Summarization of Product Reviews

Aashi Manglik
13006

Neeraj Kumar
13427

Under Guidance of Prof. Harish Karnick

Abstract

We propose an unsupervised approach to generate concise summary of the product from its reviews. To generate summary, we have extracted the features of product and the opinions on each feature using double propagation algorithm. This is followed by popularity analysis to identify the pros and cons (if any) of the product which will form the summary. By summary, here we mean a set of non-redundant phrases, in particular, bigrams that represent the key opinions in review. This is useful when two products have a same sentiment score, say 3 out of 5, and the customer is interested in knowing the major opinion about popular features of each but do not have the time to read several user reviews.

Contents

1 Introduction

Generating textual opinion summaries is a hard task primarily due to two reasons. First, the summaries have to be representative of the key opinions to avoid bias. Second, it should be easily understood by the reader. For the summarisation of product reviews, it needs to be find out what features have been appreciated or criticised the most. Extracting the opinion words that mostly occur with a particular feature will give an insight as what are the general remarks of the population about a particular feature of a product.

2 Motivation

Summarization of opinions is crucial in helping users digest the many opinions expressed on the web. With the increased use of hand-held devices for various on-line activities such as shopping and finding places to eat, the conciseness or compactness of such summaries is also crucial. Consider shopping sites where there could be hundreds of reviews per product. In this case, a concise pros and cons summary consisting of a list of short opinion phrases would help convey critical information about a product, thus facilitating decision making. Such concise summaries are also suitable as tweets that are automatically generated based on blogs or news articles.

3 Related Work

3.1 Feature and Opinion Extraction

Guang Qiu et al [1] proposed a novel method called the Double-Propagation based on bootstrap aggregation to extract features and opinion words modifying them using several syntactic relations that link opinion words and features. Their method is semi-supervised due to the use of opinion word seeds. The basic idea is to extract feature and opinion words iteratively using known and extracted (in previous iterations) words through the identification of syntactic relations.

3.2 Summarisation of Reviews

Ganesan et al [2] have generated non-redundant phrases of two to five words each as summary by formulating summarization as an optimisation problem. Their objective function considers both representativeness (major opinion in text) and readability to ensure that the summaries reflect opinions from the original text and are also reasonably well-formed. The compactness is captured by setting a threshold on the maximum length of summary and a threshold on the similarity between any two phrases in the summary to minimize redundancy. They started with a set of high frequency unigrams from the original text and then merge them to generate higher order n-grams until their readability and representativeness scores remain considerably high.

4 Methodology

We have extracted the key-opinions from the reviews using the concept of opinion mining. The method which we have implemented is un-supervised. In opinion mining, we extract opinion lexicon and opinion targets. Opinion lexicon are those words which indicate positive or negative sentiments and opinion target words are topics on which opinions are expressed. For e.g. in *"tasty chocolate"*, *tasty* is an opinion about *chocolate*. So, *tasty* is an opinion word and *chocolate* is opinion target.

4.1 Relational Identification

There exist a syntactic relation between opinion words and targets because opinion words are used to modify targets.These relations can be seen via dependency parser based on dependency grammar. And, these dependencies can be used to extract new opinion words and targets. We have extracted new opinion words using existing opinion words and targets and similarly extracted new target words. Because of this, it is also called double propagation algorithm.

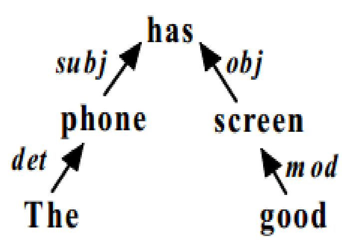

Figure 1: The dependency tree for the sentence *"The phone has good screen"*

Relation between opinion word and targets are of three types

- `OO-Rel:` Relation between two opinion words, example, conjunction
- `OT-Rel:` Relation between opinion word and target word
- `TT-Rel:`Relation between two target words

4.2 Syntactic Relation

Two words A and B are directly or indirectly dependent on each other through syntactic relations. There are two kind of dependency- Direct and Indirect dependency

- `Direct Dependency(DD):` Indicates that one word depends on the other word without any additional words in their dependency path (i.e., directly) or they both depend on a third word directly
- `Indirect Dependency(IDD):`An indirect dependency indicates that one word depends on the other word through some additional words (i.e., indirectly) or they both depend on a third word through additional words.

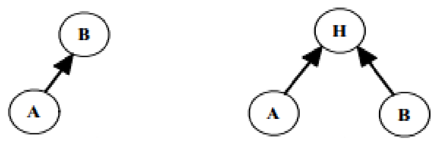

Figure 2: Direct Dependency

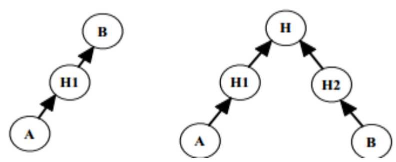

Figure 3: Indirect Dependency

4.3 Propagation Rules

Direct Dependency and Indirect Dependency describes the set of all the relations. So, we have put some more constraints like Part Of Speech(POS) of words and potential syntactic relations between them. We have used NLTK POS Tageer for finding the POS of words and standford dependency parser to find the syntactic relation between words. We also saw that potential opinion words are adjectives(JJ) and target words are nouns(NN). So, we have extracted those words which are nouns and adjectives. We also saw that dependency relations (MR) between opinion words and targets includes- *amod* (adjectival modifier),*nsubj* (nominal subject), *dobj* (direct object) and that for opinion words includes- *conj:and* (CONJ). So, a quadruple describe OT, OO, TT relation - (POS(w_i),DT, R, POS(w_j)) where DT is the dependency type (i.e., DD or IDD) and R is the syntactic relation.

RuleID	Observations	output	Examples
$R1_1$	$O \rightarrow$*O-Dep*$\rightarrow T$ **s.t.** $O \in \{O\}$,*O-Dep*$\in \{MR\}, POS(T) \in \{NN\}$	$t = T$	The phone has a good "screen". ($good \rightarrow$ ***mod*** $\rightarrow screen$)
$R1_2$	$O \rightarrow$*O-Dep*$\rightarrow H \leftarrow$*T-Dep*$\leftarrow T$ **s.t.** $O \in \{O\}$,*O/T-Dep*$\in \{MR\}, POS(T) \in \{NN\}$	$t = T$	"iPod" is the best mp3 player. ($best \rightarrow$ ***mod*** $\rightarrow player \leftarrow$ ***subj*** $\leftarrow iPod$)
$R2_1$	$O \rightarrow$*O-Dep*$\rightarrow T$ **s.t.** $T \in \{T\}$,*O-Dep*$\in \{MR\}, POS(O) \in \{JJ\}$	$o = O$	same as $R1_1$ with screen as the known word and good as the extracted word
$R2_2$	$O \rightarrow$*O-Dep*$\rightarrow H \leftarrow$*T-Dep*$\leftarrow T$ **s.t.** $T \in \{T\}$,*O/T-Dep*$\in \{MR\}, POS(O) \in \{JJ\}$	$o = O$	same as $R1_2$ with iPod as the known word and best as the extract word
$R3_1$	$T_{i(j)} \rightarrow T_{i(j)}$-*Dep* $\rightarrow T_{j(i)}$ **s.t.** $T_{j(i)} \in \{T\}$,$T_{i(j)}$-*Dep*$\in \{CONJ\}, POS(T_{i(j)}) \in \{NN\}$	$t = T_{i(j)}$	Does the player play dvd with audio and "video"? ($video \rightarrow$ ***conj*** $\rightarrow audio$)
$R3_2$	$T_i \rightarrow T_i$-*Dep*$\rightarrow H \leftarrow T_j$-*Dep*$\leftarrow T_j$ **s.t.** $T_i \in \{T\}$,T_i-*Dep*==T_j-*Dep*, $POS(T_j) \in \{NN\}$	$t = T_j$	Canon "G3" has a great len. ($len \rightarrow$ ***obj*** $\rightarrow has \leftarrow$ ***subj*** $\leftarrow G3$)
$R4_1$	$O_{i(j)} \rightarrow O_{i(j)}$-*Dep*$\rightarrow O_{j(i)}$ **s.t.** $O_{j(i)} \in \{O\}$,$O_{i(j)}$-*Dep*$\in \{CONJ\}, POS(O_{i(j)}) \in \{JJ\}$	$o = O_{i(j)}$	The camera is amazing and "easy" to use. ($easy \rightarrow$ ***conj*** $\rightarrow amazing$)
$R4_2$	$O_i \rightarrow O_i$-*Dep*$\rightarrow H \leftarrow O_j$-*Dep*$\leftarrow O_j$ **s.t.** $O_i \in \{O\}$,O_i-*Dep*==O_j-*Dep*, $POS(O_j) \in \{JJ\}$	$o = O_j$	If you want to buy a sexy, "cool", accessory-available mp3 player, you can choose iPod. ($sexy \rightarrow$ ***mod*** $\rightarrow player \leftarrow$ ***mod*** $\leftarrow cool$)

Figure 4: Rules for target and opinion word extraction. In each example, the underlined word is the known word and the word with double quotes is the extracted word.

4.4 The Propagation Algorithm

We initialised the opinion words and targets on the basis of frequency. For this, we POS tagged the words obtained from the review of a particular product and selected nouns(NN) and adjectives(JJ) from them. We then assigned the most frequent noun as product. For example, most frequent noun in a coffee review is *'coffee'*. The next 10 frequent nouns are selected as feature/target seeds. Similarly, adjectives are sorted according to count and top 10 are selected as opinion words. The dependency tree of each sentence is then checked for the rules given in Figure 4 and opinion and feature lists are extended. It stops when no more new opinion words or targets can be added. The propagation algorithm is explained in Figure 5.

4.5 Generation of Micro-opinions using opinion words and features

In our naive approach, we have recorded the counts of each pair of feature and opinion occurring together in the review. The first bigram conveys the most popular opinion associated with the most talked-about feature.

Most of the products have a unique selling point and we have tried to incorporate that in the summary. For this, we computed the probabilities of occurrence of each word w_i, $P(w_i)$ from the corpus created by all the text reviews and summaries from Amazon-fine-food-reviews dataset. To extract a unique opinion about the product, we extracted the opinion word which has a small

Input: Opinion Word Dictionary {O}, Review Data R
Output: All Possible Features {F}, The Expanded Opinion Lexicon {O-Expanded}
Function:

1. {O-Expanded} = {O}
2. $\{F_i\} = \emptyset$, $\{O_i\} = \emptyset$
3. for each parsed sentence in R
4.   if(Extracted features not in {F})
5.   Extract features $\{F_i\}$ using $R1_1$ and $R1_2$ based on opinion words in {O-Expanded}
6.   endif
7.   if(Extracted opinion words not in {O-Expanded})
8.   Extract new opinion words $\{O_i\}$ using $R4_1$ and $R4_2$ based on opinion words in {O-Expanded}
9.   endif
10. endfor
11. Set $\{F\} = \{F\} + \{F_i\}$, {O-Expanded} = {O-Expanded} + $\{O_i\}$
12. for each parsed sentence in R
13.   if(Extracted features not in {F})
14.   Extract features {F'} using $R3_1$ and $R3_2$ based on features in $\{F_i\}$
15.   endif
16.   if(Extracted opinion words not in {O-Expanded})
17.   Extract opinion words {O'} using $R2_1$ and $R2_2$ based on features in $\{F_i\}$
18.   endif
19. end for
20. Set $\{F_i\} = \{F_i\} + \{F'\}$, $\{O_i\} = \{O_i\} + \{O'\}$
21. Set $\{F\} = \{F\} + \{F'\}$, {O-Expanded} = {O-Expanded} + {O'}
22. Repeat 2 till size($\{F_i\}$) = 0, size($\{O_i\}$) = 0

Figure 5: The Propogation Algorithm

probability in the corpus and high frequency in the opinion list of a particular product.

5 Results

5.1 Amazon Fine-Food Reviews Dataset

The products which have been reviewed by around 40 different users were selected. All the reviews of a particular product were seen to summarise it.

5.1.1 Feature Popularity Analysis and Summary

Product	Popular features and their opinions	Summary
Coffee Machine	1. espresso - good, cheap, favorite 2. capsule - good, biodegradable, green 3. flavor - good, excellent, better 4. machine - excellent, great, expensive	good espresso, biodegradable capsule
Toffee Candies	1. case - licorice, good, similar 2. caramel - great, soft, luscious 3. texture - delicious, hard, grainy 4. candies - licorice, wonderful, fresh	good case, luscious caramel
Bread mix	1. bread - gluten-free, tasty, cheese 2. dough - sticky, quick, easy 3. buns - easy, sticky, good 4. recipe - different, good, nice	gluten-free bread, stick dough
Cocoa Powder	1. taste - waxy, bland, new 2. version - bland, new, cheap 3. cocoa - stronger, original, hard 4. texture - dry, regular, smooth	bland taste, dry texture

5.2 Evaluation

PRODUCT7.TXT	our algo	Average_R:0.33333	Average_P:0.75000	Average_F:0.46154
	opinosis	Average_R:0.37451	Average_P:0.67548	Average_F:0.4819
PRODUCT10.TXT	our algo	Average_R:0.40000	Average_P:1.00000	Average_F:0.57143
	opinosis	Average_R:0.39457	Average_P:0.84325	Average_F:0.5376
PRODUCT11.TXT	our algo	Average_R:0.27273	Average_P:0.75000	Average_F:0.40000
	opinosis	Average_R:0.32415	Average_P:0.72145	Average_F:0.4473
PRODUCT12.TXT	our algo	Average_R:0.33333	Average_P:0.75000	Average_F:0.46154
	opinosis	Average_R:0.24689	Average_P:0.79545	Average_F:0.3768
PRODUCT13.TXT	our algo	Average_R:0.33333	Average_P:0.50000	Average_F:0.40000
	opinosis	Average_R:0.32451	Average_P:0.47124	Average_F:0.3843

Figure 6: Comparing our result with *opinosis*

Recall-Oriented Understanding Understudy for Gisting Evaluation, ROUGE is a software package used for evaluating automatic summarization. ROUGE is recall based metric and similar to BLEU(used in machine translation) which is a precision-based measure. It compares the reference summary with automatic generated summary.

ROUGE includes many methods:

- `ROUGE-N:` N-gram based co-occurance statistics
- `ROUGE-L:` LCS based statistics

- `ROUGE-S:` Skip bigram based co-occurance statistics

We have used ROUGE-1 for evaluating our results, shown in Figure 6.

6 Conclusion

This work focuses on two important tasks in opinion mining, namely, opinion and target extraction and summarising the review in form of readable phrases. The results obtained on sample reviews using ROUGE shows that the generated summary in form of bigrams conveys the major opinion. The bigrams can be merged into larger n-grams using conjunction. Since n-gram language models give zero probabilities when corpus is not sufficient enough to contain all possible n-grams, we faced difficulty in moving beyond bigrams. We aim to use smoothing techniques to create longer phrases using n-gram models.

References

[1] Guang Qui et al. Opinion word expansion and target extraction through double propagation. *Computational linguistics 37.1 (2011): 9-27*, pages 9–27.

[2] Ganesan, Zhai, and Viegas. Micropinion generation: An unsupervised approach to generating ultra-concise summaries of opinions. *WWW 2012 Session: Information Extraction*, April 16-20, 2012, Lyon, France.