ISSN No:-2456-2165

Sentiment Analysis and Opinion Summarization in Product Reviews Using Random Forest Algorithm

Asep Aprianto
Faculty of Informatics Engineering
Telkom Univeristy
Bandug, Indonesia

Abstract:- Product review is one of the criteria that is useful for prospective buyers to make decisions in purchasing a product. The large number of product reviews makes it difficult to make conclusions on the contents of product reviews so that consumers have difficulty in deciding to buy a product. To overcome this problem, we need a system that can automatically identify product features in product reviews. There are two steps before entering the summary generation: the first step is the extraction of product features which is carried out using the association mining method to get frequent itemsets with two word selection schemes, namely noun filtering and noun phrase filtering. The second step is the classification of extracted product features using a supervised learning approach with the Random Forest algorithm. Summarization of product reviews on each feature is carried out extractively by displaying product features with an orientation to separate positive and negative reviews.

The use of association mining method with two word selection schemes can produce an f-score of around 20% -40%, in accordance with the specified minimum support. This can occur because many product features are extracted but are not the same as the expert judgment product features, and there are also many errors in the labeling of expert judgment that affects the value of the evaluation calculation. In the classification process, the use of several classification attributes affects the resulting accuracy value.

Keyword:- *Association Mining, Classification, Opinion Summarization, Product Feature Extraction, Product Review, Random Forest.*

I. INTRODUCTION

Product reviews are very useful in helping consumers decide on the purchase of a product and in assisting producers in seeing consumer responses to their products. Consumers and producers who read a set of product reviews want to know whether the product has a positive or negative opinion. Pang and Lee [2008] pointed out the importance of customer reviews for a product and revealed that around 73% - 83% of readers of online reviews for restaurant, hotel and tourist service products are influenced by customer reviews in deciding whether or not to purchase these products [1].

Freedom to give a review resulted in a large number of reviews and many of the reviews are not written in the correct grammar. Consumers as readers of reviews often find it difficult to understand the reviews and ultimately cannot draw conclusions from existing product reviews. Therefore, a summary of opinions on product features is expected to help consumers understand and improve the accuracy of drawing conclusions from product reviews.

Sentiment analysis and product review summarization are carried out at the aspect level, which is by extracting the product features available at the review. Sentiment analysis and review summarization are carried out in three stages: extracting product features, classifying product features, and summarizing the reviews in an extractive way. About 80% of product features are nouns, so the selection process of nouns and noun phrases is used to get product candidate features. To ensure that the noun is a candidate feature, an association mining process is conducted for frequent itemset searches with minimum support threshold.

The product feature extraction process has not yet fully identified the product features because in product reviews, many consumers do not write in good and correct grammar. For this reason, supervised learning is needed to help orient opinions on extracted product features [2]. Supervised learning is done using the Random Forest Classifier. The initial step is to build a model of training data. The model is then tested on test data to train the identified product features along with the class they are aiming for. The output generated from this classification process will be a summary of product features.

II. EASE OF USE

A. Sentiment Analysis at Aspect and Entity Levels

Sentiment analysis at the aspect and entity level shows a good performance between sentiment analysis at the document and sentence level. This aspect level was originally referred to as a feature level (feature based opinion mining and summarization) [3]. Sentiment analysis at the aspect and entity level is smaller than at the document and sentence level. The purpose of this aspect level is to find sentiment on the entity and on its different aspects. Different from the sentence level which only sees whether the sentence is oriented on positive or negative sentiment, the aspect level looks at something that is highlighted in the sentence. So it could be that in one sentence there are 2 or more aspects or entities in the spotlight. If so, then it's the aspect or entity whose sentiment orientation must be seen.

B. Lemmatization

Stemming is a process that aims to reduce the amount of variance in the representation of a word [4]. But there is a risk when we use stemming, which is the loss of information from the words in the stem. This is likely to reduce the accuracy or precision produced. Meanwhile, the benefit of using this stem is, the stemming process can increase the ability to increase recall. In general, stemming algorithm is transforming a word into a standard form of morphological representation. An example of the stemming process is the words "computable, computability, computation, computational, computed, computing" will change to "compute" as "compute" is the basic word for these words.

Meanwhile, lemmatization is a process to find the basic form of a word [5]. The lemmatization process aims to normalize text or words based on the basic form which is the form of the lemma. Normalization here is the process of identifying and removing prefixes and suffixes from a word. Lemma is the basic form of a word that has a certain meaning based on the dictionary. An example of the lemma process is, for example, the sentence "The boy's cars are different colors" will change to the phrase "The boy car be differ color" due to the transformation of some of the words in the sentence.

The results of this lemmatization process are generally better than the stemming process. Lemmatization is processed better because it does not eliminate the meaning of the word itself, whereas stemming directly changes the word to its standard morphological form, so the meaning of the word may disappear.

C. Stop Words Removal

Stop words are common words that usually appear in a sentence in a very large number and the word is considered to have no meaning. A very large number here means that the appearance of this general word is very often but the word is considered to have no meaning or cannot be used as a feature of a sentence or text. 80% of words from the existing documents are words that are not useful for the extraction process [6].

Stop words removal means removing words that are considered to have no meaning in a sentence so the process of stop words removal in text mining is very useful to reduce noise in a sentence. The examples of stop words in English are "of", "the", "is", "i", "am" etc.

D. Association Mining

Association mining is a data mining technique to find interesting relationships in large data sets. Data for input in association mining is in the form of a set of transaction data consisting of itemset for each transaction. One of the common activities carried out using association mining is finding relationships between items purchased by customers through a series of purchase transactions. This process is carried out through two main steps, namely [7]:

- *Frequent Itemset Generation*

This step aims to find an itemset that meets the minimum support threshold. The itemset which then passes the predetermined threshold is called a frequent itemset. The frequent itemset combinations that are generated differ depending on the data they process. Algorithm generation of frequent itemset using apriori algorithm can be seen in the following figure.

1. k=1
2. Fk= { i|i € 1^ α({i}) >= N x minsup} % searching for all 1-itemset is frequent.
3. repeat steps 4-10, until Fk=Ø
4. k=k+1
5. Ck=*apriori*_gen(Fk−1) % awaken itemset candidates
6. for each transaction t € T, take steps 7-9.
7. Ct=subset (Ck,t) %identify all candidates owned by t
8. for each itemset candidates c € Ct take step 9
9. A (c)= α(c) +1 %raise the support count
10. Fk={c|ck ^ α(c) >= N x minsup} % extracted frequent k-itemset .
11. result = U Fk

Fig. 1. Frequent itemset generation algorithm

- *Rule Generation*

This stage aims to extract rules that have a Confidence value above a certain value. The resulting rule is the final output of association mining which will later be considered as a relationship between the items. Apriori algorithm for rule generation can be seen in the picture below.

1. for each frequent k-itemset Fk, k > 2
2. H_t = {t | t € f_k } {1-item consequents of the rule,}
3. call ap-genrules(f_k, H_1)
4. end for

Fig. 2. Apriori algorithm for rule generation

E. Supervised Learning

Supervised learning is carried out when the expected output has been known beforehand. Usually this learning is done using existing data. Supervised learning is a method used to find the relationship between input attributes (can be referred to as independent variables) and target attributes (can be referred to as dependent variables). The relationship is considered as a representation of the structure called a model.

Supervised learning usually has attributes and labels. From the known attributes and labels of the data, we can make a model. The model can then be used to classify further testing data. In sentiment analysis, supervised learning is used in the classification process to determine the polarity of an opinion sentence, whether the sentence is oriented positively or negatively [8].

F. Random Forest

Random forest is a classifier that evolves from decision tree. In random forest, decision tree has been training done using individual sample and each attribute is broken down in a thee chosen between random subset attributes and in the classification process, each individual

is based on the most votes in the population tree collection [9].

Given an ensemble from classifiers *h1(x),h2(x), ... ,hK (x),* and the training set which is taken randomly from random vector distribution. Y,X defined the margin function as follows,

$$mg(X,Y) = av_k I(h_k(X) = Y) - max_{j \neq Y} av_k I(h_k(X) = j)$$

wheres I(,) is the indicators function. Margin measures the extent to which the average number of votes in X,Y for the right class exceeds the average vote for the other classes. If the margin value is greater, then the onfidence value will be greater in the classification process. For generalization errors indicated by

$$PE^* = P_{X,Y}(mg(X,Y) < 0)$$

Wheres subscript X,Y indicates that the probability exceeds X,Y space.

G. Feature Based Opinion Summarization

Minqing Hu and Bing Liu carrying a system that aim to summarize a series of comments or opinions about the features of a product. The steps taken to build the system are:

- Extracting product features in a series of reviews
- Classifying the product features, whether positive or negative orientation
- Summarizing features that have been successfully classified as positive or negative [3].

The complete overview of the system that minqing and Bing Liu made can be seen in the diagram below.

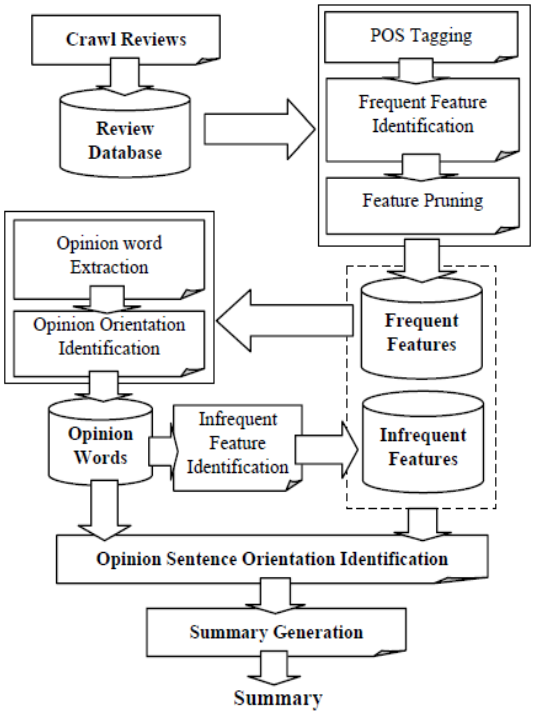

Fig. 3.Feature based opinion summarization [3]

An example of summarizing the feature opinions generated by the system can be seen in figure below.

```
Digital_camera_1 :
    Feature: picture quality
        Positive : 253
            <individual review sentence>
        Negative : 6
            <individual review sentence>
    Feature: size
        Positive : 134
            <individual review sentence>
        Negative : 10
            <individual review sentence>
```

Fig. 4.example of the opinion summarization results

H. Evaluation

Evaluation calculations can be done at the sentence and document level. Calculation of product feature extraction results is done using document-based evaluation, while classification evaluation is done using sentence-based evaluation. Following is an explanation for each evaluation approach.

➢ *Document-Based Evaluation*

The results of product extraction are grouped into one document before an evaluation calculation is performed. In the product feature extraction process, the extracted product features are collected and compared with a list of features that should be extracted (expert judgment). The following is an example of document-based evaluation calculation.

Extracted Feature	Expert Judgement Feature
1. Camera	1. camera
2. picture	2. picture
3. macro	3. macro
4. day	4. size
5. feature	5. weight
6. manual	6. feature
7. battery	7. manual
8. scene	8. auto focus
9. mode	
10. situation	

Table 1:- Example of Document Based Evaluation

The features listed in the table above are a list of product features in all sentences in the review dataset. From the comparison between extracted product features and expert judgment product features, there were 5 true product features out of 10 extracted product features. The number of extracted product features exceeded the number of product features it should have, which was 8. Therefore, the values of precision and recall can be determined for the data above. The precision value is 0.5, and the recall was is 0.625.

➢ *Sentence-Based Evaluation*

The evaluation conducted on the classification aims to determine the extent of the performance of the random forest method in supervised learning. This is done by calculating the amount of test data whose class is predicted

to be correct by the system. Evaluation calculations are done using accuracy. This accuracy shows the value of the measurement results with the actual value. This accuracy is defined by the equation,

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

No.	Polarization of Feature Extraction Process	Polarization of Prediction Feature	Accuracy
1.	battery life [+]	battery life [+]	1.0
2.	-	-	1.0
3.	battery life [-]	battery life [-]	1.0
4.	battery life [+]	battery life [-]	0.0
5.	battery life [+] camera [-]	battery life [+]	1.0
6	battery life [+]	battery life [+] battery life [-]	0.5
Rata-rata			4.5/6 = 75%

Table 2:- Example of evaluation of accuracy in classification

The table above shows an example of an evaluation calculation on the classification results, where the accuracy of each sentence or row is calculated and then averaged by the total sentences that have product features that match the extraction feature label. The classification process involves pairs of product features and opinions that are extracted and correct according to the label on each product feature.

III. RESEARCH METHODS

A. System Overview

The system built in this study is a system that can determine the positive or negative orientation of an opinion on a product feature based on existing comment data, which will eventually produce a summary to facilitate the reading of the comment / review data. This system has three main steps: the extraction of product features from the preprocessed comment data, the classification using several attributes to determine the positive or negative orientation of the feature, and the generation of summaries. The complete overview of the system can be seen in the diagram below.

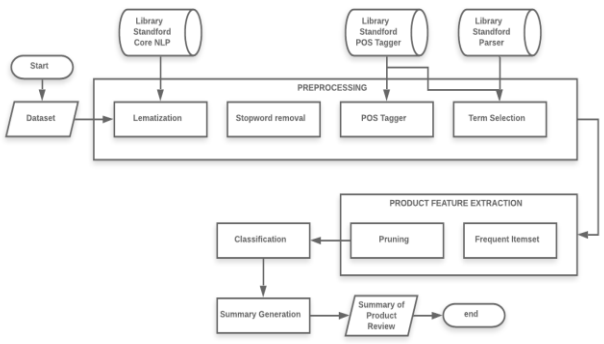

Fig. 5.The complete overview of the system

B. Product Feature Extraction

Product feature extraction is done using Association mining to find frequent itemset.

➢ *Frequent Itemset*

A priori process was carried out using the spmf library, where the process involved calculating the appearance of each word in each sentence and the results must be in accordance with the predetermined threshold. The threshold range used in this process was 0.3% to 1.9%, where the author extracted only the results of frequent itemset from each of these datasets with a threshold of 1% for the next process.

Nouns or noun phrases that have been obtained from the previous process would be encoded for each word in each dataset. That code would be input to the built a priori algorithm. Then the output of a priori processing would appear. The following is an example of the extraction process carried out by the system. For example, there are several sentences from the Canon G3 dataset as shown in Table 3.

No	Sentences
1.	*i recently purchased the canon powershot g3 and am extremely satisfied with the purchase*
2.	*the camera is very easy to use in fact on a recent trip this past week i was asked to take a picture of a vacationing elderly group*
3.	*after i took their picture with their camera they offered to take a picture of us*
4.	*i just told them press halfway wait for the box to turn green and press the rest of the way*
5.	*they fired away and the picture turned out quite nicely as all of my pictures have thusfar*

Table 3:- Example of apriori process

From the data above, the noun or phrase noun selection process is then carried out, and coding or numbering is done to each word from the noun or phrase selection results. Examples of numbering of data where the noun has been selected can be seen in Table 4.

No	Sentences from Noun Selection Result	Apriori Process Numbering
1.	*canon powershot g3 purchase*	1 2 3 4
2.	*camera fact trip week picture vacation group*	5 6 7 8 9 10 11
3.	*camera picture*	5 9
4.	*box rest*	12 13
5.	*picture picture thusfar*	9 14

Table 4:- Apriori Numbering on the Noun Selection Result

Previously the minimum support had been set at 2. Then the support of the appearance of each word is calculated, as shown in figures below.

Itemset	Support
{1}	1
{2}	1
{3}	1
{4}	1
{5}	2
{6}	1
{7}	1
{8}	1
{9}	3
{10}	1
{11}	1
{12}	1
{13}	1
{14}	1

Itemset	Support
{5}	2
{9}	3

Itemset	Support
{5,9}	2

Fig. 6.Result of Apriori of Itemset-1

Items that do not meet minimum support of 2 will be discarded and not processed. After that the second itemset generation process is carried out with a combination of two aspects, as seen below:

Because the 2 itemset combination still met the minimum support, the above combination was entered into a priori results. However, because only two itemset passed, the combination of three itemsset was not carried out. The system built may have a combination of 3 or even 4 itemset, depending on the data entered into the system. After obtaining the results of the priori of itemset {5}, {9}, {5,9}, then the itemset was translated into words according to the previous itemset numbering so that the product features extracted were "camera", "picture", and "camera picture".

➢ *Pruning*

The words that appear together in a certain order in human language usually give a certain meaning or what is commonly called a phrase. Associating mining itself produces a set of words that appear alone or together in a sentence, no matter the closeness between words in the sentence. So it's possible to produce a combination of words that have no meaning. Therefore compactness pruning is done to overcome these problems. Compactness pruning is done to eliminate the combined words. Eliminating candidate features can be done by calculating the distance between the two words that appear.

In compactness pruning, there are two parameters for eliminating product candidate features which have two or more words. The parameters used for compactness pruning are minimum distance and minimum occurrence. What is meant by minimum distance is the minimum distance between two words that are candidates of product features which are considered to be true product features. Whereas the minimum occurrence is the minimum number of product features appearing in the sentence which are considered to be the correct product features. This is algorithm of compactness pruning.

```
For each sentences do
   For each featurePhrraseInSentence do
      Words <- Tokenize (FeaturePhraseInSentence)
       Calculate distance between two words
       If distance > minDistance then
            Add to listNewFeaturePhrase
       EndIf
  EndFor
EndFor
For each featurePhrase do
   Count feature occurrence in listNewFeaturePhrase
   If occurrence < minOccurrence then
       Remove(featurePhrase)
    EndIf
EndFor
```

Fig. 7.Compactness Prunning Algoritthm

The Algorithm for calculating the distance between two words used in the compactness pruning process can be seen in the picture below,

```
For each tokenSentence do
If tokenFeature[0] equals tokenSentence[i] then
flag <- flag + 1
IndexWord1 <- i
EndIf
If tokenFeature[1] equals tokenSentence[i] then
flag <- flag + 1
IndexWord2 <- i
EndIf
IF (flag == 2)then
Distance <- IndexWord2 – IndexWord1
EndIf
EndFor
```

Fig. 8.Calculation of distance between words algorithm

In candidate features that consist of one word, redundant features must be eliminated (redundancy pruning). If the pure support is smaller than the minimum pure support and the identified candidate feature is a subset of the other candidate feature phrases, then the candidate feature will be eliminated. Pure support is the number of sentences that have candidate features in the form of words or phrases and that have no candidate feature phrases that are supersets of the candidate features. For example, the "life" candidate feature will not be considered a meaningful feature, while "battery life" is a feature phrase that is considered to have more meaning and so it is considered to be a product feature. If this "life" fulfills the condition to be called redundant feature, then the "life" feature will be eliminated. The algorithm of redundancy pruning can be seen in picture below,

```
For each sentences do
 For each featureSingleWord In Sentence do
  Check whether its superset appear in sentence
  If not appeared then
    Increment pureSupport of featureSingleWord
  EndIf
 EndFor
EndFor
For each pureSupport of featureSingleWord do
 If pureSupport of featureSingleWord < minPureSupport
then
Remove(featureSingleWord)
 EndIf
EndFor
```

Fig. 9.Redundancy Pruning algorithm

In the example above we have already got 3 candidate features from the Apriori process to determine the frequent itemset of "camera, picture, camera picture". Then in the pruning process, only 2 candidate features were found as the "camera picture" was not considered a meaningful candidate feature because it did not meet the minimum distance (threshold) determined previously. So the candidate features produced up to this stage were "camera and picture".

C. Identification of Product Feature Opinions

Until the process of product feature extraction with association mining methods, the next step is matching process the match product feature with the corpus. Corpus here is a list of product feature from hand labeling (expert judgement) that already exist in dataset. Matching product features with corpus is performed in the dataset and aims to identify product features in the opinion document along with labeling the class intended for the classification process.

Example of corpus:

auto mode[+], manual mode[-], scene mode[+],

candidate of product feature :

"auto mode". "manual mode", "scene mode"

Identified of product_feature :

Auto mode [+], manual mode[-], scene mode[+]

D. Product Feature Classification

After the product features or aspects are obtained, the next process is to classify the product features based on their polarity. The classification conducted in this study used the supervised learning method. In supervised classification, labels are needed to build a model on training data. Then a testing is done to the model to be built. This process is called data testing. In this study the classification process was built using the random forest algorithm. Attributes or features for random forest input were generated from the previous extraction process. Attributes or features to be used and arranged in the table form are product features or product aspects, product features in sentences, and their labels. The following is an explanation of these attributes.

- *Product features or product aspects are product features that are identified through the previous feature extraction process. Examples of product features are "camera", "screen", "feature", "camera resolution", etc*
- *Product features attributes in sentence are product features that are combined with the local context. This local context is itself a before and after term of certain product features. In this study, only one before and after term was used the intended product features. Examples of the formation of product feature attributes in sentences can be seen in the Table 5.*

Fitur Produk	Local Contex Before	Local Contex After	Fitur Produk dalam Kalimat
camera	made	easy	made camera easy

Table 5:- Example of Sentence Feature Formation

Features obtained from the two ways above will be labeled directly with the intended aspects. The label of the corpus aims to be used in training data in the model development process. Figure 3 illustrates the classification process carried out on the system

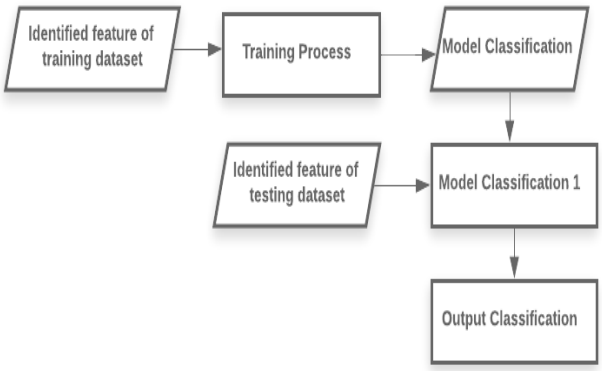

Fig. 10. Product Feature Classification Process

From Figure 10, it can be seen that the classification training process originates from the product features identified in the training dataset. The training process then produces a model that can be used for the learning process. To test the model we need the identified product features of the test dataset from the same extraction process as the training data. In this test dataset, the model that has been built is learned and an output classification will appear and be calculated for its accuracy value to find out how good the model is.

The following is an example of identified product features to be inputted into the random forest classification of product feature extraction.

Sentence	Product Feature	Product Feature in a Sentence	Label
the only two minor issues i have with the camera are the lens cap it is not very snug and can come off too easily and the lens itself it partially obstructs the view through the viewfinder but not views through the lcd	viewfinder	Viewfinder views	-
	Lens cap	Lens cap minorl	-

Table 6:- Example of Sentence Feature Formation

E. Summary Generation

Summary generation is the final stage of this research, where the results of extraction and the classification of positive or negative orientation towards a product feature are formed in an easy-to-read format. The summary built on this system is a summary of product reviews generated using an extractive summary approach that is easier if it has many data sources. It differs from the abstractive approach which produces a summary by paraphrasing the review.

The summary of this research display information on product names, product features, sentiment orientation based on product features, opinion sentences and the number of positive and negative reviews on product features. Following is an example of the results of the opinion sentence polarity test.

Output of the summary :

Sum of positive opinion : xxxx
camera[+] camera perfect enthusiastic amateur photographer
picture[+] picture razorsharp macro
macro[+] picture razorsharp macro
feature[+] operate feature easy obvious i be annie lebovitz figure ability mess camera store
manual[+] manual fine job fill blank remain

Sum of negative opinion : xxxx
auto mode[-] camera autofocussing auto mode buzz sound can not stop
image[-] slightest shake totally distort image
lens cap[-] lens cap annoy
movie[-] movie clip ' noise ' can not avoid
closeup shooting[-] good camera ' good ' picture clarity exceptional closeup shooting capability servicing[-] send camera nikon servicing 6 week diagnose problem

IV. RESEARCH RESULT

A. Dataset Exploration

The dataset used consists of five documents, each of which contains a collection of comment sentences for each product. This documents come from Amazon's Electronic Product Reviews in which one document represents a product that is commented on. These products include Apex AD2600 Progressive-scan DVD Player, Canon G3, Creative Labs Nomad Jukebox Zen Xtra 40GB, Nikon Coolpix 4300 and Nokia 6610. Each document has been given a manual labeling (human tagger), namely labeling in the form of product features and the value of their opinion orientation. Manual labeling have been done in previous research by Hu Minqing and Bing Liu.

The dataset used consists of several reviews where each sentence is cut and one line represents one review sentence. Each review sentence has a label containing a product feature with two conditions, explicit or implicit. Implicit product features have the [u] or [p] tag along with their orientation information. Implicit product features on the system cannot yet be overcome. So when there is a product feature label that is followed by [u] or [p] tag it will be deleted and not entered into the system.

Not all product review sentences have product feature labels and their polarity orientation. In each document there are approximately 50% to 60% of the review sentence that does not have a product feature label. This is possible because in the product review sentence there is no product feature commented on or there is expert judgment errors in manual labeling and giving opinion orientation to each sentence manually. Following are the detailed information obtained from the dataset used in this research.

No	Documents	Sum of sentences	Sum of sentence without features	Sum of product features	Implisit	
					Sum of feature with [u] tag	Sum of feature with [p] tag
1	*Apex AD2600 Progressive-scan DVD Player*	740	386	817	37	45
2	*Canon G3*	597	358	644	20	10
3	*Nikon Coolpix 4300*	346	186	389	14	4
4	*Nokia 6610*	546	280	620	24	5
5	*Creative Labs Nomad Jukebox Zen Xtra 40GB*	1716	977	1825	60	52

Table 7:- Detailed Information of Each Dataset

The dataset entered into the system is still very raw, with many tags that are not needed in further processing and a lot of misspelling. However, in this study the original dataset was still used. An example of misspelling in a dataset is "canera" which probably means "camera". The system does not handle misspelling errors because the form of misspelling can vary from word to word.

B. Noun Selection Analysis

The initial process carried out by the system is reading the data then preprocessing, which is to get clean data to be processed in the next step. The preprocessing stages include lemmatization, stopword removal and post tagging. Data that has gone through preprocessing will then go through a term selection process. In this study two word selection processes were carried out, namely nouns and noun phrases. Noun selection was done because about 80% of the product features were nouns. The noun selection is done for the next process, which is taking product candidate features using frequent itemset. In addition to nouns, existing product features can be a combination of nouns and other words. This means that not only nouns are included in the product features, and therefore a noun phrase selection is also done. Here is an evaluation comparison table on two test schemes.

Dataset	Noun			Noun Phrase		
	Prec (%)	Rec (%)	F-src (%)	Prec (%)	Rec (%)	F-src (%)
Apex DVD Player	33.71	30.30	31.91	37.93	33.33	35.48
Canon G3	25.23	27.72	26.42	27.97	32.67	30.14
Nikon Coolpix	23.42	38.81	29.21	20.45	40.30	27.14
Nokia 6610	41.11	37.00	38.95	39.29	33.00	35.87
Zen Mp3 Player	43.62	23.16	30.26	43.33	22.03	29.21

Table 8:- Comparison of Word Selection Accuracy

The values of precision and recall obtained from the nouns and noun phrase test schemes in each dataset with a minimum support of 1% were in the range of 20% and 40%, with the resulting evaluation values relatively small. This happened because there were still many data variants in the document that were not in accordance with the good and correct English structure. In addition there were still many comments given by customers which did not directly comment on product features, or in a sense, the product feature comments were implicit. A relatively small evaluation value was also caused by labeling errors in the dataset created manually by expert judgment so that during the evaluation calculation, the candidate features extracted from this process were not in the dataset's label, and during the evaluation calculation, the extracted features were considered incorrect.

Based on the table above, the resulting f-score values of the two schemes appear to be balanced without any striking advantages. The f-score values on the noun phrase are only higher on Apex DVD Player and Canon G3 data while on the other three the f-scores are smaller compared to noun selection. This means that the number of product features resulting from extraction with the two word selection schemes is more of nouns than noun phrases. The f-score that looks striking is the Nokia 6610 dataset in the noun selection. It also results in almost balanced prec and recall which shows that the combination of data in the Nokia 6610 dataset is good enough. The following is a comparison table of the number of features extracted in each dataset with two word selection schemes.

Dataset	Noun		Noun phrase	
	minsup	fitur	minsup	fitur
Apex DVD Player	1,00%	89	1,00%	87
Canon G3	1,00%	111	1,00%	118
Nikon Coolpix	1,00%	111	1,00%	132
Nokia 6610	1,00%	90	1,00%	84
Zen Mp3 Player	1,00%	94	1,00%	90

Table 9:- Comparison of Word Selection Accuracy

At the time of product extraction, minimum support for each test scheme will be distinguished. In the minimum support determination, the number of features extracted in a dataset will also be determined. By using a minimum of 1% support, it is clear that the number of features extracted is approximately 84 to 132 in each dataset. The number of features extracted will also affect the resulting evaluation value, because the evaluation calculation on the product feature extraction is document based. This can be seen in the extracted product features and product features in the dataset. Based on the evaluation values above, the two word selection test scenarios are good, but some rules still need to be added so that the extracted features are true product features contained in the dataset. The following figure shows the results of the calculation of accuracy (F-score) on the five datasets

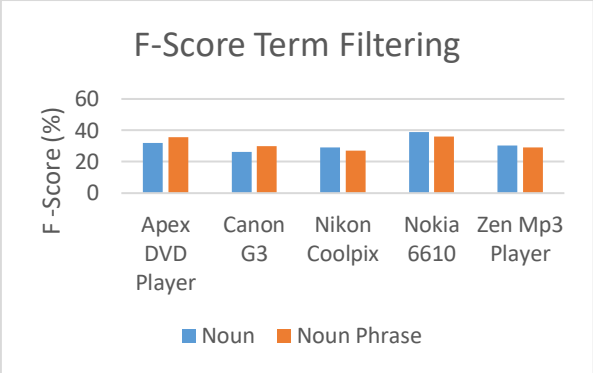

Fig. 11. F-score Term Filtering for each dataset

C. Product Feature Extraction Analysis

Feature extraction is done using a priori algorithm with the help of SMPF library in the java programming language. Testing at the product feature extraction stage is carried out by applying different minimum support values in the range of 0.3% - 1.9%. The minimum value of this support can affect the number of product feature candidates extracted in each dataset. The smaller the minimum support value, the higher the number of extracted product feature candidates will be. The number of extracted product feature candidates will also affect the value of precision. If the number of extracted product feature candidates exceeds the number of product features contained in the corpus dataset, the value of precision will be less than the recall value. Here is a table of features extracted in the Apex DVD player dataset with noun selection and with a minimum support range of 0.3% - 1.9%.

Min Support	Count of extracted	Sum of matched	Precision	Recall	F-Score
0,003	385	49	12,73%	49,49%	20,25%
0,004	385	49	12,73%	49,49%	20,25%
0,005	224	41	18,30%	41,41%	25,38%
0,006	151	36	23,84%	36,36%	28,80%
0,007	151	36	23,84%	36,36%	28,80%
0,008	118	31	26,27%	31,31%	28,57%
0,009	89	30	33,71%	30,30%	31,91%
0,01	89	30	33,71%	30,30%	31,91%
0,011	79	28	35,44%	28,28%	31,46%
0,012	67	26	38,81%	26,26%	31,32%
0,013	67	26	38,81%	26,26%	31,32%
0,014	59	24	40,68%	24,24%	30,38%
0,015	52	22	42,31%	22,22%	29,14%
0,016	52	22	42,31%	22,22%	29,14%
0,017	47	21	44,68%	21,21%	28,76%
0,018	39	19	48,72%	19,19%	27,53%
0,019	39	19	48,72%	19,19%	27,53%

Table 10:- Evaluation of Noun Selection and Minimum Support Differences In Apex Dvd Player

In the table above, it can be seen that the highest f-score results are stable at minimum support of 09% and 1% with a value of 31.91%. A smaller support minimum indicates more extracted features, resulting in decreased precision and increased recall. Conversely, a larger minimum support will show a smaller number of extracted features, resulting in increased precision and decreased recall. So minimum support becomes a very important variable to know the number of features to be extracted. The following is a figure of product feature extraction in the Apex DVD Player dataset with noun retrieval selection,

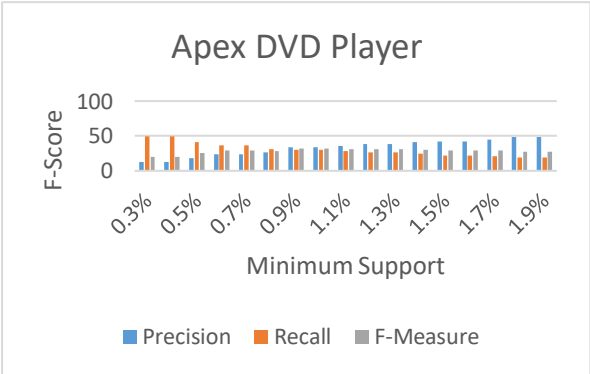

Fig 12

Seen from the figure above the value is stable at the minimum support of 0.9% and 1%, continues to decline at the minimum support of 0.8% and so on, and continues to

rise at the minimum support of 1.1% and so on. That is because more product candidate features will be extracted if the minimum support is smaller and less product candidate features will be extracted if the minimum support is greater. This is in contrast to recall.

The number of sentences in the dataset can also affect the determination of the minimum support to get the results of product feature extraction that matches the number of extracted candidate features. The greater the number of sentences in the dataset, the greater the appropriate minimum support, and vice versa. This happens because the higher the number of sentences in a dataset, the higher the number of occurrences of words in the dataset. In the product feature extraction process using a priori algorithm, determining minimum support is very important to determine the extracted product feature candidates that are in accordance with their frequent itemset.

The following are examples of images that show the results of precision, recall, and F-Score in the smallest dataset, which is Nikon Coolpix with 346 sentences, and the largest dataset, which is Zen Mp3 Player with 1716 sentences.

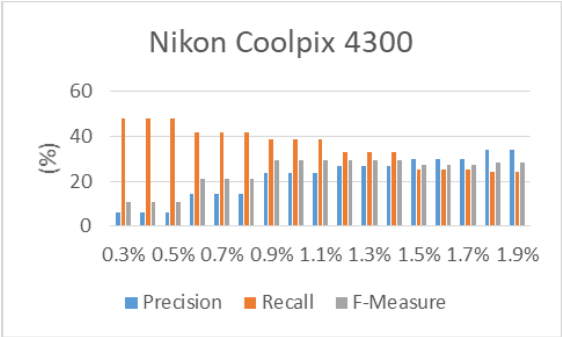

Fig 13:-Extraction of Nikon Coolpix's Datasett Product Features

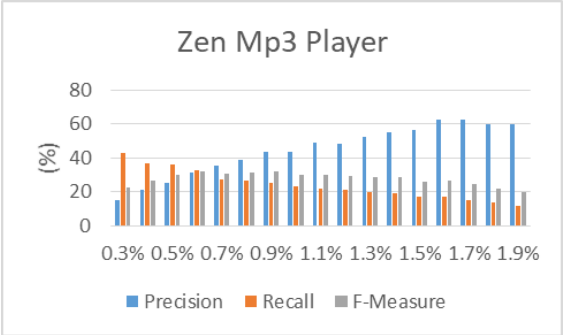

Fig 14:- Extraction of Zen Mp3 Player's Datasett Product Features

The difference in the number of sentences in the dataset can be a parameter in analyzing to determine the appropriate minimum support. The minimum support value that produces the greatest F-Score on the Nikon Coolpix dataset is at 0.9% - 1.4%, while on the Zen Mp3 Player dataset, it is obtained at 0.9%.

Frequent itemset obtained from apriori algorithm is product candidate features. These product candidate features will later enter the next process, namely pruning, where product candidate features that are not suitable to be product features are re-selected. Product candidate features extracted from frequent itemset may be less meaningful to be product features because the distance between words is too far (does not meet the minimum distance given) or word redundancy occurs. There are two types of pruning conducted in this study, redundancy pruning and compactness pruning, with parameters such as minimum distance, minimum occurrence and minimum pure support. Product features that do not comply with the specified parameters and threshold will be deleted. Determination of the appropriate parameter values can produce good pruning results so as to increase the evaluation value of the product feature extraction. The following are the results of pruning testing on the Nokia 6610 dataset using a different threshold value for each parameter,

Min Distance	Min Occurance	Min Pure Suport	After			Δ Prec (%)	Δ Rec (%)	Δ Fsc (%)
			Prec(%)	Rec(%)	Fsc(%)			
1	0	1	41,11	37,00	38,95	0,00	0,00	0,00
1	3	1	51,39	36,27	42,53	10,28	-0,73	3,58
1	5	1	52,17	35,29	42,10	11,06	-1,71	3,15
3	0	1	41,11	37,00	38,95	0,00	0,00	0,00
3	3	1	46,84	37,00	41,34	5,73	0,00	2,40
3	5	1	50,00	37,00	42,53	8,89	0,00	3,58
1	0	3	40,91	36,00	38,30	-0,20	-1,00	-0,65
1	3	3	52,17	36,00	42,60	11,06	-1,00	3,66
1	5	3	53,03	35,00	42,17	11,92	-2,00	3,22
3	0	3	40,91	36,00	38,30	-0,20	-1,00	-0,65
3	3	3	47,37	36,00	40,91	6,26	-1,00	1,96
3	5	3	50,70	36,00	42,10	9,59	-1,00	3,16

Table 11:- Testing of Pruning Parameters on THR Nokia 10 Dataset

Based on the table above, with a combination of minimum distance, minimum occurrence and minimum pure support, the Δ F-score in the Nokia 6610 dataset is quite significant compared to if it does not use the pruning process. The biggest Δ F-score obtained is 3.58%. This pruning process does not significantly affect recall, but can cause recall to decrease. This is because the eliminated features are candidates for the correct product features in the dataset. Conversely, this pruning process is quite influential to increase the precision because as the number of extracted candidate features decreases, so does the divider in the precision calculation.

Testing on each dataset is done using document-based evaluation calculations with the results of the F-score in the range of values 20% - 40%. The results of the F-score for each dataset are shown in the following table.

Dataset	Apriori			Pruning		
	Prec(%)	Rec(%)	Fsc(%)	Prec(%)	Rec(%)	Fsc(%)
Apex DVD Player	33,71	30,30	31,91	41,43	29,29	34,32
Canon G3	33,33	24,75	28,41	35,90	27,72	31,28
Nikon Coolpix 4300	26,83	32,84	29,53	31,25	37,31	34,01
Nokia 6610	36,97	44,00	40,18	51,39	36,27	42,53
Zen Mp3 Player	43,27	25,42	32,03	48,81	23,16	31,41

Table 12:- Product Feature Extraction Evaluation Result

From the table above it can be seen that the F-score value in each dataset from the a priori process to the pruning process is always greater except for the Zen Mp3 Player dataset. The F-score value of Zen Mp3 Player dataset in the a priori process is greater than in the pruning process. This happens because there are too many sentences in the Zen Mp3 player document and the precision generated is quite high. After the pruning process, the resulting accuracy evaluation is quite improved. This is because in the pruning process there is a trimming of the candidate product features that do not meet the specified threshold. The table above is taken from the greatest accuracy value from the a priori process and the greatest value from pruning accuracy. So the pruning process can be used to improve the accuracy evaluation of the product feature extraction process using frequent itemset.

D. Classification Analysis

Product features resulting from the extraction process will then become one of the inputs or attributes in the classification process using a random forest algorithm. Product features and some other attributes will be paired with the appropriate labels on the corpus of the dataset as training data for the construction of classification models. The model will be used on test data. In the process of testing the polarity classification of product features, two tests will be performed. The first test is done by analyzing the effect of the input classification attributes on each of the same training data documents and test data. And the second test is done by analyzing the effect of each input document on the results of classification using a combination of training data and test data for each product review document.

- *Analysis of Classification Attributes*

In this test the classification process will be carried out using the Ranfom Forest algorithm with a crossvalidation model testing using Numfold = 10. In this test three experiments are conducted for each attribute, including product features, product features in sentences, and a combination of both. The training data and test data used are the same dataset for each test. The following are the results of the evaluation using accuracy,

Dataset	Product feature	Product feature in sentences	Product feature + product feature in sentences
Apex Dvd Player	77,94%	54,05%	83,65%
Canon G3	87,52%	60,30%	89,45%
Nikon Coolpix 4300	85,74%	53,76%	88,92%
Nokia 6610	85,84%	52,20%	88,64%
Zen Mp3 Player	81,31%	58,22%	87,37%

Table 13:- Evaluation of Input Attributes in Clasifications

Based on the test results listed in the above table, it can be seen that the result of the combination of product feature attributes and product feature attributes in sentences is greater than the other two schemes. In the product feature attribute testing scheme, a better value is obtained compared to the product feature attributes in the sentence. Product feature attribute schemes can classify opinions more precisely than product feature attributes in sentences. This is because in the product feature attribute, the existing record is in the form of a product feature word, so the

construction of the model will be better, while the product feature attribute in the sentence has a different data record so the classifier will have difficulty training the information in the product feature attribute in the sentence. The resulting model is not good for testing.

The combination of product feature attributes and product feature attributes in sentences consistently has a better value than product feature attributes only. The increase in accuracy in the attributes of a combination of product features and product features in a sentence can reach 6.06%. This can occur because the product feature attributes in the sentence will help add information in the learning (training process) for the model formation so that it can produce better accuracy.

➢ *Analysis of Classification Document input*

In this classification test, a combination of input training data and different test data are used. Here are some combination of documents on the process if training data and test data,

1. **Training dataset** : Apex DVD Player
Testing dataset : Apex DVD Player
Canon G3
Nikon Coolpix4300
Nokia 6610
Zen MP3 Player

2. **Training dataset** : Canon G3
Testing dataset : Apex DVD Player
Canon G3
Nikon Coolpix 4300
Nokia 6610
Zen MP3 Player

3. **Training dataset** : Nikon Coolpix
Testing dataset : Apex DVD Player
Canon G3
Nikon Coolpix 4300
Nokia 6610
Zen MP3 Player

4. **Training dataset** : Nokia 6610
Testing dataset : Apex DVD Player
Canon G3
Nikon Coolpix 4300
Nokia 6610
Zen MP3 Player

5. **Training dataset** : Zen MP3 Player
Testing dataset : Apex DVD Player
Canon G3
Nikon Coolpix 4300
Nokia 6610
Zen MP3 Player

Each test scheme will be compared with the accuracy of the results obtained from the calculation of each line of review sentences, and the average in each document will be calculated. The following are evaluation and accuracy calculations for each input scheme of training data and test data combination.

	Product of Feature				
	Apex DVD Player	**Canon G3**	**Nikon Coolpix**	**Nokia 6610**	**Zen Mp3 Player**
Apex DVD Player	77,94%	60,05%	59,54%	51,65%	66,89%
Canon G3	59,12%	87,56%	82,66%	63,39%	66,80%
Nikon Coolpix 4300	59,73%	81,91%	85,16%	63,49%	66,56%
Nokia 6610	60,81%	71,98%	69,22%	86,05%	66,39%
Zen Mp3 Player	63,67%	71,02%	66,33%	66,56%	81,31%

Table 14:- Calculation of Accuracy Evaluation in the Product Feature Attribute Classification Process

In Table 14 we can see the accuracy values of the product feature attributes and the five document combination input schemes. The greatest accuracy value are obtained in the training data and the same test data, whereas if the training data and test data are different, the accuracy value is relatively smaller. This happens because if the construction of the model from the training data and the test data carried out on the model is relatively the same, the accuracy value will be relatively high

	Product Feature in Sentences				
	Apex DVD Player	**Canon G3**	**Nikon Coolpix**	**Nokia 6610**	**Zen Mp3 Player**
Apex DVD Player	54,05%	59,97%	53,76%	51,28%	58,10%
Canon G3	53,65%	60,30%	53,76%	51,28%	58,10%
Nikon Coolpix 4300	53,65%	60,13%	53,76%	51,28%	58,10%
Nokia 6610	53,51%	59,97%	53,76%	52,20%	57,99%
Zen Mp3 Player	53,65%	59,97%	53,76%	51,28%	58,22%

Table 15:- Calculation of Accuracy Evaluation in the Process of Classification of Product Feature Attribute in Sentences

In Table 15 it can be seen that the accuracy value on the product feature attribute in a sentence has an accuracy value that is not too high compared to the previous attribute. In contrast to the results of previous accuracy, in this test the highest accuracy value occurs in training data and different test data due to the influence of attributes that are product features in the sentence. On the Apex DVD player training data, the highest accuracy value is in the Canon G3 test data, which is 59.97%. This happens because of a classifier error in determining the polarity of the test data of the model being built.

	Product Feature + Product Feature in Sentences				
	Apex DVD Player	**Canon G3**	**Nikon Coolpix**	**Nokia 6610**	**Zen Mp3 Player**
Apex DVD Player	83,65%	64,50%	58,53%	60,12%	66,84%
Canon G3	59,26%	89,23%	82,80%	63,45%	58,10%
Nikon Coolpix 4300	59,66%	81,32%	88,63%	63,49%	66,49%
Nokia 6610	60,79%	73,20%	71,68%	88,83%	65,97%
Zen Mp3 Player	64,26%	70,81%	67,73%	66,94%	87,33%

Table 16:- Calculation of Accuracy Evaluation in the Classification Process of Product Feature Attributes and Product Features in Sentences

In table XII it can be seen that the highest accuracy value is on the same training data and test data, while different training data and test data have relatively smaller accuracy. When compared with accuracy of product feature attributes, testing on product feature attributes and combination of product features and product features in sentences results in relatively higher value.

Generally testing conducted on a combination of documents for training data and test data results in inconsistent accuracy values. That is because the dataset used has different characteristics in each review document. The learning model of training data can be used to test different test data from the previous training data. It can be seen that the average classification accuracy results are more than 50% and it proves that the determination of orientation of electronic product review data opinions can predict different documents that are still within the scope of the product. Product features and product features in sentences in the Zen MP3 Player dataset have a higher predictive value for accuracy in every other document, with an average accuracy value of 69.44%. That is because the number of Zen MP3 player product reviews has an effect on other electronic product reviews.

ACKNOWLEDGMENT

Authors thanks to Mrs Warih Maharani and Mrs Anisa Herdiani as lacturers at Telkom University faculty Informatics Engineering who have guided me to conduct the research on sentiment analysis of product reviews.

REFERENCES

[1]. A. Weichselbraun, S. Gindl and A. Scharl, "A Context-Dependent Supervised Learning Approach to Sentiment Detection in Large Textual Database," Journal of Information and Data Managemment, vol. 1, no. 3, pp. 329-341, 2010.
[2]. D. K. Gupta and A. Ekbal, "Supervised Machine Learning for Aspect based Sentiment Analysis," in the 8th International Workshop on Semantic Evaluation (SemEval 2014), Dublin, 2014.
[3]. M. Hu and B. Liu, "Mining and Summarizing Customer Review," 2004.
[4]. M. kowalski, Information Retrieval Architecture and Algorithms, New York: Springer, 2011.
[5]. A. K. Ingason, S. Helgadottir, H. Loftsson and E. Rognvaldsson, "A Mixed Method Lemmatization Algorithm Using a Hierarchy of Linguistic Identities (HOLI).," in Advances in Natural Language Processing, 2008.
[6]. N. P.Katariya and Chaudari, "Text Preprocessing For Text Mining Using Side Information," International Journal of Computer Science and Mobile Applications, vol. 3, no. 1, pp. 01-05, 2015.
[7]. P.-N. Tan, M. Steinbach and V. Kumar, Introduction to Data Mining, Boston: Addison-Wesley Longman Publishing, 2005.
[8]. O. Maimon and L. Rokach, Data Mining and Knowledge Discovery Handbook, New York: Springer, 2010.
[9]. L. Bremian, “Random Forests,” 2001.