IRE Project 9

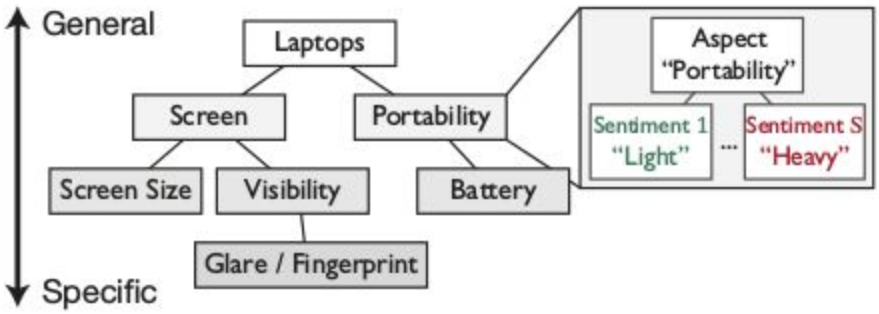
Customer Review Analytics

Problem Statement

Understand the sentiment of user reviews and provide useful information for the end-user as well as the product manufacturer regarding public opinion of the product.

Introduction

Online reviews contain rich information on different aspects of a product and the sentiment polarities of users. For example, in laptop reviews, there are comments on aspects such as the overall design, battery, screen, and CPU. Before making purchases, consumers often seek opinions from other users by reading their reviews. The key information a consumer wants to get from the reviews is: whether the product is good, and what aspects received positive or negative opinions. This task is quite challenging because it is difficult for a human being to extract statistical aspect- sentiment information from a massive set of online reviews.



We can understand the necessity of a hierarchical structure in sentiment analysis from the following two viewpoints, consumer and technology.

From the viewpoint of consumers, different users need different information and hence are interested in different granularities of aspects and sentiments. For example, some consumers care about the opinions of general aspects such as screens and CPUs, while others may pay more attention to more specific aspects such as CPU frequency and cache size. Analysis of aspects and sentiments at some single granularity cannot satisfy all the users. Therefore it is desirable to convey a hierarchy of aspects and sentiments to users so they can easily navigate to the desired granularity. Additionally, a well-organized hierarchy of aspects and sentiments from the general to the specific is easy to understand by human being.

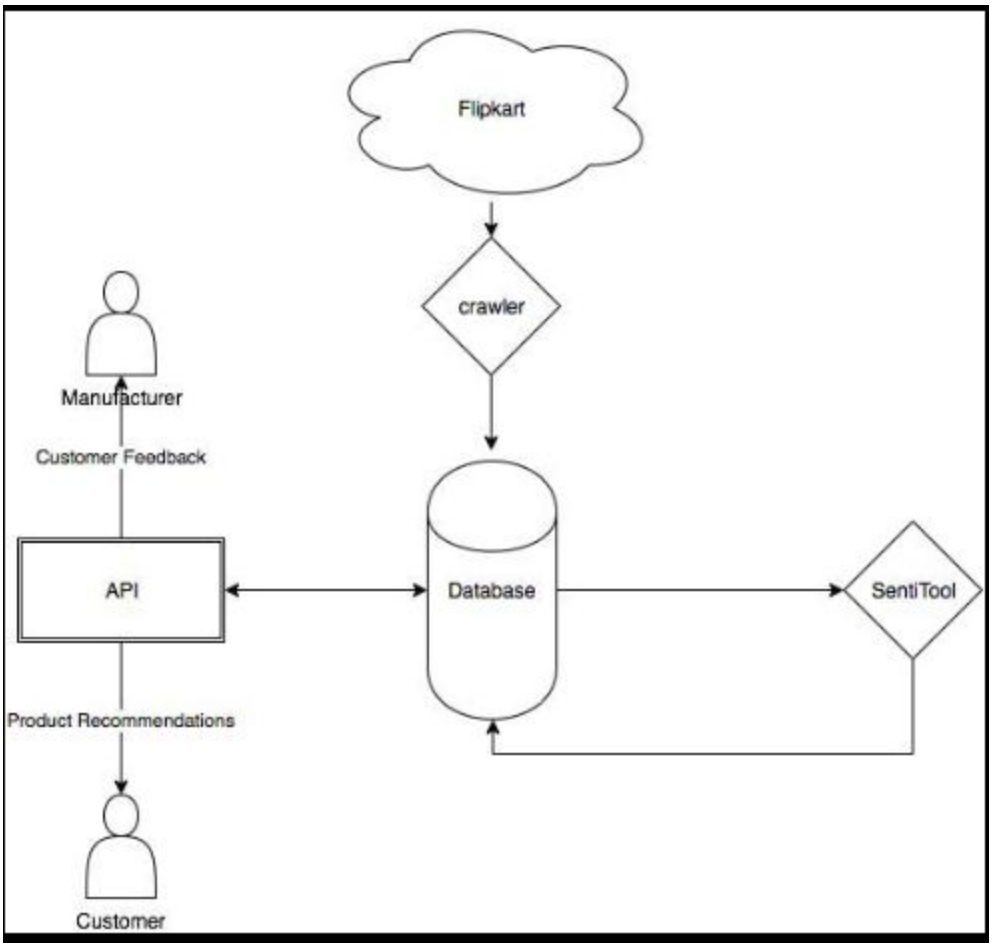
From the viewpoint of technology, the tree structure helps sentiment analysis. Well-identified sentiment words contribute much to the accuracy of sentiment analysis. In real world applications, the polarities of many words depend on the aspect and the differentiation of prior and contextual polarity is crucial. For example, the word “fast” is positive when used to describe a CPU, but it would be negative when describing a battery. This problem is a great challenge for aspect-based sentiment analysis, especially for unsupervised models. Commonly, we provide only general sentiment seed words such as “good” and “bad” which of of little help in identifying aspect-specific sentiment expressions.

Existing unsupervised models try to propagate the polarity of these general words to the aspect-specific sentiment words by their co-occurrences within some context, but it is difficult to do so from the most generic sentiment words to the fine-grain aspect-specific words since the

co-occurrences can be quite sparse. Instead, by discovering the hierarchical structure, we can propagate the polarities along the hierarchy from the general to the specific so that most aspect-specific sentiment words can be identified.

In this project, we aim to extract an aspect-sentiment hierarchy from reviews. It provides users an overall evaluation of the products, aspect-based sentiments, sentiment- based aspects, and it also supports navigation of aspects and the associated sentiments over the tree at different granularities. However, it is not easy to jointly model aspect hierarchy and sentiments because it needs to learn both the hierarchical structure and the aspect-sentiment topics from an unlabeled corpus.

Approach



Our approach is designed to be as unsupervised and knowledge-lean as possible, so as to make it transferable across different types of products and services, as well as across languages.

We go for unsupervised methods because of two main reasons :

- Due to the wide range and variety of products and services being reviewed, the framework must be robust and easily transferable between domains.

- The second reason is the nature of the data. Online reviews are often short and unstructured, and may contain many spelling and grammatical errors, as well as slang or specialized jargon.

These factors often present a problem to methods relying exclusively on dictionaries, manually constructed knowledge resources, and gazetteers, as they may miss out on an important aspect of the product or an indicator of sentiment. Unsupervised methods, on the other hand, are not influenced by the lexical form, and can handle unknown words or word-forms, provided they occur frequently enough. This ensures that any emergent topic that is salient in the data will be addressed by the system.

The project can be divided into three major tasks namely data extraction and processing, aspect detection and sentiment analysis.

Data Extraction involves collecting data (user reviews and other meta-data) from popular ecommerce websites.

Processing step converts unstructured data (raw html) into a structured format (relational tables) which can be used by our tool to determine the various aspects and their corresponding sentiments for each product.

Aspects are determined via a local version of LDA, which operates on sentences, rather than documents, and employs a small number of topics that correspond directly to aspects. This approach overcomes the problems of frequent-term methods, as well as the issues raised by Titov and McDonald (2008b).

In **Sentiment analysis,** we use morphological negation indicators to automatically create a seed set of highly relevant positive and negative adjectives, which are guaranteed to be pertinent to the aspect at hand. These automatically-derived seed sets

achieve comparable results to the use of manual ones, and the work of Zagibalov and Carroll (2008) suggests that the use of negation can be easily transferred to other languages.

Data Extraction and Processing

For the purposes of this project, user reviews were collected from e-commerce websites: flipkart and amazon.

Tools used :
Scrapy : contains mechanisms for crawling and scraping
www.scarpy.org
Selenium : contains mechanisms to render javascript and ajax enabled web pages
http://www.seleniumhq.org/docs/

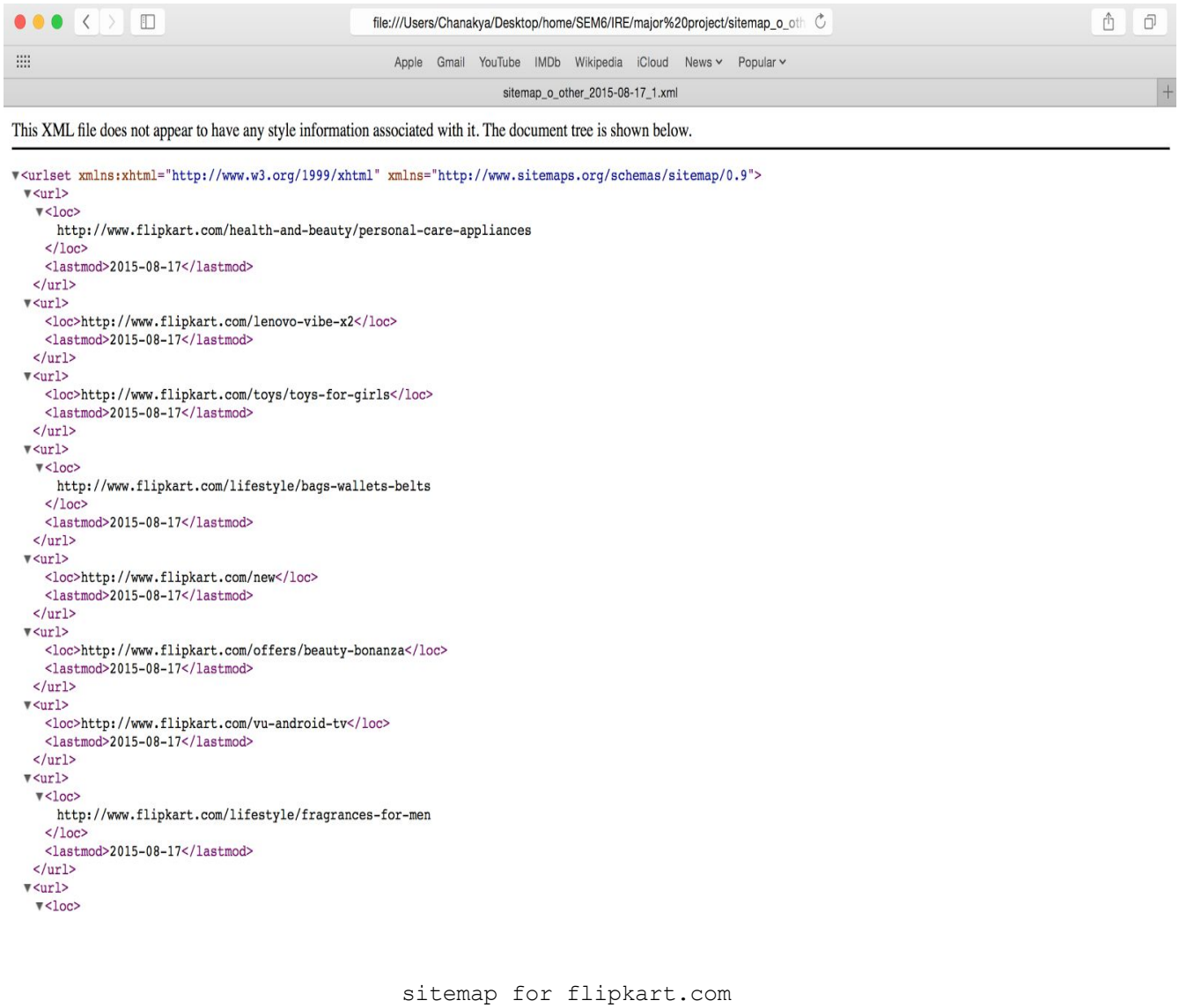
The crawling process was divided into 2 steps:

1. Collect a complete list of products under various categories (electronics, clothing, home appliances etc)
2. Collect all the user reviews for each product.

The data was stored in a relational database to allow for easy access in the future.

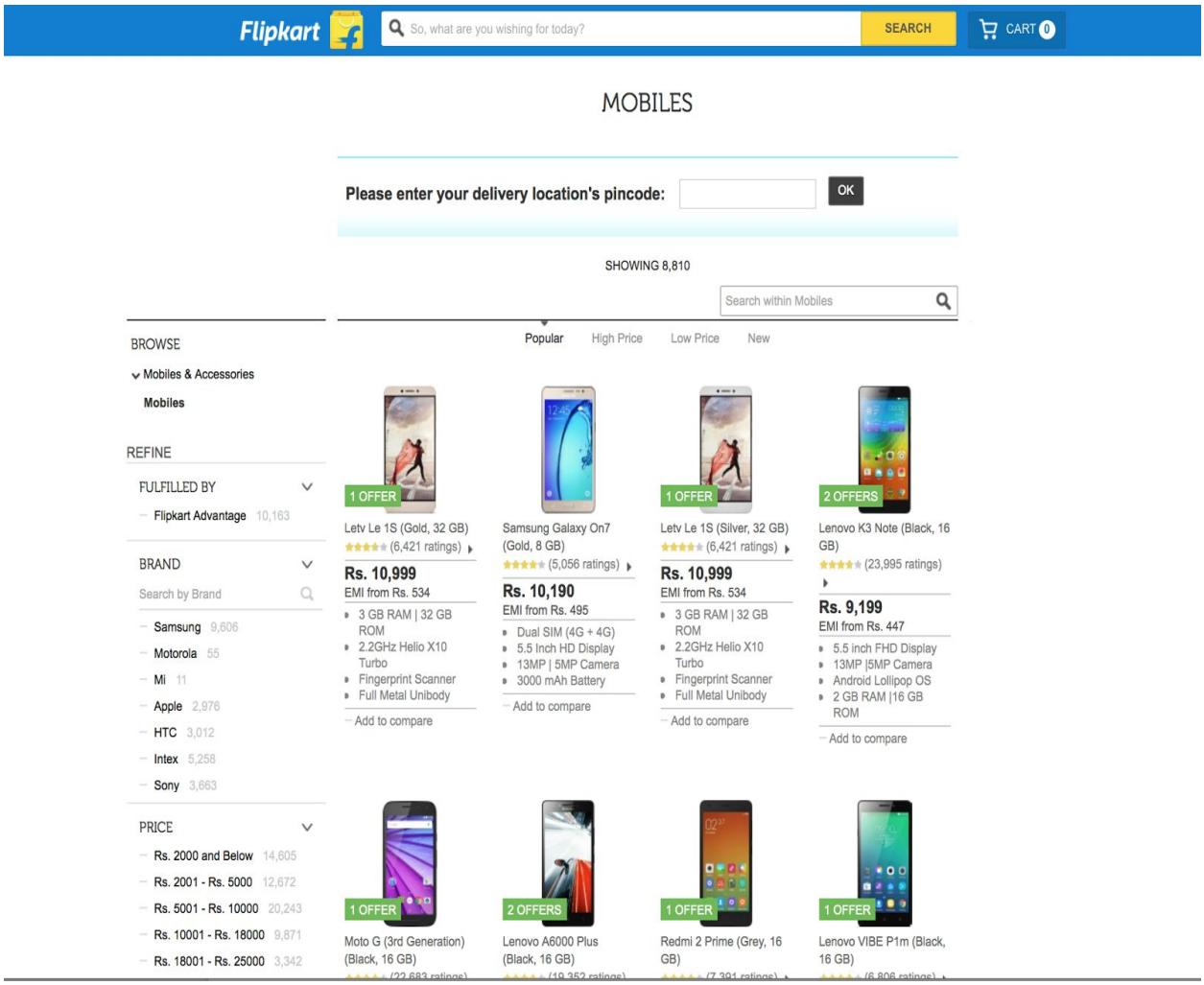
The following section explains the process of crawling in detail using the example of flipkart:

1. http://www.flipkart.com/robots.txt was used to obtain the sitemap. The various urls provided in the sitemap were used to create the seed set to start the crawl.



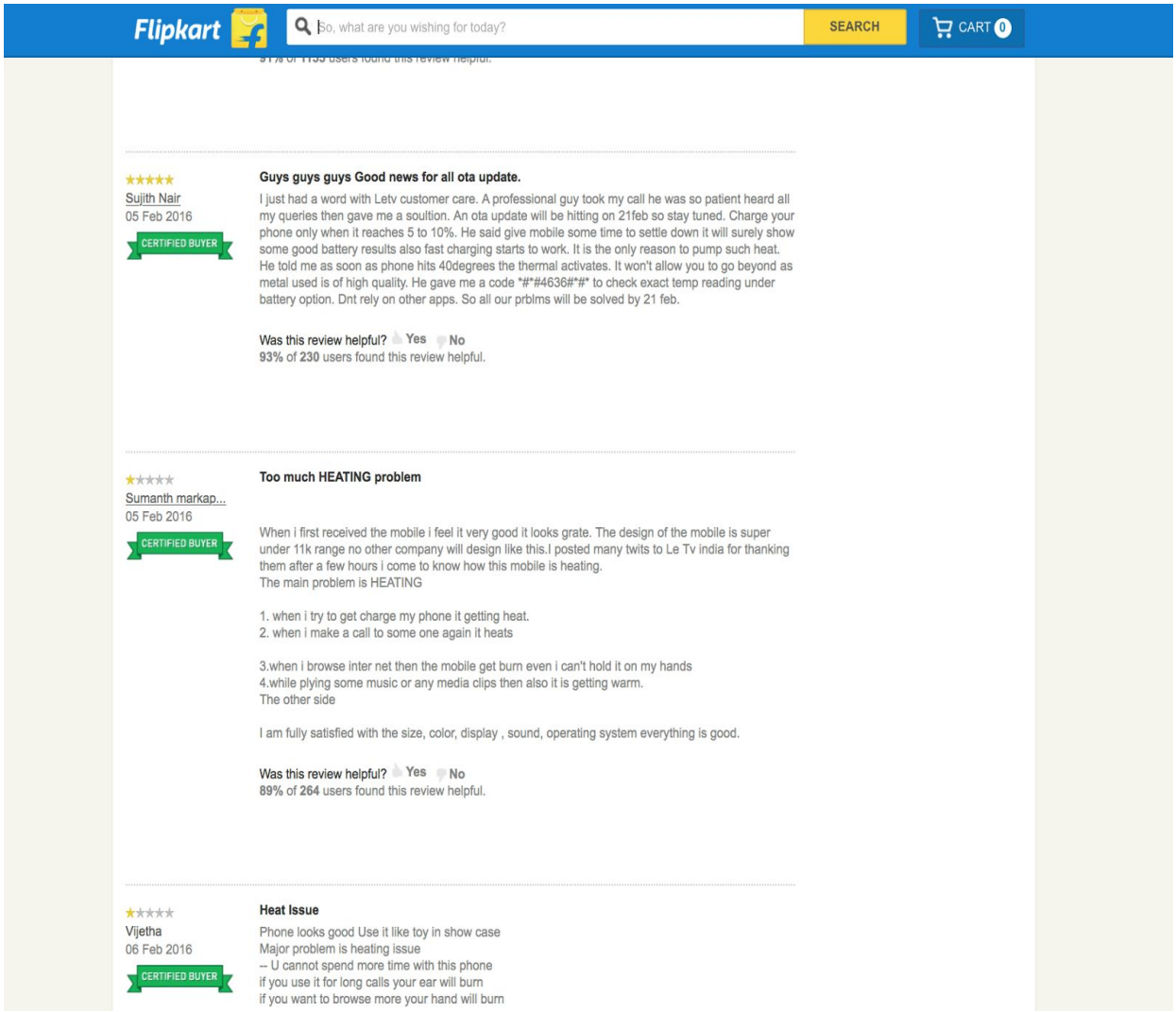
sitemap for flipkart.com

2. Each of the seed urls provide a list of all products for a particular category.



All products in the mobile category

3. The urls for each product were then followed to obtain the complete list of user reviews.



All the user reviews for *Le TV 1s* phone

4. The crawled data was stored in a relational database. We decided to use PostgreSQL to store our data.
    - It is an industry standard capable of storing large volumes of data while still maintaining the speed of retrieval.
    - The SentiTool will use the reviews (along with meta data such as user rating of the product, reliability of the user, rating of the review itself etc) to provide sentiment to the various aspects of the product being discussed.

Aspect Detection

To identify the various aspects mentioned in the reviews. Understand the usage of topic modelling tools (LDA) for the project.

Stanford LDA tool to be used - http://nlp.stanford.edu/software/tmt/tmt-0.4/

- It's implemented in Scala (runs on JVM) and very easy to customize (good integration with other NLP tools to pre- and post-process documents -- such as removing frequent or infrequent tokens). Also, it has implemented other complex LDA enhancement models as well.
- It provides both Gibbs sampling and variational Bayes approximation. The latter option converges much faster, but consumes more memory. It's running multi-threaded at least for the latter option.
- It's easy to use and has much more detailed tutorial compared to other similar tools.

Cons

- It addresses global topics(entity- type) rather than local ones (attributes).

Solution

- We will treat each sentence as a separate document. The output of the model will be a distribution over the inferred aspects for each sentence in the data.

Input parameters

- Since LDA takes two parameters we will see what fits in as of now the standard input ($\alpha = 0.1$, $\beta = 0.1$, 3000 iterations), with no specific tuning to our data.

- Need to run the algorithm with varying k (number of aspects)and employ a cluster validation scheme to determine the optimal number.

Cluster Validation Procedure

- The issue with determining the correct number of clusters, is an important element in unsupervised learning.
- A common approach (Levine and Domany, 2001; Lange et al., 2004; Niu et al., 2007) is to use a cluster validation procedure. In such a procedure, different model orders are compared, and the one with the most consistent clustering is chosen.
- For the purpose of the validation procedure, we'll have a cluster corresponding to each aspect, and we'll label each sentence as belonging to the cluster of the most probable aspect.
- Given the collection of sentences in our data, D , and two connectivity matrices C and ˆ C , where a cell i , j contains 1 if sentences d i and d j belong to the same cluster, we define a consistency function F (following Niu et al. 2007):

$$F(C,\hat{C}) = \frac{\sum_{i,j} 1\{C_{i,j} = \hat{C}_{i,j} = 1, d_i, d_j \in \hat{D}\}}{\sum_{i,j} 1\{C_{i,j} = 1, d_i, d_j \in \hat{D}\}} \quad (1)$$

Methodology

- Run the LDA model with k topics on D to obtain connectivity matrix C_k .
- Create a comparison connectivity matrix R_k based on uniformly drawn random assignments of the instances.
- Sample random subset D_i of size δ | D | from D .

- Run the LDA model on D_i to obtain connectivity matrix C_k .
- Create a comparison matrix R_{ik} based on the uniformly drawn random assignments of the instances in D_i .
- Calculate the score

$$score_i(k) = F(\hat{C}, C) - F(\hat{R}, R)$$

where F is given in Eq. 1.

- Repeat steps 3 to 6 n times.
- Return the average score over n iterations.

This procedure calculates the consistency of our clustering solution, using a similar sized random assignment for comparison. It does this on n subsets to reduce the effects of chance. The k with the highest score is chosen.

Determining Representative Words

- For each aspect, we list all the nouns in the data according to a score based on their mutual information with regard to that aspect.

$$Score_a(w) = p(w,a) \cdot log \frac{p(w,a)}{p(w) \cdot p(a)} \qquad (2)$$

Where p (w) , p (a) , p (w , a) are the probabilities, according to the LDA model, of the word w , the aspect a , and the word w labeled with aspect a , respectively.

- We then select, for each aspect, the top k_a ranking words, such that they cover most of the word instances labeled by the LDA model with aspect $label_a$.

- This set of representative words for each aspect is used in the sentiment component of our system.

Expected output

Aspect	Representative Words	Aspect	Representative Words
Performance	power, performance, mode, fan, quiet	Mouse	mouse, right, touchpad, pad, buttons, left
Hardware	drive, wireless, bluetooth, usb, speakers, webcam	General	great, little, machine, price, netbook, happy
Memory	ram, 2GB, upgrade, extra, 1GB, speed	Purchase	amazon, purchased, bought, weeks, ordered
Software	using, office, software, installed, works, programs	Looks	looks, feel, white, finish, blue, solid, glossy
Usability	internet, video, web, movies, music, email, play	OS	windows, xp, system, boot, linux, vista, os
Portability	around, light, work, portable, weight, travel	Battery	battery, life, hours, time, cell, last
Comparison	netbooks, best, reviews, read, decided, research	Size	screen, keyboard, size, small, enough, big

The above table presents the expected aspects predicted by our system. Some aspects would probably be missed unless the annotators carefully read through all the reviews, e.g., the Memory aspect, which includes advice about upgrading specific models. This capability of our system is important, as it demonstrates that our method can be used to produce customized comparisons for the user and will take into account the important common factors, as well as the unique aspects of each item.

As we can see this system does not require specially designed models or additional information in the form of user-provided aspect-specific ratings. We believe the reason for this stems from the composition of online reviews. Since many reviews have similar mixtures of local topics (e.g., laptop, service), standard LDA prefers global topics, which 3 distinguish more strongly between reviews (e.g., battery type, screen type). However, when employed at the sentence level, local topics (corresponding to rateable aspects) provide a stronger way to distinguish between individual sentences.

References

- An Unsupervised Aspect-Sentiment Model for Online Reviews : http://people.dbmi.columbia.edu/noemie/papers/naacl10.pdf
- ASPECT BASED SENTIMENT ANALYSIS: http://www.aueb.gr/users/ion/docs/pavlopoulos_phd_thesis.pdf
- An Unsupervised Aspect Detection Model for Sentiment Analysis of Reviews: https://pdfs.semanticscholar.org/88af/42a6303c3edfe64b0cbf42e28d2253016e49.pdf

Sentiment Analysis

To obtain metric for features of the products we need to identify the opinion of that feature from various reviews available. We view the goal of reading multiple reviews as finding widely-held opinions and weighing the positive against the negative, and we wish to automate this sort of task using NLP and machine-learning techniques.

Methodology

For each aspect.

1. extract relevant adjectives
2. build a conjunction graph,
3. determine the seed set
4. Propagate polarity scores to the rest of the adjectives.

Extracting adjectives

As a pre-processing step the data was parsed, from the parsed data, identify conjunction and negation.

- Replace all adjectives (say A) which are associated with a negation with “not-A”.
- Identify all adjectives which modify a noun.
- Identify nouns which are relevant to our product (i.e is a desired feature) and the adjective(s) associated with it.
- Eg. “The display was nice and clear, but the battery life was not good.”, in the example we extract the pairs => (display, nice), (display, clear) and (battery, not-good).

Building the Polarity Graph

The approach we will be following to determine sentiment polarity is based on an adaptation of Hatzivassiloglou and McKeown (1997) .
But we have the following issues:

1. In the original article, adjectives with no orientation were ignored.
2. It is unclear how this can be easily done in an unsupervised fashion, and such sentiment-neutral adjectives are found everywhere in real-world data.
3. Adjectives whose orientation depended on the context were also ignored. These are of particular interest in our task, and are likely to be missing or incorrectly labeled in standard sentiment dictionaries
4. We need to handle adjectives expressing various shades of sentiment, not only strongly positive or negative ones, we are interested in a scoring method, rather than a binary labeling.
5. We do not want to use a general corpus,but rather the text from the reviews themselves. This usually means a much smaller corpus than the one used in the original paper, but has the advantage of being domain specific.

Therefore,we will build the polarity graph in some ways different from the original paper.

We will not use disjunctions (e.g., ‘but’) as indicators of opposite polarity
As these sometimes does not imply opposite polarity.

(e.g.”dainty but strong necklace”,”cheap but delicious food”)
So instead of using regular expressions to capture explicit conjunctions,we will extract all the cases where two adjectives modified a single noun in the same sentence.

To handle aspect-specific adjectives correctly, we will build a separate graph for each aspect with each modified noun.

Constructing a Seed Set

Seed Set is the initial Training Set provided to the learning Algorithm in an Active Learning process. The Documents in the Seed Set may be selected based on Random Sampling or Judgmental Sampling. Some commentators use the term more restrictively to refer only to Documents chosen using Judgmental Sampling. Other commentators use the term generally to mean any Training Set, including the final Training Set in Iterative Training, or the only Training Set in non-Iterative Training.

- We plan to use the identification, analysis and description of the structure of a language's linguistic units, such as root words, affixes, parts of speech, intonations and stresses, or implied context, explicit negation to find pairs of opposite polarity.
- The adjective pairs which will be distinguished only by one of the prefixes ‘un’,‘in’, ‘dis’, ‘non’,or by the negation marker ‘not-’ will be selected for the seed set.
- Starting with the most frequent pair, we assign a positive polarity to the more frequent member of the pair.
- Now in order of decreasing frequency, polarity to the other seed pairs will be assigned based on the shortest path either of the members had to a previously labeled adjective. That member will receive its neighbor’s polarity, and the other member of the pair will receive the opposite polarity.

- After we finish labelling all pairs we will run corrective algorithms for misclassifications by some methods like iterating through the pairs and reversing the polarity if that improved consistency, i.e., if it can cause the members of the pair to match the polarities of more of their neighbors.
- Lastly we will reverse the polarity of the seed groups if the negative group has a higher total frequency.

Propagating Polarity

The adjectives in the positive and negative seed groups are assigned a polarity score of 1 and 0, respectively. All the rest start with a score of 0.5. Then, an update step is repeated. In update iteration t, for each adjective x
that is not in the seed, the following update rule is applied.

$$p^t(x) = \frac{\Sigma_{y \in N(x)} w(y,x) \cdot p^{t-1}(y)}{\Sigma_{y \in N(x)} w(y,x)}$$

Where pt(x) is the polarity of adjective x at step t, N(x) is the set of the neighbors of x, and w(y,x) is the weight of the edge connecting x and y. We set this weight to be 1 + log(#mod(y,x)), where #mod(y,x) is the number of times y and x both modified a single noun. The update step is repeated to convergence.

Evaluation

The performance of the sentiment component of our system will be evaluated based on an aspect-specific gold standard. We will identify top eight automatically inferred aspects for each of the products and construct polarity graph for each of the aspects. We will retrieve a list of all adjectives that participated in five or more modifications of

nouns. We plan to split the data into ten portions and, for each portion rate each adjective according to the polarity of the sentiment it expresses in the context of the specified aspect with ratings like Strongly Negative, Weakly Negative, Neutral,Weakly Positive, Strongly Positive, and N/A. Then,we will translate the annotator ratings to a numerical scale, from −2 (Strongly Negative) to +2 (Strongly Positive) at unit intervals. After discarding adjectives where one or more annotators gave a 'N/A' tag, we will average the two annotator numerical scores, and use this data as the gold standard for our evaluation.

References

http://people.dbmi.columbia.edu/noemie/papers/naacl10.pdf

http://nlp.stanford.edu/courses/cs224n/2009/fp/14.pdf

https://www.sccs.swarthmore.edu/users/15/crain1/files/NLP_Final_Project.pdf

Hierarchical Sentiment Model

After calculating the sentiment score of each product we update the scores for products and propagate a fraction of the score to its parents (ancestors).

Methodology

- Ontology tree creation
    - Create a hierarchy of products like the one shown in the above diagram.
- Insert scores at the nodes.
- Do a depth first search through the tree and while back-tracking update the scores of the parents (non - terminal nodes) with the leave score multiplied by a constant factor.

References

- A Hierarchical Aspect-Sentiment Model for Online Reviews : https://www.google.co.in/url?sa=t&rct=j&q&esrc=s&source=web&cd=2&cad=rja&uact=8&ved=0ahUKEwjqrKqT8-PLAhULU44KHRqICE0QFggrMAE&url=https%3A%2F%2Fwww.aaai.org%2Focs%2Findex.php%2FAAAI%2FAAAI13%2Fpaper%2Fdownload%2F6486%2F7202&usg=AFQjCNF4ABbJct2LPVL_eGlXus5Wh5jorQ&sig2=rSeieWcWWudV5nYl8zZz4g&bvm=bv.117868183%2Cd.c2E