking about amazon warranty issue, that researcher decides not to choose it as the reviewed features of the phone, because it talks about the warranty of the amazon service instead of the phone. The carrier card network is mostly talking about the carrier availability of the carrier card towards the phone that is sold to customer from the seller, which is correlated to the seller quality not the phone and Wi-fi calling feature that is correlated to the carrier features such as Sprint, Verizon, etc that researcher found a lot on the dataset.

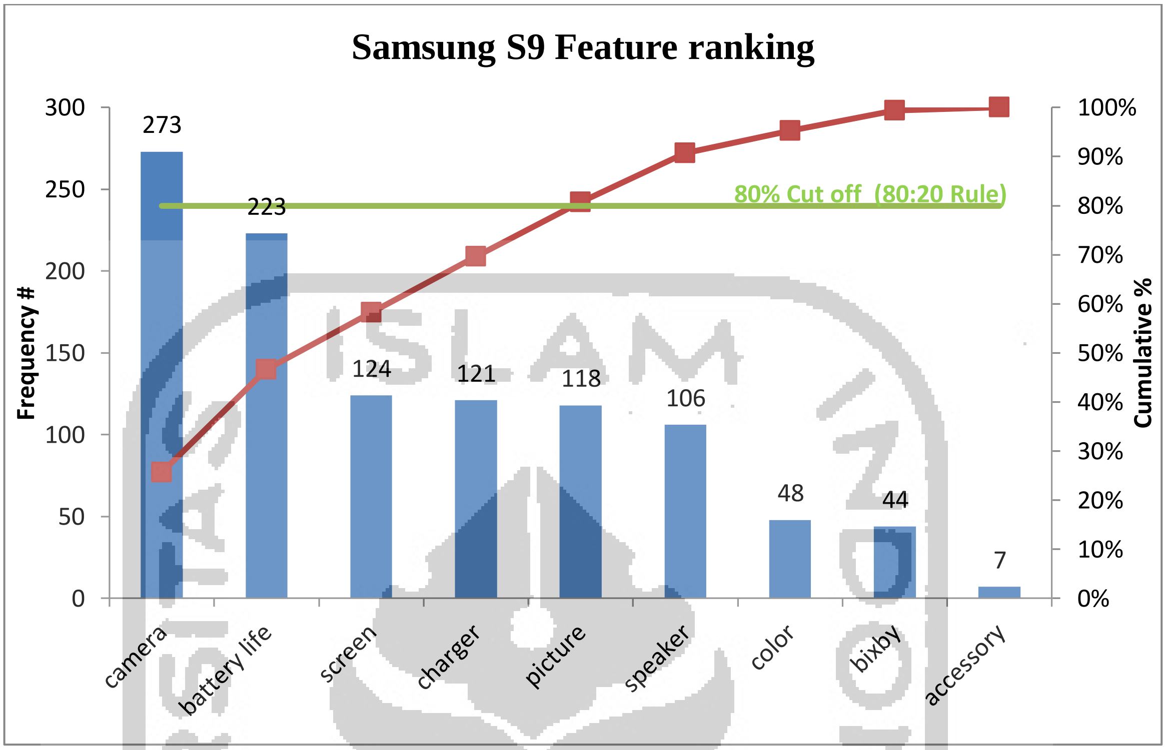

Figure 4. 20. Samsung S9 feature ranking chart

By ranking the feature of these reviews, it is sorted by the number of the positive reviews attached to the features. Hence, it can be seen the rank in order; camera as the 1st ranked feature, secondly battery life, 3rd is the screen feature, charger feature, the 4th is the picture quality feature, 5th is the speaker feature, 6th is the colour of the phone feature, 7th is Bixby feature and the last ranked feature is the accessory feature. However, these features are not readily to be implemented in the guidelines, further analysis on the kansei words occurrences toward these features is conducted to support the decision on what features to be used in the improvement guidelines.

b. Other Smartphone brands reviews dataset

As we can see in Figure 4.19, the same case happens as the previous process, there are some words which a bit like a noise, that researcher chooses directly to look-up on what the features on each topic that can be understood by researcher. Some of the features of the phone that can be understood are:

- Battery Life
- Speaker
- Network
- Charger
- Software
- Colour
- Keypad
- Display
- Camera Quality
- Bluetooth
- Price

In this case, there are some features that assumed as uncorrelated with the phone feature, but with the other parties that hold the transactions of the phone like the reception problems from the seller and also the Verizon card availability on the phone. Those are related to the card instead of the phone. Besides, there are other features that decided to be excluded from analysis, which are ringtone, signal, and screen due to their similarity towards the other feature that has been analysed previously like network, speaker, and display because for the result of the analysis almost identical.

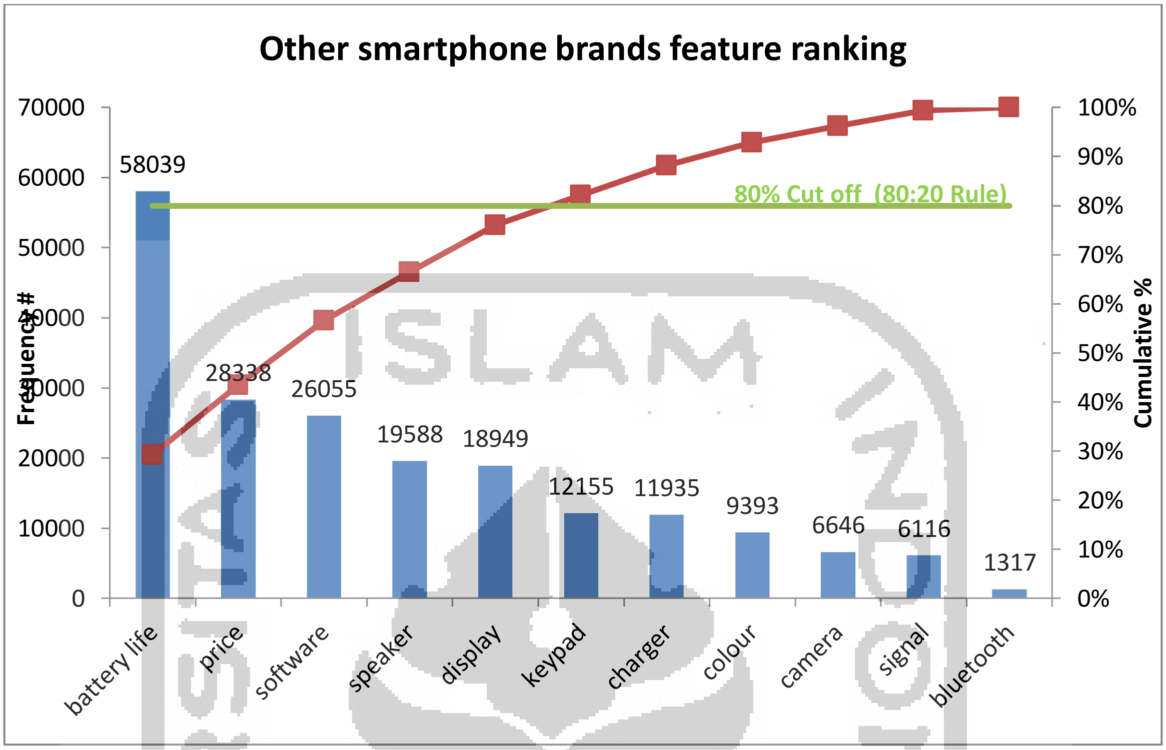

Figure 4. 21. Other smartphone brand features' ranking chart

As for this other smartphone brand features, it is analysed by the occurrences of the positive reviews in whole dataset, we can see that the ranking of the features in order are battery life as the best feature and Bluetooth feature as the last ranked features on these dataset reviews. However, these pre-liminary conclusion still needs a supporting data towards the decision on the improvement guidelines by finding the kansei words occurred to these features in the next chapter