RECOMENDR-ENTITY RECOMMENDATION BASED ON AD-HOC DIMENSIONS

BY

PREEYAA RAWLANI

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2011

Urbana, Illinois

Adviser:

Associate Professor ChengXiang Zhai

ABSTRACT

The growing availability and popularity of opinion rich resources on the online web resources, such as review sites and personal blogs, has made it convenient to find out about the opinions and experiences of layman people. But, simultaneously, this huge eruption of data has made it difficult to reach to a conclusion.

In this thesis, I develop a novel recommendation system, Recomendr that can help users digest all the reviews about an entity and compare candidate entities based on ad-hoc dimensions specified by keywords. It expects keyword specified ad-hoc dimensions/features as input from the user and based on those features; it compares the selected range of entities using reviews provided on the related User Generated Contents (UGC) e.g. online reviews. It then rates the textual stream of data using a scoring function and returns the decision based on an aggregate opinion to the user. Evaluation of Recomendr using a data set in the laptop domain shows that it can effectively recommend the best laptop as per user-specified dimensions such as price. Recomendr is a general system that can potentially work for any entities on which online reviews or opinionated text is available.

This is for you, Mummy and Papa and my motivator, my companion, Sanjay

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my advisor, Professor ChengXiang Zhai, for his continuous support and guidance during my research. His energetic, enthusiastic and knowledgeable personality always served as a motivation to me.

I would also like to thank Naoki Tanaka and Naveen Cherukuri for their contributions in web crawling and front end design.

I would also like to thank senior collaborators who guided me in the beginning and all my friends for their constant support. I would specially thank Manish Gupta who served as a mentor to me during the course of this thesis.

Most importantly, I would like to thank my life partner, Sanjay Rajwani, whose presence and support motivated me at every setback and helped me in maintaining my focus towards my aim.

Finally, I am indebted to my parents and my younger brother and sister for their understanding and encouragement. I am able to reach to this position only because of their well-wishes.

TABLE OF CONTENTS

CHAPTER 1. INTRODUCTION

“What other people think” has always been an important piece of information for most of us during the decision-making process. Long before awareness of the World Wide Web became widespread, many of us asked our friends to recommend an auto mechanic or to explain who they were planning to vote for in local elections, requested reference letters regarding job applicants from colleagues, or consulted Consumer Reports to decide what dishwasher to buy. But the Internet and the Web have now (among other things) made it possible to find out about the opinions and experiences of those in the vast pool of people that are neither our personal acquaintances nor well-known professional critics — that is, people we have never heard of. And conversely, more and more people are making their opinions available to strangers via the Internet.

With the growing availability and popularity of opinion-rich resources such as online review sites and personal blogs, new opportunities and challenges arise as people can, and do, actively use information technologies to seek out and understand the opinions of others. The sudden eruption of activity in the area of opinion mining and sentiment analysis, which deals with the computational treatment of opinion, sentiment, and subjectivity in text, has thus occurred at least in part as a direct response to the surge of interest in new systems that deal directly with opinions as a first-class object.

To help people digest and exploit the online opinion information, in this thesis, we propose to develop a novel entity recommender system, called Recomendr. This system accepts ad-hoc dimensions as an input from the user, compares selected entities by rating the reviews using a sophisticated scoring function and returns the aggregate opinion to the user.

1.1 Motivation

According to two surveys of more than 2000 American adults each [12, 13]:

- 81% of Internet users (or 60% of Americans) have done online research on a product at least once;
- consumers report being willing to pay from 20% to 99% more for a 5-star-rated item than a 4-star-rated item (the variance stems from what type of item or service is considered);

- 32% have provided a rating on a product, service, or person via an online ratings system, and 30%(including 18% of online senior citizens) have posted an online comment or review regarding a product or service.

All this illustrates the importance of creating a system that helps in understanding the opinion hidden in the reviews. For example, consider the example of a computer manufacturer's concern, it would be difficult to try to directly survey laptop purchasers who have not bought the company's product. Rather, we could employ a system that

(a) finds reviews or other expressions of opinion on the Web — newsgroups, individual blogs, and aggregation sites such as Epinions are likely to be productive sources — and then

(b) Creates condensed versions of individual reviews or a digest of overall consensus points.

Such a system, if designed, would save an analyst from having to read potentially dozens or even hundreds of versions of the same complaints.

1.2 Requirement Analysis

The above mentioned statistics highlight that there should be such a system which can process huge amount of data and retrieve the overall opinion hidden in it. But, creating systems that can process subjective information effectively requires overcoming a number of novel challenges. The development of a complete review- or opinion-search application as discussed in [1] might involve attacking each of the following problems.

(1) If the application is integrated into a general-purpose search engine, then one would need to determine whether the user is in fact looking for subjective material. This may or may not be a difficult problem in and of itself: perhaps queries of this type will tend to contain indicator terms like "review," "reviews," or "opinions," or perhaps the application would provide a "checkbox" to the user so that he or she could indicate directly that reviews are what is desired; but in general, query classification is a difficult problem.

(2) Besides the still-open problem of determining which documents are topically relevant to an opinion-oriented query, an additional challenge we face in our new setting is simultaneously or subsequently determining which documents or portions of documents contain review-like or opinionated material. Sometimes this is relatively easy, as in texts fetched from review-aggregation sites in which review-

oriented information is presented in relatively stereotyped format: examples include Epinions.com and Amazon.com. However, blogs also notoriously contain quite a bit of subjective content and thus are another obvious place to look (and are more relevant than shopping sites for queries that concern politics, people, or other non-products), but the desired material within blogs can vary quite widely in content, style, presentation, and even level of grammaticality.

(3) Once one has target documents in hand, one is still faced with the problems of identifying the overall sentiment expressed by these documents and/or the specific opinions regarding particular features or aspects of the items or topics in question, as necessary. Again, while some sites make this Introduction kind of extraction easier — for instance, user reviews posted to Yahoo! Movies must specify grades for pre-defined sets of characteristics of films — more free-form text can be much harder for computers to analyze, and indeed can pose additional challenges; for example, if quotations are included in a newspaper article, care must be taken to attribute the views expressed in each quotation to the correct entity.

(4) Finally, the system needs to present the sentiment information it has garnered in some reasonable summary fashion. This can involve some or all of the following actions:

(a) Aggregation of "votes" that may be registered on different scales (e.g., one reviewer uses a star system, but another uses letter grades).

(b) Selective highlighting of some opinions.

(c) Representation of points of disagreement and points of consensus.

(d) Identification of communities of opinion holders.

(e) Accounting for different levels of authority among opinion holders. Note that it might be more appropriate to produce a visualization of sentiment data rather than a textual summary of it, whereas textual summaries are what are usually created in standard topic-based multi-document summarization.

All the above factors highlight few concerns that will be faced while developing a recommendation search engine.

1.3 Potential Applications

The envisioned recommendation system has many applications. In this section, we seek to enumerate some of the possibilities. Due to all the possible applications, there are a good number of companies, large and small, that have opinion mining and sentiment analysis as part of their mission. But this is important to mention that none of the existing systems can compare multiple entities based on ad hoc dimensions specified by users using key words, which is what our system aims at. Few applications as highlighted in [1] are as follows:

1.3.1 Applications to Review-Related Websites

Clearly, the same capabilities that a review-oriented search engine would have could also serve very well as the basis for the creation and automated upkeep of review- and opinion-aggregation websites. That is, as an alternative to sites like Epinions that solicit feedback and reviews, one could imagine sites that proactively gather such information. Topics need not be restricted to product reviews, but could include opinions about candidates running for once, political issues, and so forth. For example, such an engine would guide a user which phone he should purchase based on his needs.

1.3.2 Applications as a Sub-Component Technology

Sentiment-analysis and opinion-mining systems also have an important potential role as enabling technologies for other systems. One possibility is as an augmentation to recommendation systems, since it might behove such a system not to recommend items that receive a lot of negative feedback.

Detection of “flames” (overly heated or antagonistic language) in email or other types of communication is another possible use of subjectivity detection and classification.

In online systems that display ads as sidebars, it is helpful to detect web pages that contain sensitive content inappropriate for ads placement; for more sophisticated systems, it could be useful to bring up product ads when relevant positive sentiments are detected, and perhaps more importantly, nix the ads when relevant negative statements.

It has also been argued that information extraction can be improved by discarding information found in subjective sentences.

Question answering is another area where sentiment analysis can prove useful. For example, opinion-oriented questions may require different treatment.

Summarization may also benefit from accounting for multiple view-Additionally, there are potentially relations to citation analysis, where, for example, one might wish to determine whether an author is citing a piece of work as supporting evidence or as research that he or she dismisses. Similarly, one effort seeks to use semantic orientation to track literary reputation.

1.3.3 Applications in Business and Government Intelligence

The field of opinion mining and sentiment analysis is well-suited to various types of intelligence applications. Indeed, business intelligence seems to be one of the main factors behind corporate interest in the field. Consider, for instance, the following scenario. A major computer manufacturer, disappointed with unexpectedly low sales, finds itself confronted with the question: “Why aren’t consumers buying our laptop?” Answering this question requires focusing more on people’s personal views of such objective characteristics. Moreover, subjective judgments regarding intangible qualities — e.g., “the design is tacky” or “customer service was condescending”. Sentiment-analysis technologies for extracting opinions from unstructured human-authored documents would be excellent tools for handling many business-intelligence tasks related to the one just described.

Hence, by designing such a system and by understanding weaknesses of a product, computer vendors compare Dell vs. Apple in a better way.

1.3.4 Applications Across Different Domains

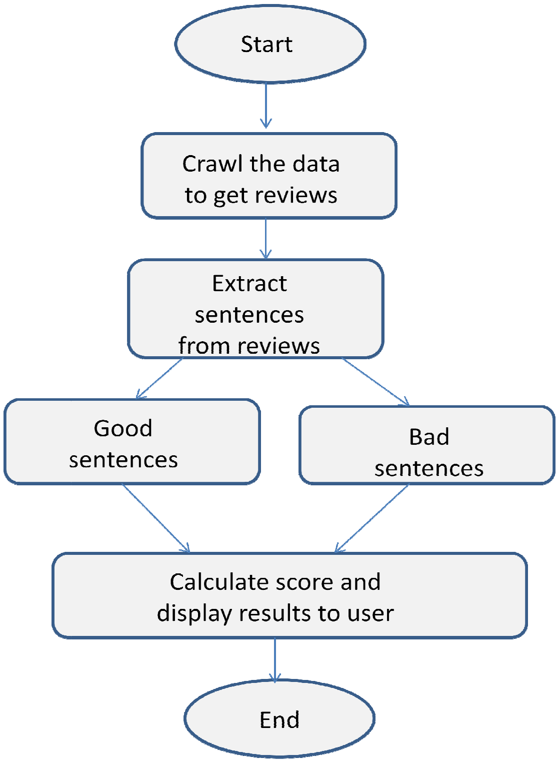
One exciting turn of events has been the confluence of interest in opinions and sentiment within computer science with interest in opinions. As is well known, opinions matter a great deal in politics. Some work has focused on understanding what voters are thinking, whereas other projects have as a long term goal the clarification of politicians' positions. Sentiment analysis has been proposed as a key enabling technology in e- rulemaking, allowing the automatic analysis of the opinions that people submit about pending policy or government-regulation.

CHAPTER 2. RECOMENDR- NOVEL ENTITY RECOMMENDER SYSTEM

In this thesis, we develop such a system (called Recomendr). It is a system where user can give any ad-hoc dimension as an input and then based on the set of dimensions, it compares the selected range of entities using reviews provided on related review based websites (that has been crawled by the web-crawler for this project). It then rates those reviews using a sophisticated scoring function and returns the aggregate opinion to the user. Making this system completely generic is definitely a difficult target to achieve in a small duration, hence for now Recomendr is specific to computer domain and recommends the best laptop as per user specified dimensions. Also, it is a necessary assumption for this system that we have huge collection of reviews as it uses words to categorize a sentence, instead of the sentence structure.

2.1 Flow Diagram

The above described flow of Recommendr can be well illustrated by the following simple flow diagram.


Start
Crawl the data to get reviews
Extract sentences from reviews
Good sentences
Bad sentences
Calculate score and display results to user
End

Each of the components described above as a black box are now explained in detail below.

2.2 Crawling the web

The first step towards this target was to extract the websites that contain user generated contents (opinions, sentiments etc provided by users) related to laptops. Examples of such websites include CNet rview.com, Epinions.com, Amazon.com etc. I used the amazon website as a prototype to extract user reviews. The amazon website contains a well-structured format and has the following features for each product:

Product ID: Unique identifier to distinguish various laptops of same brand.

Reviewer ID: ID of reviewer who commented on the product

Rating: Star Rating assigned to that product by the reviewer.

Date: Date when the review was written.

Review Title: Title of the review.

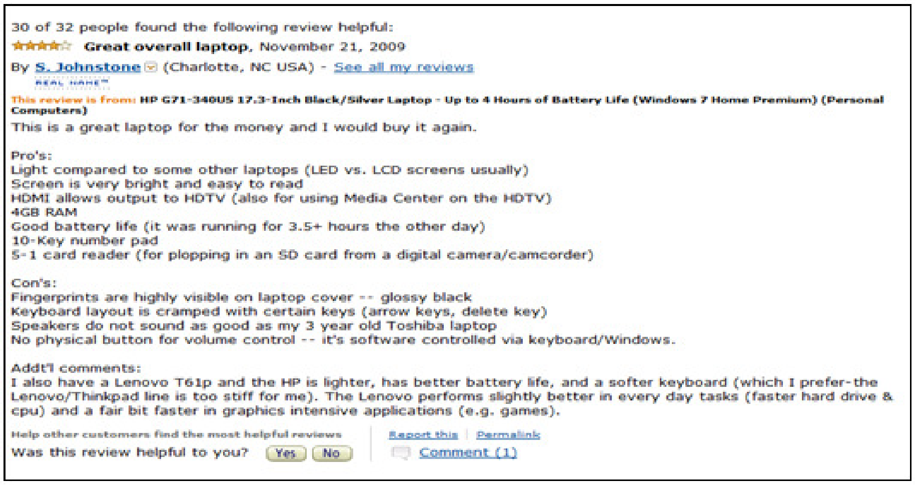
Review Body: The detailed comment about the product. This is the most useful part for Recomendr. The aggregate score calculations are based on the sentences in these reviews.

Number of Helpful feedbacks: The count of users who found this review useful.

Number of Feedbacks: The number of feedbacks on that particular review.

Comment: Further comments on review

A sample review on amazon which contains the above specified parts is as follows:



30 of 32 people found the following review helpful:
★★★★☆ **Great overall laptop**, November 21, 2009
By S. Johnstone (Charlotte, NC USA) - See all my reviews
REAL NAME™
This review is from: **HP G71-340US 17.3-Inch Black/Silver Laptop - Up to 4 Hours of Battery Life (Windows 7 Home Premium) (Personal Computers)**
This is a great laptop for the money and I would buy it again.

Pro's:
Light compared to some other laptops (LED vs. LCD screens usually)
Screen is very bright and easy to read
HDMI allows output to HDTV (also for using Media Center on the HDTV)
4GB RAM
Good battery life (it was running for 3.5+ hours the other day)
10-Key number pad
5-1 card reader (for plopping in an SD card from a digital camera/camcorder)

Con's:
Fingerprints are highly visible on laptop cover -- glossy black
Keyboard layout is cramped with certain keys (arrow keys, delete key)
Speakers do not sound as good as my 3 year old Toshiba laptop
No physical button for volume control -- it's software controlled via keyboard/Windows.

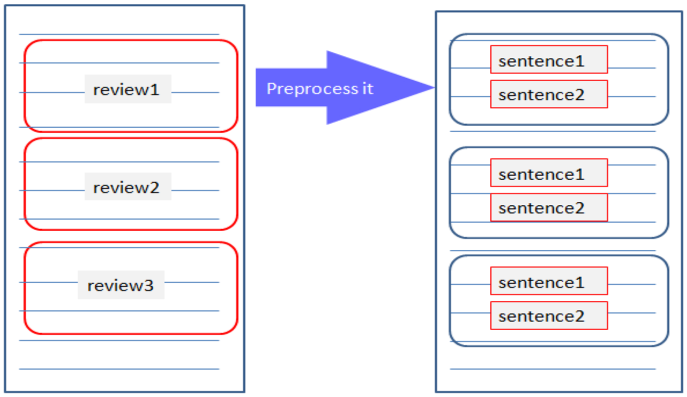
Addt'l comments:
I also have a Lenovo T61p and the HP is lighter, has better battery life, and a softer keyboard (which I prefer-the Lenovo/Thinkpad line is too stiff for me). The Lenovo performs slightly better in every day tasks (faster hard drive & cpu) and a fair bit faster in graphics intensive applications (e.g. games).

Help other customers find the most helpful reviews
Was this review helpful to you? Yes No
Report this | Permalink
Comment (1)

Using the crawler, the webpage was crawled and results were stored in a static database for future access to calculate score.

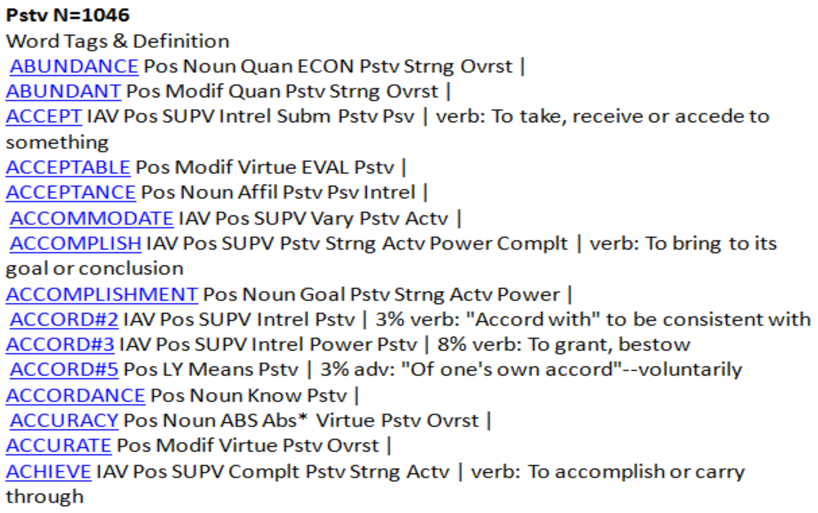
2.3 Extracting sentences from Review Body

The review body extracted from the amazon website (stored in the database) was then used for further analysis. Each review of a particular product was divided into sentences. These sentences were stored in a hash for future access. This division was done on the basis of the occurrence of period in that review. This approach although resulted in a few discrepancies as certain sentences were ending up with other punctuation characters like “;”, “,”, “ !” etc. But such cases were ignored as most of those sentences contained feature words as a pronoun (i.e. it instead of battery). Also, as per the assumption, we have a large collection of reviews, so these small discrepancies were ignored for simplicity.



2.4 List of positive and negative words

List of positive and negative words was created using an online dictionary which was used for an NLP project [22]. The list contained the positive and the negative words with their Pos tags. It was originally in the following format:

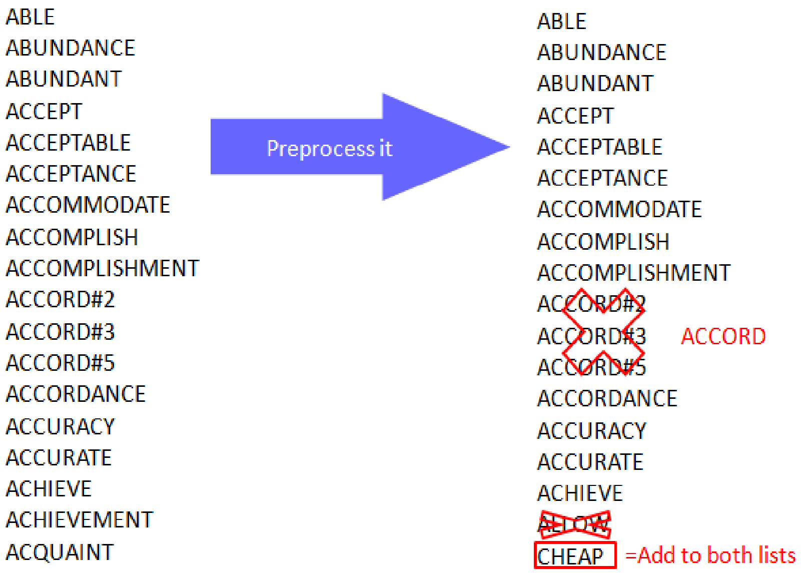


Pstv N=1046

Word Tags & Definition

ABUNDANCE Pos Noun Quan ECON Pstv Strng Ovrst |
ABUNDANT Pos Modif Quan Pstv Strng Ovrst |
ACCEPT IAV Pos SUPV Intrel Subm Pstv Psv | verb: To take, receive or accede to something
ACCEPTABLE Pos Modif Virtue EVAL Pstv |
ACCEPTANCE Pos Noun Affil Pstv Psv Intrel |
ACCOMMODATE IAV Pos SUPV Vary Pstv Actv |
ACCOMPLISH IAV Pos SUPV Pstv Strng Actv Power Complt | verb: To bring to its goal or conclusion
ACCOMPLISHMENT Pos Noun Goal Pstv Strng Actv Power |
ACCORD#2 IAV Pos SUPV Intrel Pstv | 3% verb: "Accord with" to be consistent with
ACCORD#3 IAV Pos SUPV Intrel Power Pstv | 8% verb: To grant, bestow
ACCORD#5 Pos LY Means Pstv | 3% adv: "Of one's own accord"--voluntarily
ACCORDANCE Pos Noun Know Pstv |
ACCURACY Pos Noun ABS Abs* Virtue Pstv Ovrst |
ACCURATE Pos Modif Virtue Pstv Ovrst |
ACHIEVE IAV Pos SUPV Complt Pstv Strng Actv | verb: To accomplish or carry through

The actual words were extracted from the list above and the word list was processed to get the final list of positive and negative words. There were three operations done on the actual list of words, shown in the figure below:

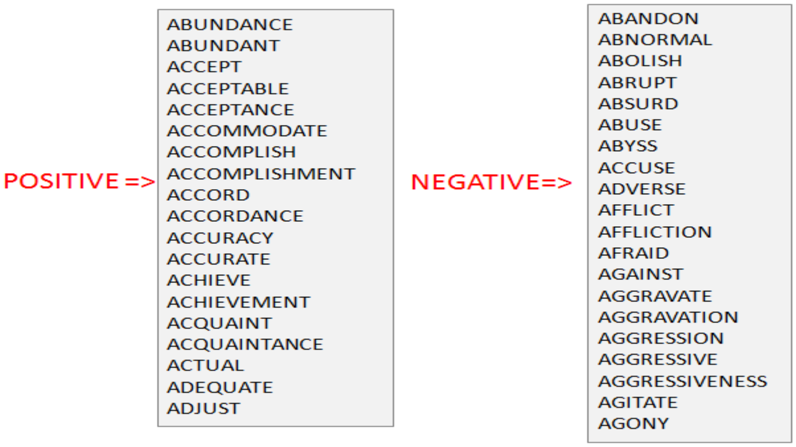


As shown in the figure above, a dictionary may contain multiple meanings of a single word, hence the same word appeared in the list many times. So the first pre-processing step was to remove such multiple occurrences of a single word. For example, in the figure above, the word “accord” can be used in 3 different meanings. This was changed to one.

There were certain positive or negative words which were not useful for specifying any entity as good or bad. Such words were manually removed from the list. For example, in the figure above, “allow” is one such word. This pre-processing step highly improved the ranking score.

Also, some words could be used in both positive and negative polarity sentences (e.g. cheap, beat). For example, a word cheap can be used as “The quality of laptop is cheap” or “The price of this laptop is cheap enough”. Such words were added to both the lists which balanced out the overall aggregate score.

After doing the above three operations on the list, the final list of positive and negative words was created, each containing approximately 1000 words. A snapshot of a part of that list is shown in the figure below:



POSITIVE =>

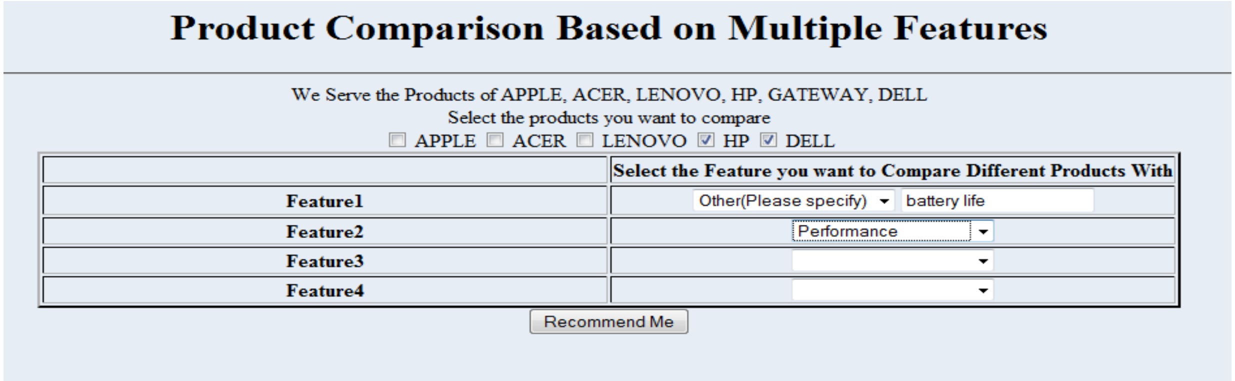
ABUNDANCE
ABUNDANT
ACCEPT
ACCEPTABLE
ACCEPTANCE
ACCOMMODATE
ACCOMPLISH
ACCOMPLISHMENT
ACCORD
ACCORDANCE
ACCURACY
ACCURATE
ACHIEVE
ACHIEVEMENT
ACQUAINT
ACQUAINTANCE
ACTUAL
ADEQUATE
ADJUST

NEGATIVE=>

ABANDON
ABNORMAL
ABOLISH
ABRUPT
ABSURD
ABUSE
ABYSS
ACCUSE
ADVERSE
AFFLICT
AFFLICTION
AFRAID
AGAINST
AGGRAVATE
AGGRAVATION
AGGRESSION
AGGRESSIVE
AGGRESSIVENESS
AGITATE
AGONY

2.5 Selecting entities and accepting ad-hoc dimensions from the user

User has an option of selecting a bunch of entities that it wants to compare. Also, most of the applications related to opinion mining either finds the features from the review body, or accepts static dimensions. RECOMENDR, however is capable of accepting ad-hoc dimensions from the user using the option "Other" at the time of input. It can also accept a multiple keyword based dimension as a query from the user. The application used an html page to take input. This page sends the selected input to a cgi file. It was noticed that html replaced spaces with special characters. The cgi file was hence made capable of handling this input. The figure illustrates the input given in the system by the user.



Product Comparison Based on Multiple Features

We Serve the Products of APPLE, ACER, LENOVO, HP, GATEWAY, DELL

Select the products you want to compare

□ APPLE □ ACER □ LENOVO ☑ HP ☑ DELL

	Select the Feature you want to Compare Different Products With
Feature1	Other(Please specify) battery life
Feature2	Performance
Feature3	
Feature4	

2.6 Sentence Polarity Analysis

Each sentence in the array of sentences for a particular review of a particular product was analysed to determine its polarity. By polarity, we mean the positivity or the negativity of each sentence. In this array of sentences, we discarded the following for simplicity:

- All those sentences which did not contain any of the feature words (battery, performance etc) provided by the user.
- The sentences which had negative words like "not" in it. This was not considered in this project for simplicity. E.g. battery is not good is a sentence which should have incremented the positive counter.
- The sentences where an object was described as a pronoun. E.g. "Battery is awesome. It works well". So, here the second sentence was discarded. (Note, the delimiter between the two sentences is period in this case, hence it will be discarded.)

For this analysis, the above lists of words were used to increment the good counter and bad counter for each sentence.

2.6.1 Algorithm

The following algorithm illustrates afore mentioned steps:

```
Foreach product, view each sentence in each review
{
    if it is a negative sentence
    {
        Discard it.
    }
    Else
    {
        For each sentence that has a feature
        {
            If it has a positive word
            {
                Increment good counter.
            }
            If it has a negative word
            {
                Increment  bad counter.
            }
        }
    }
}
```

2.6.2 Possible Ranking strategies

Once the normalized counters are obtained for a particular feature, we need to rank those sentences. To rank a particular sentence, we tried two different approaches:

2.6.2.1 Ranking each sentence as good or bad

Based on the good and the bad words counter of that sentence, we defined heuristic approach to characterize it as a good or bad sentence for a particular dimension. For example, if the good counter was twice as high as the bad counter, then it was termed as a positive sentence. However, if the counters were equal, such a sentence was ignored in the overall ranking. But, the drawback of this heuristic is that it can result in ignoring many sentences and can also gives equal weightage to a sentence with few or more occurrences of a particular feature.

2.6.2.2 Giving weight to each sentence for its polarity

All sentences were considered towards ranking using this approach. The good counters assigned the positive polarity to the sentence, and the bad counters assigned negative polarity. Hence, higher good counter automatically characterized it as a good sentence. This approach was better than the above as the above heuristic which was working well in the laptop domain may fail in other domain. Also this heuristic assigns a weight to a sentence. Hence more occurrences of characteristic words will add more weight to overall scoring of the sentence. We followed the results obtained by the second approach to calculate the score of each product in our prototype system.

2.7 Ranking Function

The system ranks the products in a scale of 0-10 where the highly recommended product is rated as 10. The total score for a particular product was a sum of its score for each feature in the sentences of all reviews. Hence this was the most critical part of the project as it required the tuning of parameters and helped in refining the crude list of positive and negative words. Two approaches were used for ranking.

2.7.1 Basic Ranking Score

The Basic Ranking can be judged on the basis of the good and the bad counters of each feature. Since the quantity of reviews per product varies in each case, the analysis cannot be made on the counters directly. For example, in the test dataset, number of reviews for apple was quite high as compared for acer, hence the counters need to be normalized with the total number of occurrences of good and bad words in all sentences of a review.

To get the overall rating for the product, we can sum up the individual feature scores. The system worked well on this approach.

$$Score\{feature\} = \frac{GC\{feature\}}{\theta} - \frac{BC\{feature\}}{\theta}$$

Where,

Feature= User specified feature e.g. battery, performance etc.

GC= Goodcounter for that feature.

BC= Bad words counter for that feature.

θ = Normalization factor i.e. Summation of good counters and bad counters

Simple normalization however failed to consider one special scenario when the bad score of both was negative. It is necessary while normalization to consider positive range. Hence, the final score was transformed in the range of 0 to 1 using logistic regression as follows:

$$\text{Logistic Regression}=\frac{1}{1+e^{-\text{score}\{\text{feature}\}}}$$

Recommended product was obtained using maximum of all scores, done twice for better scoring-range.

2.7.2 Modified Ranking Score

This approach was tried as a modification to the basic ranking score. It focused on some other details to be considered in deciding which entity is better. It has been seen that user mostly wish to give priority to each dimension when asking for a recommendation. For example, in some cases the user may have the constraint of low price; however some user may want a system that entails maximum performance. Hence, through modified ranking, we tried to fulfil this goal. Also, in some websites, like amazon, we get star ratings for each product. This function aimed to utilize it too towards the ranking of that product.

However, because of the user constraints, the overall star ratings cannot be authenticated in a interactive recommendation system. By this, we mean that for example apple has an overall ranking of 5 out of 5 stars. But, users constraint is that the price of the recommended product should be low. In such a case, we should not rate apple the best, although it star ratings rate it as the best.

To consider these sorts of concerns, the ranking function was upgraded in many different ways, one of those is:

$$Score\{feature\}=\left[\frac{GC\{feature\}}{\theta}-\frac{BC\{feature\}}{\theta}\right]*\frac{\Pr iority\{feature\}}{\phi}+\frac{TotalStarRating}{TF\{feature\}}$$

Where,

GC= Good counter for that feature

GC= Bad words counter for that feature

θ = Normalization factor

Φ= Weightage to positive words

TF {feature} = Term frequency of the feature so as to divide star rating for the particular feature

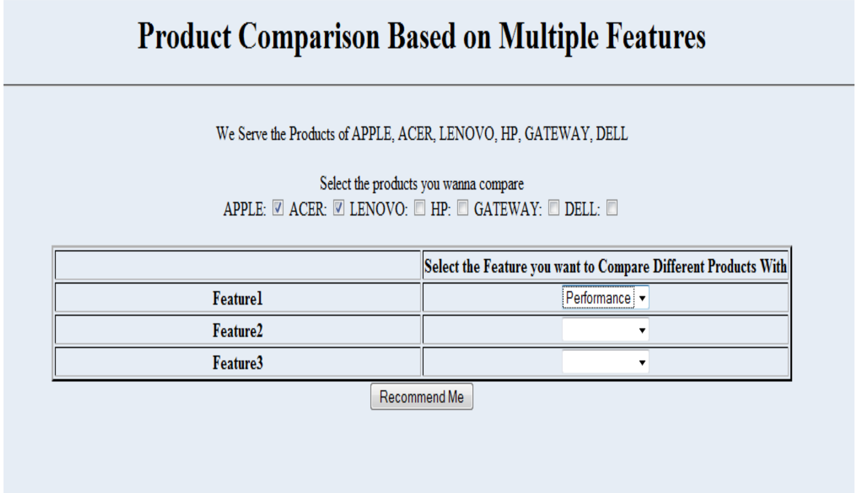
Total Star rating= Star rating of each review

Star ratings were majorly useful in adding some weight for each feature when we have no relevant sentences for a particular feature in the set of reviews. We even tried to give weightage to star ratings but it would have resulted in parameter tuning and the value would have been different for different domains. Since, the aim is to make the system generic for all domains, hence due to above factor this ranking function seems less promising.

Also, there was one more issue with a sophisticated ranking function. To evaluate the results, maintaining the ground truth is non-trivial for such a function which decides a sentence as good or bad for the feature as per user's defined priority. This is non-trivial to accomplish manually, hence we just used the simple ranking score for all the evaluation and analysis.

CHAPTER 3. USER- INTERFACE AND SAMPLE RESULTS

The front-end of the prototype system accepts the products to compare and also asks the user to specify the features which he wants to be compared. It looks as follows:



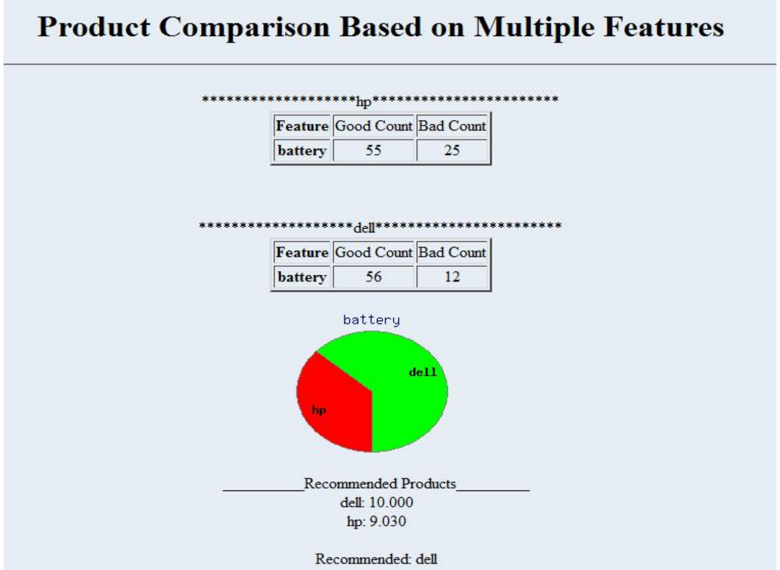
Product Comparison Based on Multiple Features

We Serve the Products of APPLE, ACER, LENOVO, HP, GATEWAY, DELL

Select the products you wanna compare
APPLE: ☑ ACER: ☑ LENOVO: □ HP: □ GATEWAY: □ DELL: □

	Select the Feature you want to Compare Different Products With
Feature1	Performance ▾
Feature2	▾
Feature3	▾

Using the basic Ranking approach, the prototype system was used to calculate the score of each product. The results for each feature were represented in a graphical way for better analysis of the differences in each feature for every product as follows:



Product Comparison Based on Multiple Features

********************hp************************

Feature	Good Count	Bad Count
battery	55	25

********************dell************************

Feature	Good Count	Bad Count
battery	56	12

__________Recommended Products_________
dell: 10.000
hp: 9.030

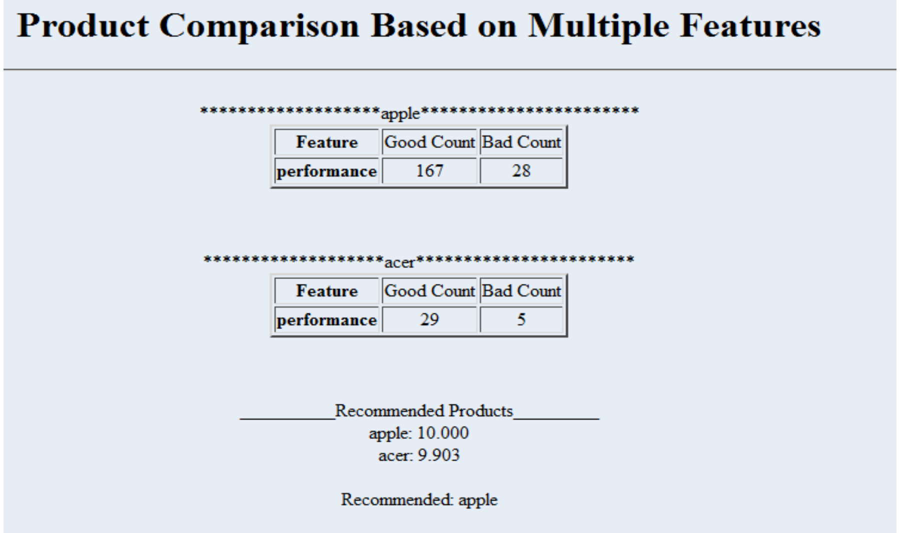
Recommended: dell

Also, the system is capable of showing the good words and their counters for each entity. This information helps the user to believe the correctness of the system. Also, there is an option to view the sentences, which were considered in scoring.

Few worth mentioning experiments were as follows:

3.1 Comparing Apple and Acer if user specifies "Performance" as a dimension.

When Apple and Acer were compared on basis of performance, the system recommended Apple as most of the reviews rated Apple as best in performance. The following is the snapshot of the results provided for the individual counters of each product and the recommended score.



Product Comparison Based on Multiple Features

********************apple************************

Feature	Good Count	Bad Count
performance	167	28

********************acer************************

Feature	Good Count	Bad Count
performance	29	5

__________Recommended Products_________
apple: 10.000
acer: 9.903

Recommended: apple

3.2 Comparing Apple and Acer if user specifies “Price” as a dimension.

When Apple and Acer were compared on basis of price, the system recommended Acer as most of the reviews rated Apple as expensive. The following is the snapshot of the results provided for the individual counters of each product and the recommended score.

```
___________Recommended Products_________
              apple: 8.991
              acer: 10.000

          Recommended: Acer
```

3.3 Comparing a list of laptops on price, battery and performance as dimensions.

If a number of laptops were compared over a list of features, the system returned a ranked list and recommended the following:

```
___________Recommended Products_________
               dell: 10.000
               gateway: 9.686
              acer: 9.434
              lenovo: 8.793
              apple: 8.787
              hp: 6.469

           Recommended: Dell
```

All afore mentioned results are some well known observations. The system was able to recommend it which is really interesting.

CHAPTER 4. QUANTITATIVE EVALUATION

To evaluate such a system, finding a gold standard was difficult. Hence, one way to evaluate is by analysing if the system's recommendation is consistent with manual judgments.

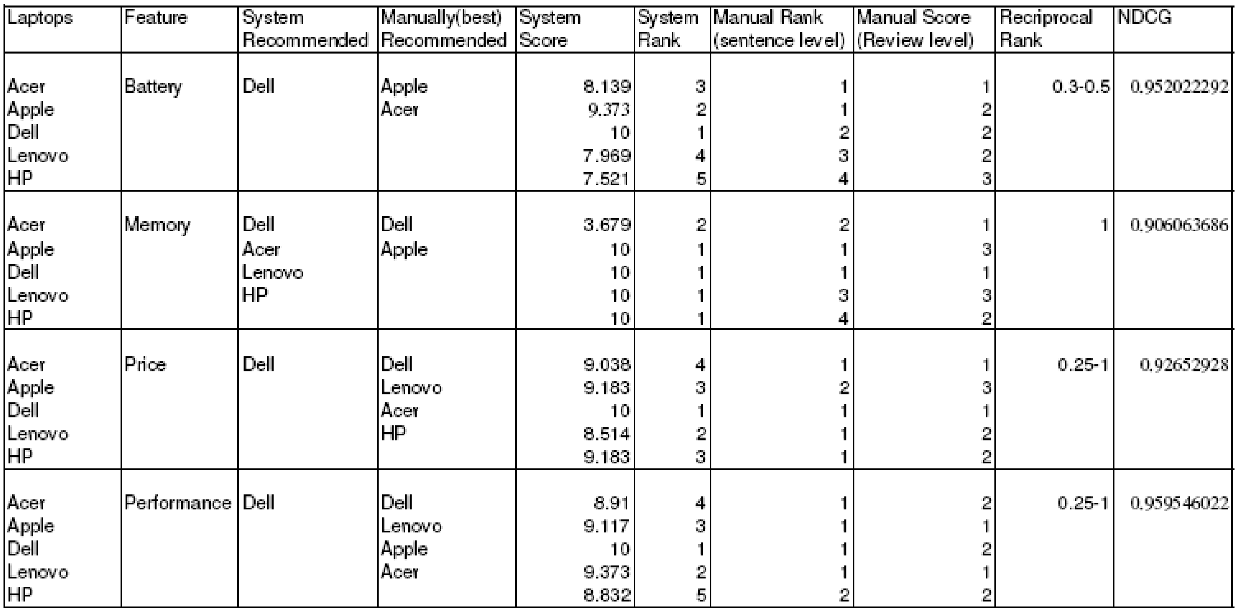
In order to have manual judgement, I manually selected 4 reviews for each dimension for each entity by just randomly selecting few reviews that contained some specific features. This process was repeated for all the laptops. The features were recorded as query set. These reviews were read and the opinion over a set of features with all entities selected was recorded.

The same set of reviews was passed to the RECOMMENDR. The queries in the query set were inquired from the system. The two results were then compared (i.e. system vs. Manual).

Manual evaluation was done in two ways:

1) Evaluating the sentiment over each dimension on each sentence of the review.
2) Evaluating all reviews on product level and forming a collective recommendation for each dimension.

The results obtained were as follows:



Laptops	Feature	System Recommended	Manually(best) Recommended	System Score	System Rank	Manual Rank (sentence level)	Manual Score (Review level)	Reciprocal Rank	NDCG
Acer	Battery	Dell	Apple	8.139	3	1	1	0.3-0.5	0.952022292
Apple			Acer	9.373	2	1	2		
Dell				10	1	2	2		
Lenovo				7.969	4	3	2		
HP				7.521	5	4	3		
Acer	Memory	Dell	Dell	3.679	2	2	1	1	0.906063686
Apple		Acer	Apple	10	1	1	3		
Dell		Lenovo		10	1	1	1		
Lenovo		HP		10	1	3	3		
HP				10	1	4	2		
Acer	Price	Dell	Dell	9.038	4	1	1	0.25-1	0.92652928
Apple			Lenovo	9.183	3	2	3		
Dell			Acer	10	1	1	1		
Lenovo			HP	8.514	2	1	2		
HP				9.183	3	1	2		
Acer	Performance	Dell	Dell	8.91	4	1	2	0.25-1	0.959546022
Apple			Lenovo	9.117	3	1	1		
Dell			Apple	10	1	1	2		
Lenovo			Acer	9.373	2	1	1		
HP				8.832	5	2	2		

The above table shows the results when the specified selected laptops were compared based on the specified features, what was recommended by the system and by manual judgments respectively. The fifth column in the table represents the overall score reported by the system and the sixth column ranks that score. The seventh and the eighth column represent the ranks given to each entity manually by the judge using afore mentioned two strategies (i.e. sentence level and review level).

To find the match between the system ranking and the manual ranking, we need to compare the two lists of scores. To obtain this similarity between the two list of scores, Reciprocal Rank was used. In the above table, the system rank was compared to the manual rank (sentence level). Reciprocal Rank of a system calculates arithmetic mean average precision over a set of products and returns the top best option. It is calculated as follows:

$$\text{Reciprocal Rank}=1/\text{Rank of relevant document}$$

However in few cases, multiple products might have same score, which cannot be dealt with a Reciprocal Rank evaluation. Hence, in such cases the range of accuracy was calculated selecting the best and worse matches between the two systems.

For example, in query 1 i.e. comparison on the basis of battery, the manual judgements ranked Apple Acer as rank 1. System however generated Dell giving Acer=rank 3 and Apple=rank 2. Therefore, Reciprocal Rank will be 1/3=0.3 or 1/2 =0.5, hence ranging from 0.3 to 0.5.

Hence, Mean Reciprocal Rank resulted in an overall accuracy range of 80-88%.

Another measure, known as Normalized Discounted Cumulative Gain (NDGC), was used to compare the ranking list generated manually and by the system. NDCG first calculates the Cumulative Gain as follows for 'n' ranks:

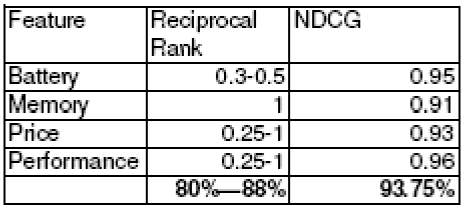
$$CG= r_1 + r_2 + \ldots + r_n$$

Then, it calculates Discounted Cumulative Gain as follows:

$$DCG= r_1 + r_2 /\log_2 2+ r_3 /\log_2 3+..+ r_n /\log_2 n$$

Finally, NDGC is a ratio of calculated DCG to ideal DCG which in this case is considered as 3-manual rank. It gave results in range of 0-1, which was finally mapped to a scale of 0-10. This resulted in system accuracy of 93.75%.

Concise results of the two measures, Reciprocal Rank and NDCG, obtained from the table above are illustrated as follows:



Feature	Reciprocal Rank	NDCG
Battery	0.3-0.5	0.95
Memory	1	0.91
Price	0.25-1	0.93
Performance	0.25-1	0.96
	80%—88%	**93.75%**

The evaluation exhibits that the manual judgements and the system judgements agreed to each other with an accuracy of 93.75%. Such a result shows the system, RECOMENDR, to be successful, exhibiting the fact that if the data is huge, the system can result in quite accurate recommendation even if you ignore many sentences/instances while evaluating them automatically.

CHAPTER 5. RELATED WORK

To the best of my knowledge, no previous work has proposed an entity comparator based on ad-hoc dimensions but there are several lines of related work. Hu and Liu [3] apply association mining to extract product features and decide the polarity of opinions using a seed set of adjective expanded via WordNet. A similar work of OPINE [4] outperforms Hu and Liu's system both in feature extraction and opinion polarity identification. Lu et al. [2] studies the problem of generating a "rated aspect summary" of short comments, which is a decomposed view of the overall ratings for the major aspects, found by the algorithm, so that a user could gain different perspectives towards the target entity. But [3,4,5] fails to accept ad-hoc dimensions. A different approach in the supervised framework is to learn the rules of aspect extraction from annotated data. For example, Zhuang and others [5] focused on movie review mining and summarization. The short coming is that the techniques are limited to the specific domain and highly dependent on the training data. Sentiment classification is usually defined as the problem of binary classification of a document or a sentence [9, 8, 11, 10] but even these fail to consider ad-hoc dimensions from the user. In some recent work, Pang and Lee generalize the definition into a rating scale [6]. Snyder and Barzilay [7] improve aspect level rating prediction by modelling the dependencies between aspects. This line of work aims at improving classification accuracy. One work which is the most closest to RECOMENDR is in [21], which finds ratings based on reviews, but even this assumes the aspect ratings to be latent. Many other related works focus on a single entity rating and fixed aspects but none compares the two entities on ad-hoc dimensions. Recomendr focuses to achieve this target by using simplest approaches but gain an overall effect of reviews and then compare both to decide the best amongst the two.

CHAPTER 6. CONCLUSION

The enormous availability of opinion-rich resources such as online review sites and personal blogs, it has been necessary to have built automated opinion-oriented information-seeking systems which can determine the opinion and subjectivity in text, based on the specified features of concern provided by the user.

RECOMENDR is such a system which takes keyword specified ad-hoc dimensions as input from the user and then based on those feature set; it compares the selected range of entities using reviews/opinions provided on related websites. It rates the textual stream of data using a scoring function and returns the decision based on an aggregate opinion to the user.

Although the approach used to do the above task is simple, this results in system accuracy of 93.75%. With this experiment, it is learnt that if we have huge dataset for processing, we can still get better accuracy.

Although we only evaluated the system using a small laptop data set, Recomendr is a general system that can support comparison and recommendation of any entities for which we have online reviews or opinionated text available. This is because it does all the processing on sentence level. Thus it can potentially help users digest opinions and support decision making in many different domains for example, it is capable of telling you which politician is better based on their democratic behaviour, which university you should opt to go if Artificial Intelligence is your majors.

CHAPTER 7. CHALLENGES AND FUTURE WORK

The system RECOMENDR just considered the direct opinion sentences i.e. the sentences that expressed a direct opinion on the specified dimension i.e.

E.g., “the picture quality of this camera is great”.

However, while working on the sentence polarity, many challenges were faced in its ranking. This is because the sentences in the user generated contents had various syntax involved. The sentence categories described below are some major directions which can further improve the accuracy of the results , hence can be considered as a extension in future.

1. Comparisons: Certain sentences were relations expressing similarities or differences of more than one object. Usually, it was expressing an ordering. Such sentences were difficult to parse as it involves Natural Language Processing to understand the comparative nature of the sentence and identifying the subject and object in it.

E.g., “car x is cheaper than car y.”

2. Negative Sentences: Certain sentences had sentiments expressing some object with a negative intention but using positive words. Such sentences were different as they were incrementing the opposite counters i.e. for the sentence below it needs to increment the positive counter for the feature battery.

E.g. Battery is not good

Handling negation can be an important concern in opinion- and sentiment-related analysis. While the bag-of-words representations, Classification and Extraction of “I like this book” and “I don’t like this book” are considered to be very similar by most commonly-used similarity measures, the only differing token, the negation term, forces the two sentences into opposite classes.

Another difficulty with modeling negation is that negation can often be expressed in rather subtle ways. Sarcasm and irony can be quite difficult to detect, but even in the absence of such sophisticated rhetorical devices, we still see examples such as “Its battery avoids power loss”. — the word “avoid” here is an arguably unexpected “polarity reverser.”

4. Subject Identification: Some sentences had pronouns used instead of the feature itself hence using the sentence split strategy, the second sentence in the example below will be “It is awesome”. Here, it is difficult to link a subject to the pronoun “it” as it again needs NLP syntactic analysis.

E.g. Battery works well. In fact, it works awesome!

These are few directions which can be improved. But on the contrary, the ranking algorithm to decide polarity which is actually a component of the system can be completely replaced by a more sophisticated algorithm, but the major idea of this whole effort highlights the utility of this field, opinion mining, into a system that can help a layman.

REFERENCES

[1] Bo Pang and Lillian Lee: Opinion Mining and Sentiment Analysis

[2] Yue Lu , ChengXiang Zhai, Neel Sundaresan: Rated Aspect Summarization of Short Comments in WWW(2009)

[3] M. Hu and B. Liu. Mining and summarizing customer reviews. In W. Kim, R. Kohavi, J. Gehrke, and W. DuMouchel, editors, KDD, pages 168–177. ACM, 2004.

[4] A.-M. Popescu and O. Etzioni. Extracting product features and opinions from reviews. In Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, pages 339–346, 2005.

[5] L. Zhuang, F. Jing, and X.-Y. Zhu. Movie review mining and summarization. In Proceedings of the 15th ACM international conference on Information and knowledge management, pages 43–50. ACM, 2006.

[6] B. Pang and L. Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In Proceedings of the ACL, pages 115–124, 2005.

[7] B. Snyder and R. Barzilay. Multiple aspect ranking using the good grief algorithm. In HLT-NAACL, 2007.

[8] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up? Sentiment classification using machine learning techniques. In Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 79–86, 2002.

[9] P. D. Turney. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, pages 417–424, 2002.

[10] S.-M. Kim and E. Hovy. Determining the sentiment of opinions. In Proceedings of the 20th international conference on Computational Linguistics, page 1367, 2004.

[11] H. Cui, V. Mittal, and M. Datar. Comparative experiments on sentiment classification for online product reviews. In Twenty-First National Conference on Artificial Intelligence, 2006.

[12] comScore/the Kelsey group. Online consumer-generated reviews have significant impact on offline purchase behavior. Press Release, November 2007. http://www.comscore.com/press/release.asp?press=1928.

[13] John A. Horrigan. Online shopping. Pew Internet & American Life Project Report, 2008

[14] B. Liu, M. Hu, and J. Cheng. Opinion observer: Analyzing and comparing opinions on the web. In WWW '05, pages 342-351, 2005

[15] K. Dave, S. Lawrence, and D. M. Pennock. Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In WWW '03, pages 519-528, 2003.

[16] A. Devitt and K. Ahmad. Sentiment polarity identification in financial news: A cohesion-based approach. In Proceedings of ACL'07, pages 984-991, 2007.

[17] K. Jarvelin and J. Kekalainen. IR evaluation methods for retrieving highly relevant documents. In Proceedings of SIGIR'00, pages 41-48. ACM, 2000.

[18] H. Kim and C. Zhai. Generating Comparative Summaries of Contradictory Opinions in Text. In Proceedings of CIKM'09, pages 385-394, 2009.

[19] I. Titov and R. McDonald. A joint model of text and aspect ratings for sentiment summarization. In ACL '08, pages 308-316.

[20] Y. Yang and J. O.Pedersen. A comparative study on feature selection in text categorization. In Proceedings of ICML'97, pages 412 - 420, 1997.

[21] Hongning Wang, Yue Lu, Chengxiang Zhai.Latent Aspect Rating Analysis on Review Text Data: A Rating Regression Approach

[22] List of virtue and vice words: http://www.webuse.umd.edu:9090/tags/