Product Feature Discovery and Ranking for Sentiment Analysis from Online Reviews.

SHASHWAT CHANDRA

NITISH GUPTA

ADVISOR: AMITABHA MUKERJEE

Motivation

- Important task of review mining is to extract people's opinions and sentiments on features of products.

  Eg. "The phone has a good battery life" shows a positive sentiment on the feature "battery life" of the phone.

- In an unsupervised environment extracting the 'features' of a product class is the most important and difficult task when mining online reviews.

- Feature Ranking and Sentiment Analysis is important for obvious reasons of getting to know in an automated manner what features of a product do the users keep in mind and which features matter the most. Also it gives an idea about the product and also which features in a product are good or bad.

Introduction

- Recent previous work on feature extraction and ranking of features products deals primarily with *Double Propogation*[1], a state-of-the-art algorithm based on bootstrap aggregation and used for finding new product features.

- Previous work on detecting the subject of reviews worked with *part-whole* relationships[2].

- Sentiment Analysis deals with recognizing positive/negative opinions on a target feature of a product. Unsupervised sentiment analysis[3] uses two-word phrases with compatible POS tags. Semi-supervised sentiment analysis[4] uses clustering or grouping of synonym opinion words.

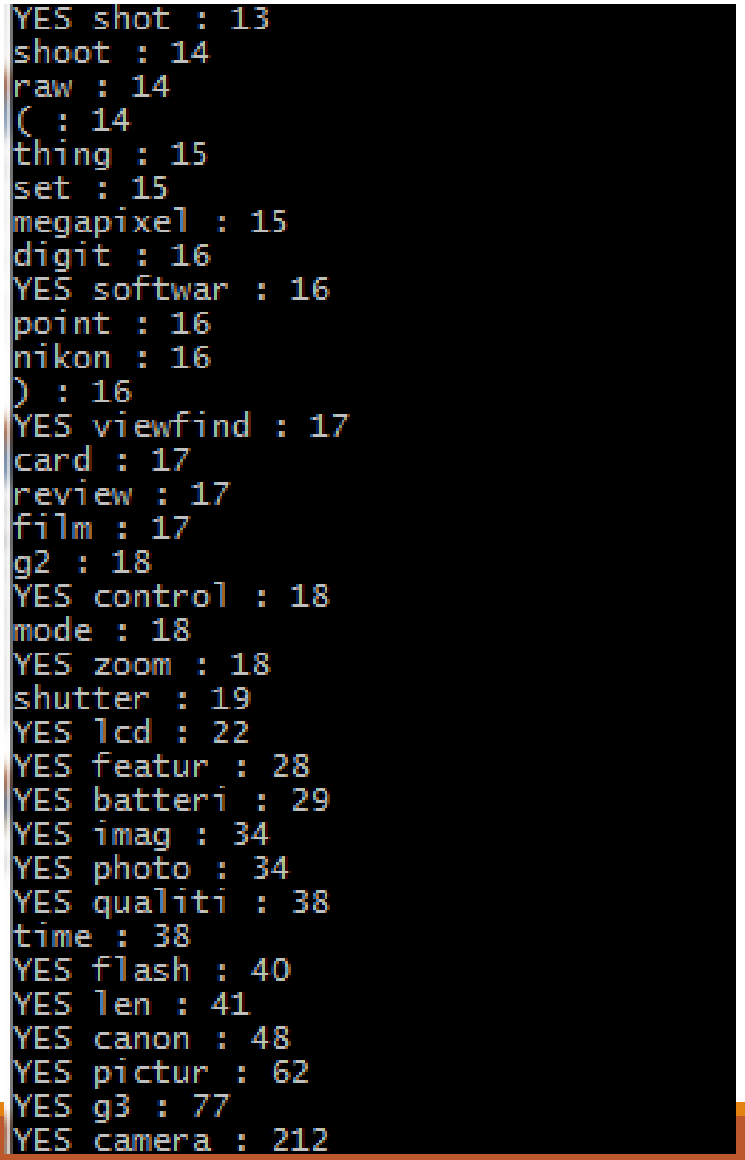
- One approach used for feature ranking[2] deals with association-rule mining.

Methodology

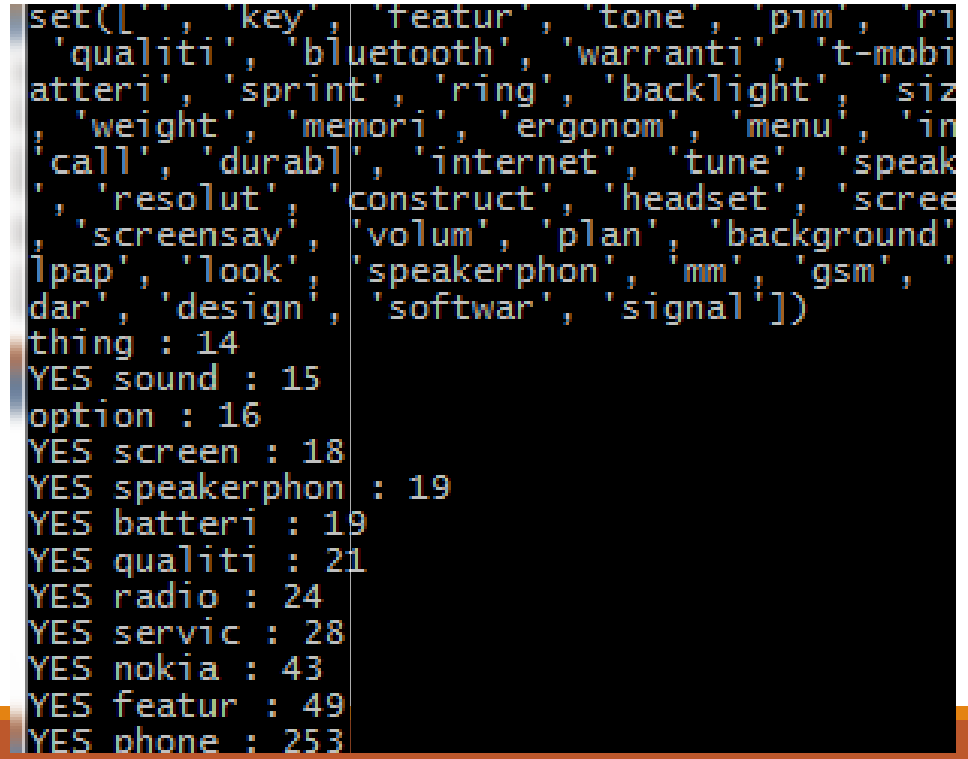
Our Approach to discovering features :

- We are considering that the features of a product nouns or noun phrases. Eg engine, screen, battery life, camera etc.
- We are trying a very naïve approach first where we extract all nouns in the reviews and lemmatize them. Calculate the frequency of their occurrence and arrange it in descending order.
- Most of the features are contained in the top frequencies, upto nouns/noun phrases that have frequency above ‘Mean + Standard Deviation’.
- As we have already tagged dataset with the features marked, we compute the precision and recall to show the effectiveness of this naïve approach.

Methodology

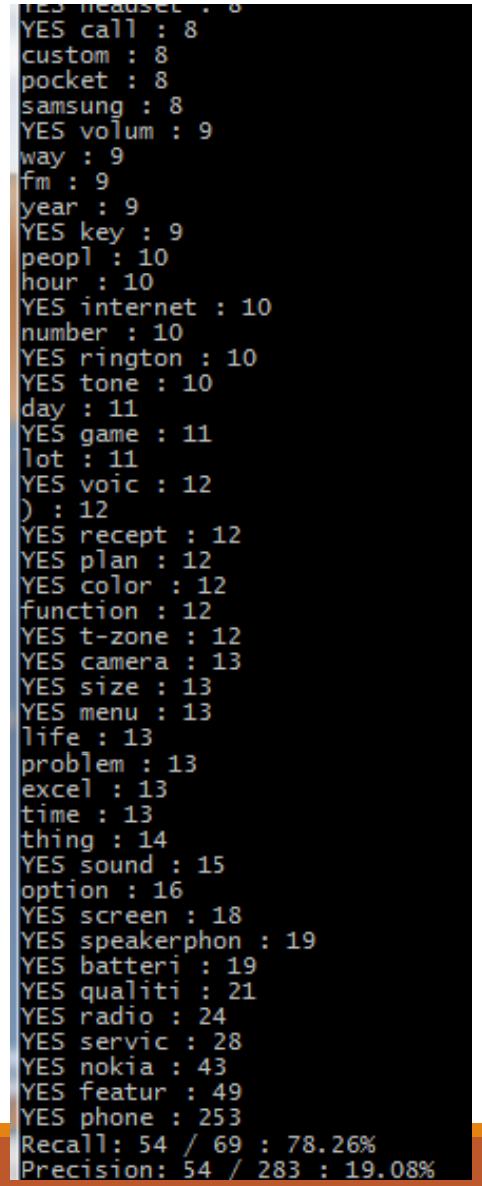


DATASET: CANON G3 Camera
Precision: 48.57%
Recall: 26.15%



DATASET: Nokia 6610
Precision: 83.33%
Recall: 14.49%

Methodology



Using Mean-Std
DATASET: Nokia 6610
Precision: 9.59%
Recall: 95.65%

Using Mean
DATASET: Nokia 6610
Precision: 19.08%
Recall: 78.26%

Using Mean+Std
DATASET: Nokia 6610
Precision: 83.33%
Recall: 14.49%

The Naïve approach is useful in detecting the product, since the most frequent noun was always the correctly deduced product name.

Product	Deduced product
Nikon Coolpix 4300 (Camera)	Camera
Nokia 6610 (Phone)	Phone
Canon G3 (Camera)	Camera
Apex AD2600 Progressive-scan (DVD player)	DVD (, Player)
Creative Labs Nomad Jukebox Zen Xtra 40GB (MP3 Player)	Player (, ipod)

Methodology

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 lens. (*lens* $\rightarrow$ *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*)

Double-Propagation Approach to finding features :

- The double propagation algorithm uses the dependency of nouns/noun phrases(possible features) and adjectives(possible opinion words) on each other and propagates through the corpus looking for new features and opinion words.

Feature Ranking

- Feature Ranking is done by comparing the frequency of different features as discovered, the frequency of opinion words, along the with frequency of the opinion words that are used to modify the features.

- This is based on the famous web-page ranking algorithm, HITS. It is assumed that there exists a mutual reinforcement relationship between the features and the opinion words i.e.
  - The opinion words used to modify important features are themselves important
  - The features that are modified by important opinion words are themselves important.

- This is an iterative process and at the end we expect to get important features.

Sentiment Analysis

- We plan to do sentiment analysis on the online reviews using the features and the opinion words we mine. This would include computing the polarity and strength of opinion that the user has on a particular feature of the product. This would also give an overall sentiment of the user on the product as a whole.

- Reinforcement Learning: A naïve form of sentiment analysis we performed on the data looked at the similarity of the opinion word to known positive/negative opinion words.
  - The similarity metric used was the shortest path connecting word senses.

- A modification of this naïve approach can be performed on all opinion words using a modified version of double-propogation, to give two classes of similar opinion words.

References

[1] Qui, Guang, et al. “Opinion Word Expansion and Target Extraction through Double Propogation” Association for Computational Linguistics, 2011

[2] Zhang, Lei, et al. “Extracting and Ranking Product Features in Opinion Documents.” Proceedings of the 23rd International Conference on Computational Linguistics: Posters. Association for Computational Linguistics, 2010.

[3] Liu, Bing. “Sentiment analysis and opinion mining.” Synthesis Lectures on Human Language Technologies 5.1 (2012): 1-167.

[4] Zhai, Zhongwu, et al. “Clustering product features for opinion mining.” Proceedings of the fourth ACM international conference on Web search and data mining. ACM, 2011.

Thank You!!

Questions