Topic Extraction from Reviews to improve Product Design

Name: Shahinur Alam
Course: EECE 7991
Title: Projects I
Electrical & Computer Engineering
University of Memphis, Memphis, TN USA

Abstract

With the rapid growth of user generated online reviews, extracting topic and detecting polarity of opinions about a product is an interesting problem. A topic is a combination of attributes (keywords), which captures important characteristics of the data. The experience of the expert users is beneficial for the future users to choose the best product. In this project, I am emphasizing on topic extraction, polarity detection and clustering document so that the designer and the customer can retrieve relevant information easily and quickly from the product reviews.

Keyword: Text mining, Topics Extraction, Machine Learning, Matrix Decomposition

1. Introduction

Nowadays, the Web has become an excellent way of expressing opinions about almost everything. Social media, blogs, and App store reviews are primary sources of information for various product and applications. The availability of online reviews facilitates the customer to choose [Hu *etal.*, 2008][2] a right product which increase the decision potentiality [Zhu and Zhang, 2010][1] .However , it is very cumbersome to read all the reviews of a product from a huge dataset. Therefore, it is very difficult to find the convenient product or app from a bag of artifacts. Now the question is how we can digest the reviews so that designer and customer will be benefited. We can answer the question by finding the contents of reviews or what the expert users writes in the reviews. Generally, reviewer cites information related to particular features, aspect, usability and accessibility of a product in the reviews with a rating. Let us see following review of the camera to identify that information.

**** Our previous camera was a Canon Powershot and when we decided to get a second camera we wanted to stay in the same family. It is **easy to use** (**Usability: very easy**) and perfect for my needs - **vacation pictures** and blog pictures.It is small enough to **slip inside pocket** (**implicit aspect: camera size**) so it can go anywhere. Plus it doesn't take a lot of explaining when you ask someone else to take your picture.
Why did I give it 4 stars if I love it so much?

1. No starter **memory card** (**demanded feature: memory card**) included.
2. No **USB cable** included (so no way to download your photos to your computer without spending more).
3. I've gone through 3 or 4 sets of batteries in a month's usage. (**aspect: battery life**)
4. The **display screen** does not always show the same quality as the picture (the picture is always better).
5. If you are in a situation when you shouldn't use a flash **moving pictures** are **always blurry**.
6. No **voice over** to capture picture (**accessibility: Voice over**)

Figure 1: A sample review for Camera where topics marked as yellow color and cyan color represents underlying category. The rating is stated by number of star.

Figure1 shows that the reviewer has mentioned several topics like “easy to use” which describes “usability”, “slip inside pocket” which is related to “implicit aspect”. Topics like “Memory Card” and “USB cable” indicates future demanded features.

The purpose of this project is to find out the topics related to products potential usefulness considering its accessibility, usability and user experience that will help both buyer and designer. Since all the topics, which conveys meaningful information, and associated polarity of the opinion has been identified the designer will get proper insight to make their product more convenient for users. In spite of this, for the large set of data we need to cluster the document so that the topics and detail information can be found very quickly and efficiently.

2. Topic Extraction and Polarity Detection

The purpose of this task is extracting useful topics, which will provide consolidated information about an entity like the problem of particular features, feature which needs more improvement, customer expectation and needs etc. The whole task has been divided into following steps:

- Review Collection
- Data Cleaning
- Stop word Removal
- Data Annotation
- Multi word topic extraction
- TF-IDF Matrix creation
- Dimension reduction using NMF (Non Negative Matrix factorization) and clustering
- Cosine similarity Measurement
- Most Frequent Single Word Topic Extraction

Almost hundreds of reviews have been collected using Python application from Apple app store by web crawling. Then all irrelevant tags (HTML) and noisy contents have been removed, and the rating has been added at the beginning to all reviews. Reviews are not well organized write up it may have a lot of grammatical error and spelling error. In spite of this, each domain may have own specific keyword or technical jargon. Considering these issue unsupervised method has been applied to extract topic.

2.1 Topic Extraction for Small dataset

Depending on reviews massiveness two strategies has been practiced. First, if the review size is comparatively small then each document has been parsed, tokenized and annotated by Python application which uses libraries from a Natural Language processing (NLP) tools called “Topia Term Extractor”. It also removes all stop words and punctuation marks from the text and stems different words, for example, it consider the word “go”, “goes” and “going” as same because those has the same origin. Sentence based mining has been employed to determine exact polarity of opinion. My focus was on building feature based text mining model which will consider noun, noun phrases, adjective, and verbs. I have considered n-gram (n>=2) topics because most of the cases single noun does not convey meaningful information. In order to generate n-gram topics all n-gram has been collected from the text maintain three rules [Hu and Liu, 2004a][3] (1) adjective followed by noun such as: “good resolution” (2) noun followed by noun such as: “display screen” (3) noun phrases. *N*-gram model predicts x_i based on $x_{i-(n-1)}, \ldots, x_{i-1}$. In probability terms, this is $P(x_i|x_{i-(n-1)}, \ldots, x_{i-1})$.

These Probabilities can be estimated from raw text based on the relative frequency of word sequences.

Bigram: $P(w_n \mid w_{n-1}) = \frac{C(w_{n-1}w_n)}{C(w_{n-1})}$

N-gram: $P(w_n \mid w_{n-N+1}^{n-1}) = \frac{C(w_{n-N+1}^{n-1}w_n)}{C(w_{n-N+1}^{n-1})}$

Here ,$C(W_n)$ denotes frequency of n^{th} word W. and $C(W_{n-1}\, W_{n)\ \text{frequency of}}$ combined occurrence.

The input and output of Topic Extractor is shown below:

- 4|Our previous camera was a Canon Powershot and when we decided to get a second camera we wanted to stay
- in the same family. It is easy to use and perfect for my needs - vacation pictures
- pictures. It is small enough to slip inside pocket so it can go anywhere. Plus it
- doesn't take a lot of explaining when you ask someone else to take your picture.
- Why did I give it 4 stars if I love it so much?
- 1. No starter memory card included.
- 2. No USB cable included (so no way to download your photos to your computer without spending more).
- 3. I've gone through 3 or 4 sets of batteries in a month's usage.
- 4. The display screen does not always show the same quality as the picture (the picture is always better).
- 5. If you are in a situation when you shouldn't use a flash moving pictures are always blurry picture.
- 6. No voice over to capture picture
- --
- 4|Excellent image quality. No voice over service to capture photo
- --
- 3|Fast and easy to use. Poor zoom quality. nice resolution. nice image quality
- --
- 3|Poor zoom quality. battery decays after taking few picture. short battery life

Figure 2: Sample input reviews of camera. Word with red bold letter denotes possible meaningful topics and green color bold number represents rating.

Output:

```
['canon powershot', 1, 2, ' 4']
['4 sets', 4, 2, 4.0]
['usb cable', 1, 2, ' 4']
['vacation pictures pictures', 9, 3, 4.0]
['easy use', 2, 1, ' 4']
['starter memory card', 1, 3, ' 4']
['camera ', 2, 2, ' 4']
['display screen', 1, 2, ' 4']
['photo computer', 1, 2, 4]
['image quality', 2, 2, 3.5]
['short battery', 2, 2, 3.5]
['voice over', 2, 2, 4.5]
```

Figure 3: Output of the Topic Extractor for the above Input reviews. Red color represents irrelevant topics. 1^{st} element of list represents topics, 2^{nd} is number of occurrence, 3^{rd} is number of words in topics, 4rth weighted rating (3=neutral, <3=negative, >3=positive).

Table 1: performance analysis of Topic Extractor

#Meaningful Topics in sample reviews	#Extracted Topics	#Relevant Topic extracted	Precision	Recall
15	12	9	9/12	9/15

2.2 Topic Extraction for Large dataset

Topic extraction for the large dataset is challenging because traditional NLP tools will take a lot of time to process and extract useful information although it has high recall and precision. To resolve this problem document has been clustered and then from that cluster some most frequent topic has been extracted. Clustering is a process to group a set of objects where objects in the same group have similarity. Document clustering is the technique of unstructured data mining where documents in the same cluster have similar topics. Document clustering helps to organize document automatically which reduce information retrieval time. The steps of clustering documents are: (1) term-by-document matrix creation (2) Decomposition of sparse matrix into lower rank.

2.3 Topic/Term weight in Document matrix

In the term–by-document matrix, each "document" is represented as a vector and "topics" represent the dimension. The elements in the vector reflect the frequency of topics in documents. Each topic has weight in the document. These weights can be of two types: Local and global weights. If local weights are used, then topic weights are normally expressed as topics frequencies (tf). If global weights are used, Inverse Document Frequency, IDF values, gives the weight of a term. The following example describes the how to create term weight document matrix using TF-IDF.

Table 2: Topic with weight in five documents

Document	Document 1	Document 2	Document 3	Document 4	Document 5
Topic	camera	camera	camera		
Topic			camera		
Weight	0.2218	0.2218	2*0.2218	0	0

dfi=number of documents containing topic i (“camera”)=3
D = Number of documents=5
dfi /D = Probability of selecting a document containing a desired topic (“camera”)= from a dataset=3/5
IDF=log(D/dfi)= inverse document frequency, represents global information.=log(5/3)=0.2218
weight of a topic= tf * IDF

From the above table, we can see that the weight of topic “camera” in document 4 and 5 is 0. So if a dataset has millions of topics and thousands of documents then the term-by-document matrix may have a large number of elements whose value is zero. This sparse matrix consumes huge storage, and it takes a considerable amount of time to complete any operation. I have built the term-by-document matrix from seven document containing reviews of Apple App, where the total number of topics is: 1499. Sample data from the generated Matrix as follows:

2.0017	0	3.0023	0	7.0011	0	0
5.0009	0.30002	0.30001	0.30001	6.0003	0.50003	0
7.0007	0.0005	5.0003	4.0002	0.60007	0	0
3.0039	0	0	0	0	0	0
5.30051	0	0	0.90011	0.80006	0	0
0.30039	0	0	0	0	0	0

Figure 4: Sample data from term-by-document matrix generated from Apple App review." Documents" are in column and "terms/topics" are in row.

2.4 Sparse Matrix Factorization and Topic Extraction

A topic extraction method creates a new set of topics far smaller than the number of original attributes by decomposing the original data. Therefore, it also enhances the speed of supervised learning for applying any machine learning technique. Unsupervised algorithms like SVD (Singular Value decomposition) or Non-Negative Matrix Factorization (NMF) can be used to decompose the sparse matrix and to find out topics by clustering document. NMF has been used to reduce the rank because "in SVD", the basis vector may have a negative number which is difficult to interpret weight.

NMF decomposes the term-by-document matrix $A_{mn,}$ where columns are text documents and rows are topics , into the product of two lower rank matrices W_{mk} and H_{kn}, such that A_{mn} is approximately equal to W_{mk} times H_{kn}. NMF uses an iterative procedure to modify the initial values of W_{mk} and H_{kn} until the root-mean-squared residual reaches to a threshold or till a specified number of iteration. NMF model maps the original data into the new set of features discovered by the model [4, 5, and 6]. The matrix decomposition can be represented as

$$A_{mn} = W_{mk} \times H_{kn,} \text{ Where } k<<\min(m,n)$$

$$\text{root-mean-squared residual , } D = \text{sqrt}(\text{norm}(A- W_{mk} * H_{kn})/(N*M))$$

$$\text{error}=|A-A_k|^2$$

$$\begin{array}{c} \\ A_1 \\ A_2 \\ . \\ . \\ A_m \end{array} \begin{array}{c} \begin{array}{cccccc} D_1 & D_2 & . & . & D_n \end{array} \\ \begin{bmatrix} A_{11} & A_{12} & . & . & A_{1n} \\ . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ A_{m1} & . & . & . & A_{mn} \end{bmatrix} \end{array} = \begin{bmatrix} W_{11} & . & W_{1k} \\ . & . & . \\ . & . & . \\ . & . & . \\ W_{m1} & . & W_{mk} \end{bmatrix} \begin{bmatrix} H_{11} & . & . & . & H_{1n} \\ . & . & . & . & . \\ H_{k1} & . & . & . & H_{kn} \end{bmatrix}$$

W_{mk} represents basis document vectors and H_{kn} is associated encoding or coefficients. Each document of Amn can be constructed as a linear combination of the basis vectors W1, W2 … Wk, with the corresponding coefficients h11, h21, … hk1 from matrix $H_{kn.}$ Optimal clustering and decomposition rank with lowest root-mean-squared error

between A_{mn} and $W_{mk} \times H_{kn}$ has been achieved by decomposing term-by-document matrix several time. The following diagram shows the residual and corresponding rank.

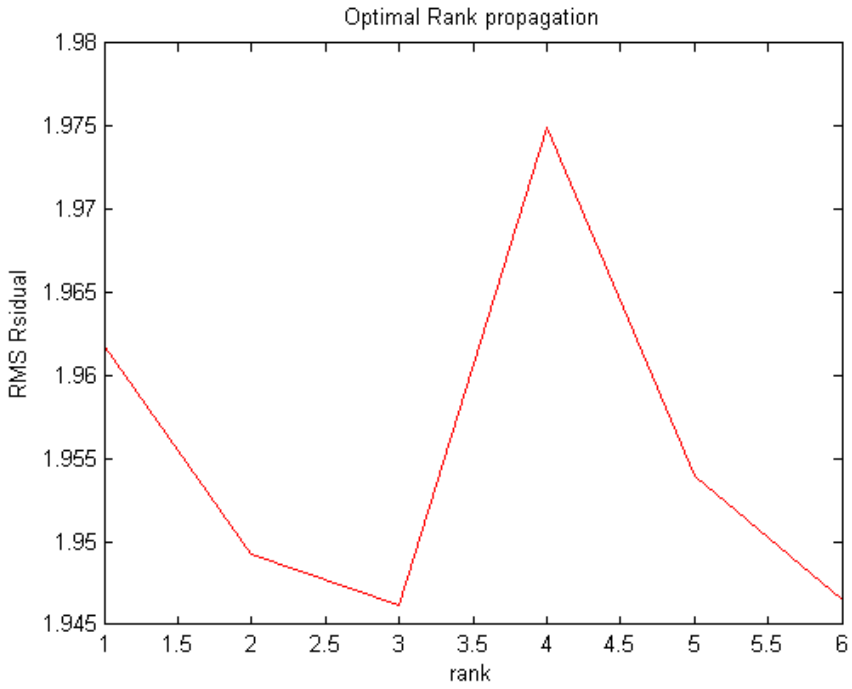


Figure 5: optimal rank propagation.

Now the all the topics and document has been clustered into three groups. We can easily extract topics from those clusters using their weight.

```
h =

         0         0    0.0021    0.0020    0.0038    0.0053    1.0000
    0.9976         0    0.0150    0.0315    0.0547    0.0231         0
         0    0.8508    0.3808    0.2087    0.1860    0.2302         0
```

Figure 6: Encoding matrix with weight for W. The value 1.0 of 1st row shows strong weight for 1st column of W. value 0.9976 shows greater weight for 2nd column of W.

Form the figure6 we can see that the value 1.0 of 1st row shows strong weight for 1st column of W. So document 7 will be grouped into cluster 1 and in same reason document 1 will grouped in cluster 2. Similarly we can find the contribution of each topic for different clusters.

```
    0.0145    0.0001    0.0196
         0    0.0002    0.0334
         0         0    0.0442
         0    0.0007    0.0314
    0.0795    0.0005    0.0298
```

Figure 7: small part of Basis vector W is shown in the figure, where topic 3 contributes only for 3rd clusters.

From the figure7 it can be shown topic 3 contributes only for 3rd clusters and document 2 has highest weight (0.8508) for 3rd cluster. So the document 2 will have considerable number of topic 2. However based on the above clustering of document top three topics has been extracted using "Scikt tools".

Cluster #1

instagram, better ,profile

cluster #2:

podcasts, Listening, FaceBook

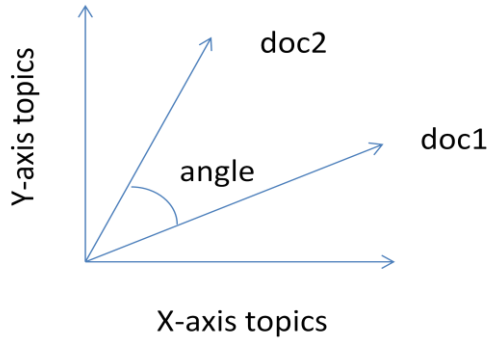
Cluster #3:

image, filters, great

The above output describes the potential word from the reviews of a Apple Color reader App. This captures image of objects and detect color and shares in social media like instagram, Facebook.

3. Cosine Similarity measurement of Documents

As all the entire documents is represented as vector it can be easily measure the Cosine similarity. Similarity of the vectors Doc1 and Doc2 = cosβ= <Doc1, Doc2> / |Doc1||Doc2| As the angle between the vectors, decreases, the cosine angle approaches to 1, meaning that the two document vectors are getting closer, and the similarity of the vectors increases.



			Cosine Similarity for Documents				
doc/doc	doc 1	doc 2	doc 3	doc 4	doc 5	doc 6	doc 7
doc 1	1	0.02179388	0.052516903	0.055800873	0.066658098	0.028557581	0.009706062
doc 2	0.02179388	1	0.197510289	0.154267038	0.141862578	0.067358923	0.005551675
doc 3	0.052516903	0.197510289	1	0.26088177	0.240276246	0.100166435	0.011215252
doc 4	0.055800873	0.154267038	0.26088177	1	0.351618654	0.156804609	0.008185319
doc 5	0.066658098	0.141862578	0.240276246	0.351618654	1	0.169245403	0.015799989
doc 6	0.028557581	0.067358923	0.100166435	0.156804609	0.169245403	1	0.015617881
doc 7	0.009706062	0.005551675	0.011215252	0.008185319	0.015799989	0.015617881	1

Figure 8: shows cosine similarity among document to verify clustering result.

4. Tools and Technology

Java, MatLab, Python, Topia term Extractor, Scikit

Conclusion

User-generated reviews are now an important source of knowledge for consumers and are known to play an active role in decision making, in many domains. In this project, I have described techniques for mining topical and sentiment information from user-generated product reviews which will help both designer and Customer to get quick and summarize information

References:

[1] [Zhu and Zhang, 2010] Feng Zhu and Xiaoquan (Michael)Zhang. Impact of online consumer reviews on sales: The moderating role of product and consumer characteristics. Journal of Marketing, 74(2):133–148, 2010.

[2] [Hu et al., 2008] Nan Hu, Ling Liu, and Jie Zhang. Do online reviews affect product sales? the role of reviewer characteristics and temporal effects. Information Technology and Management, 9:201–214, 2008. 10.1007/s10799-008-0041-2.

[3] [Hu and Liu, 2004b] Minqing Hu and Bing Liu. Mining opinion features in customer reviews. Science, 4:755 760,2004.

[4] Tropp, J., An Alternating minimization algorithm for non-negative matrix approximation.

[5] Evans, B., Non Negative Matrix Factorization, Multidimensional Digital Signal Processing.59 http://www.ece.utexas.edu/~bevans/courses/ee381k/projects/spring03/, (Date:18/ 01/06).

[6] Lee, D., Seung, H., Learning the Parts of Objects by Non-negative matrix Factorization