Report On

Mining Opinion Features in Customer Reviews

Submitted By Ratnesh Chandak (CS12B1030), Akshay Jadhav(ES12B1004), Harmanpreet Singh(CS12B1017), Udhav Sethi(ES12B1022), Group 10

Abstract

There are lots of online unstructured text, lots of online data of reviews, blogs, forums and discussion groups are created on daily basis. Need to structure it as well as make it short and precise so that it may come handful to certain people. With increase in online facilities, E-Commerce web sites are providing different kind of services, this project is implemented over dataset of Amazon E-Commerce. The aim of this project is to generate summary of reviews based on only those features which are talked by customers. In this project we are implementing a system to generate feature and its opinion based summary. Our implementation is data independent.

Introduction

With the rapid development of E-Commerce websites more and more people are buying products online, but they fear whether the product is authenticate, whether its quality is good as shown in the website, they are afraid to buy due to lack of trust, the one thing which increases their trust is the reviews written by the customers who had bought the products. Again the customer may fear the the authenticity of the reviews. Aim of this project is not to get authenticate reviews (reviews written by valid user who had bought the product), we are assuming the reviews which we are getting is authenticate. In the project we are generating any product's summary so that the reviews may help any potential customers to take informed decision and it may also help the manufacturer to get to know what more to produce and what are the things they are missing which are demanded by the customers.

In this project our task is divided into 3 major steps:

1. Identifying all the features of any product that customers have expressed their opinion on(those sentence which contain feature as well as its opinion are defined as opinion sentences).
2. Identify the orientation of the opinions whether they are good or bad, here we have consider +1 as positive orientation, -1 as negative orientation and 0 as neutral orientation.
3. Calculating the score of each of each opinion sentences. And at last obtaining the the feature wise summary of opinion sentences.

The feature based summary would look as follows:

Digital_camera_1:

Feature: picture quality

Positive: 253

<individual review sentences>

Negative: 6

<individual review sentences>

Feature: size

Positive: 134

<individual review sentences>

Negative: 10

<individual review sentences>

In above figure, picture quality and (camera) size are the product features. There are 253 customer reviews that express positive opinions about the picture quality, and only 6 that express negative opinions. The <individual review sentences> link points to the specific sentences that give positive or negative comments about the feature.

Feature Extraction

First obtain the dataset, we took amazon dataset. Then we pruned the dataset and we are concentrating only for 16 products, our choice of product selection is based on number of reviews, we took those products whose number of reviews is between 355 and 360.

For obtaining the features you need to find all the nouns, here we are not concentrating on the implicit features, we are only considering the explicit features. So for getting the nouns you have to tag the reviews, we are using stanford pos tagger to tag our reviews. To identify the nouns see those word which are tagged as 'NN' or 'NNP'.

Now we got lot features, so you need to prune those features, first step of pruning is finding the frequent features which we are doing using FPGrowth association mining algorithm. We are using 0.1 as min support value. After this we will remain with filtered frequent features. Now prune the compact features (compact features are those which comprises of more than one word, and prune all its substring if it is present) if any, in our case we didn't obtain any compact feature as our dataset is small. These will be our potential features.

Opinion Extraction

Take all the features which we obtain after pruning and find opinion sentences containing these features. To search for opinions look for adjectives, our pos tagger has already tagged our reviews, find 'JJ', 'JJR' or 'JJS' in above sentences. Find all the nearest adjectives of the

resultant features from the opinion sentences. This will be corresponding opinions expressed by customers.

Finding Opinion Orientation

This was the most difficult part, this is a semi-automatic process. We created a seedlist of size 30 of most frequently occurring adjectives and manually tagged those and using those seedlist adjectives search all its synonyms and antonyms. Synonyms are of same orientation and antonyms are of opposite orientation. For getting the synonyms and antonyms we are using wordnet dictionary. The wordnet dictionary words are saved as 2 clusters, all the synonyms are together and antonyms are together.

Scoring opinion sentences

Considering only those opinion sentences which contains our resultant feature set. Calculate the score of each sentences by adding all the score of individual feature's opinion.

Generating Summary

We form the summary in the output file by aggregating positive opinion sentences and negative sentences of individual feature.

Experiments

A system to summarize reviews based on features was implemented using Java and MySQL. A database schema was designed based on the review set.

First step was to break the reviews for each product into sentences. Then tagging each sentence using the Stanford POS tagger. The output of each word and its POS tag was stored in a database. We are only interested in nouns which would we have considered as features and adjectives which we have considered as opinions.

Second step was to generate frequent features as this will be the features which will have an impact in the summary we will generate. For this, we used associate mining to generate set of nouns which would be more frequent and would be considered our potential features. For each sentence, a transaction was created which would be a set of all the nouns in that sentence. These transactions were then passed to spmf algorithm FPGrowth to do associate mining and have used minsup = 0.01. The output was a text file containing all the frequent features and their respective supports. We did not apply any feature pruning methods as the results we got were already 1 word features, and didn't contain any feature phrases.

Third step was to get all opinion sentences. So we search for an opinion which is an adjective which would be near a feature from what we have got above that is the distance between that opinion and feature should be less than 5. If we get an opinion in a sentence, that sentence would be our opinion sentence. The output was all the opinion sentences. We got about 1602 distinct opinions across all the 16 products' review.

Fourth step was to predict the orientation of the opinion word. For this create seedlist of frequent opinion words in our database and assigned them an orientation based on average orientation of orientations we four had given to each opinion word. Then we calculated orientation of opinion words in our database using Wordnet. The idea was that an opinion word would have same orientation as its synonym and an opposite orientation as its antonym. The output was opinion words and its orientation and were stored in the database. We could predict 822 opinions and rest could not be decided. But this 822 opinions spanned across 87% of the opinion sentences, so the prediction of opinion words was good as this was our most frequent opinions.

Fifth step was to calculate orientation of the opinion sentences which we generated in step 3. For this we calculated two orientations O1 and O2 for each opinion sentence. An effective orientation for each opinion word was calculated first. If a negation word i.e not, ain't was nearby an opinion word, then its effective orientation would be opposite its current orientation. O1 is the sum of effective orientations of distinct opinions in an opinion sentence. For each feature in the opinion sentence, the sum of effective orientations of respective adjacent opinions was calculated and was stored as O2. The orientation of an opinion sentence would be equal to O1, if O1 is non-zero. If O1 is zero, orientation of an opinion sentence would be O2, if O2 is non-zero. If O2 is also zero, then orientation of an opinion sentence is equal to orientation of the previous opinion sentence of the same product.

Sixth and the final step was to generate feature based summary.

Conclusion

We generated the summary in the output file by aggregating positive opinion sentences and negative sentences of individual feature.

Limitations

1) The POS tagger does not tag words with 100% accuracy. eg.: "2nd" was tagged as adjective which is used as an opinion word in our project.
2) Finding orientation of the opinion words. As mentioned above, due to POS tagger limitation, some words cannot be predicted as there is no synonym and antonym for "2nd".

3) We have considered only adjectives as opinion word but verbs can also be opinion word.
4) All the opinion words were considered of the same strength.

References

1) https://www.cs.uic.edu/~liub/publications/**kdd04-revSummary.pdf**
2) http://nlp.stanford.edu/software/tagger.shtml
3) http://www.philippe-fournier-viger.com/spmf/index.php
4) https://wordnet.princeton.edu/
5) http://rednoise.org/rita/rita.zip