REVIEW MINING: HIERARCHY GENERATION FOR ONLINE REVIEWS

A Paper
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Prateek Rajan

In Partial Fulfillment of the Requirements
for the Degree of
MASTER OF SCIENCE

Major Department:
Computer Science

November 2015

Fargo, North Dakota

North Dakota State University

Graduate School

Title

Review Mining: Hierarchy Generation for Online Reviews

By

Prateek Rajan

The Supervisory Committee certifies that this ***disquisition*** complies with North Dakota State University's regulations and meets the accepted standards for the degree of

MASTER OF SCIENCE

SUPERVISORY COMMITTEE:

Dr. Wei Jin

Advisor

Dr. Gursimran Walia

Dr. Na Gong

Approved:

11/16/2015

Date

Dr. Brian M. Slator

Department Chair

ABSTRACT

In the present world of ecommerce more and more products are purchased and sold online then via any other medium. With such massive drive in online shopping more and more information is being added every day on web regarding the products and how good or bad are they. From the perspective of seller (such as Amazon) this information is very vital as this insight could be very helpful in making various decisions regarding inventory management, product pricing and so on. But the problem that arises in this context is the sheer volume of the reviews being added. In this paper we have proposed a way of extracting the semantics out of the reviews via use of various linguistic and statistical techniques. The idea is to extract the relevant information from the review and represent it in most concise format to make it more suitable for later processing.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to everyone who has supported me during the completion of this paper. I would like to thank Dr. Wei Jin for all her support and guidance. Without her support and influence this paper would not have been completed. I would like to thank computer science department for providing me necessary course work, which helped me a lot during my research. The course also motivated my research interest in the field of opinion mining. I would like to thank Stephanie Sculthorp (Administrative Secretary) of the Computer Science department for timely reminders and encouraging me to complete my research on time. At last I would like to thank my friends and family who always encouraged me to pursue my career at North Dakota State University.

TABLE OF CONTENTS

ABSTRACT iii
ACKNOWLEDGMENTS iv
LIST OF TABLES vii
LIST OF ALGORITHMS viii
IMPORTANT TERMINOLOGIES 1
Part-of-Speech tags 1
Grammatical dependency tags 2
CHAPTER 1. INTRODUCTION 5
CHAPTER 2. RELATED WORK 8
CHAPTER 3. PROPOSED APPROACH 12
- 3.1. Creating component database 12
- 3.2. Extracting root features 13
- 3.3. Pruning irrelevant sentences 14
- 3.4. Generating features 15
  - 3.4.1. Using grammetical dependencies 16
  - 3.4.2. Using sentence patterns 18
- 3.5. Generating opinions 21
  - 3.5.1. Using grammatical dependencies 21
  - 3.5.2. Using closest adjective approach 23

CHAPTER 4. EXPERIMENTAL RESULTS 25
- 4.1. Data sets 25
  - 4.1.1. Simple sentence structures 25
  - 4.1.2. Comma separated features/components 26
  - 4.1.3. Negative semantics structure 26

4.1.4. Conjunction separated features/sentences 27

4.1.5. Compound sentence structures 27

4.2. Evaluation parameters 28

4.3. Performance evaluation 29

4.4. System performance 31

CHAPTER 5. CONCLUSION & FUTURE WORK 33

REFERENCES 35

LIST OF TABLES

Table Page

1. POS tags 1

2. Simple sentence structures 29

3. Comma separated features/components 29

4. Negative semantics structure 30

5. Conjunction separated features/sentences 30

6. Compound sentence structures 31

7. Total system performance 31

LIST OF ALGORITHMS

Algorithm Page

1. Root feature extraction 13

2. Removing irrelevant sentences 14

3. Composite feature extraction 17

4. Standalone feature extraction 17

5. Generating standalone features using NN IN NN patterns 19

6. Extracting composite features using sentence patterns 20

7. Extracting directly opinioned features 21

8. Extracting opinions nsubj 22

9. Extracting opinions amod 22

10. Extracting opinions nusbj & advmod 23

11. Extracting opinions using proximity adjectives 24

IMPORTANT TERMINOLOGIES

Part-of-Speech tags

Stanford POS tagger tags the words in the review sentences with their corresponding part of speech tag. Below is the list of some relevant tags that we have used in our approach.

Table 1. POS tags

IN	Preposition or subordinating conjunction
JJ	Adjective
JJR	Adjective, comparative
JJS	Adjective, superlative
LS	List item marker
MD	Modal
NN	Noun, singular or mass
NNS	Noun, plural
NNP	Proper noun, singular
NNPS	Proper noun, plural
PDT	Predeterminer
POS	Possessive ending
PRP	Personal pronoun
PRP$	Possessive pronoun
RB	Adverb
RBR	Adverb, comparative
RBS	Adverb, superlative

RP	Particle
SYM	Symbol
TO	to
UH	Interjection
VB	Verb, base form
VBD	Verb, past tense
VBG	Verb, gerund or present participle
VBN	Verb, past participle
VBP	Verb, non-3rd person singular present
VBZ	Verb, 3rd person singular present
WDT	Wh-determiner
WP	Wh-pronoun
WP$	Possessive wh-pronoun
WRB	Wh-adverb

Grammatical dependency tags

The grammatical dependencies generated by the Stanford POS tagger are based on the relationships that exist between different words in the sentences. Understanding the relationships between the words in the sentence can provide a meaningful insight about the Feature/Components – Opinions relationships that we are interested in. Some of the important grammatical dependency tags that we have used in our approach are mentioned below:

a. *Nsubj*: nominal subject

A nominal subject is a noun phrase, which is the syntactic subject of a clause. The governor of this relation might not always be a verb: when the verb is a copular verb, the root of the clause is the complement of the copular verb, which can be an adjective or noun.

Example:

The size of the camera is perfect.

nsubj (perfect-7, size-2)

As evident from this example nsubj can be very useful in extracting the **component/feature-opinion relationship** in a review sentence.

b. *Poss*: possession modifier

The possession modifier relation holds between the head of an NP and its possessive determiner, or a genitive's complement.

Example:

The camera's viewfinder is perfect.

poss (viewfinder-4, camera-2)

The poss grammatical dependency can be used to extract the **parent-feature/component** relationships that exist in the review sentences.

c. *Prep*: prepositional modifier

A prepositional modifier of a verb, adjective, or noun is any prepositional phrase that serves to modify the meaning of the verb, adjective, noun, or even another preposition. In the collapsed representation, this is used only for prepositions with NP complements.

Example:

The size of the camera is perfect.

prep_of (size-2, camera-5)

Prep grammatical dependency is usually suffixed with a number of suffixes like of, in, on, at etc. This dependency is highly useful in extracting the parent-feature/component relationship that exists in the review sentences.

d. *NN*: noun compound modifier

A noun compound modifier of an NP is any noun that serves to modify the head noun. (Note that in the current system for dependency extraction, all nouns modify the rightmost noun of the NP - there is no intelligent noun compound analysis. This is likely to be fixed once the Penn Treebank represents the branching structure of NPs.).

Example:

The picture quality of the camera is awesome.

nn(quality-3, picture-2)

As evident from our example, for our particular use case we have used NN grammatical dependency for extracting the composite features from the relevant review sentences.

CHAPTER 1. INTRODUCTION

Knowing what other people think has been an important piece of information that gives direction to our ultimate decision. Even before dot COM boom when people used to do most of the shopping in stores and malls our buying decision were mostly governed by what our friends and family said about a particular product. But in present world of e-commerce we are more inclined towards online shopping rather than going to the stores and buying something. Also 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.

To understand how opinions affect our buying decisions two surveys were conducted across 2000 American adults [3],[4].

1. 81% of Internet users (or 60% of Americans) have done online research on a product at least once.
2. Among readers of online reviews of restaurants, hotels, and various services (e.g., travel agencies or doctors), between 73% and 87% report that reviews had a significant influence on their purchase;
3. 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);

4. 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.

It is evident from the above-mentioned facts that one of the best ways for an online ecommerce firm to get insights about customer's interest for a particular product is via the reviews for that particular product. Reviews provide an insight of the level of customer's satisfaction. Simple questions like; was the product doing what the customer wanted? Was the product able to satisfy customer's needs? Can provide valuable information that can help the ecommerce firm in deciding its next stock purchase, but the numbers of these reviews are increasing manifolds as the numbers of customers are increasing. This is where review mining comes into picture.

Review Mining can be defined as gathering of data from a wide range of reviews and represent it into more understandable and descriptive form. The project deals with this core issue. The core idea behind the project is to create an algorithm that can take a review as its input and from that create a hierarchy based on the components and there corresponding opinions. This sort of result is much easier for later processing then the raw reviews themselves. But the biggest challenge in solving this core issues is the sheer flexibility of the natural language. Professionals do not write the reviews online nor are the people writing reviews required to adhere to a particular format. This results in reviews that are very diverse and noise in nature. Designing a system that is able to handle every aspect of possible review sentences is very complex task and approaches NP hard problem.

One of the techniques that can be used for opinion mining is pattern matching which is based on existence of certain common structures in the fragments of the sentences. By splitting the review sentence into simpler fragments and then analyzing each fragment for a valid pattern is an effective technique to extract the crux of the review sentence. Selecting the appropriate patterns that can be used for the extracting the valid information from the review sentence is the core of this approach. Other then the pattern matching technique in this paper we also have discussed grammatical dependency approach which is based on the existence of the dependencies among different words in the review sentence. While pattern matching is more inclined towards existence of certain patterns in the review sentence the grammatical dependency approach is more focused on the relationships that exists among different components of the review sentence.

In most of the earlier approaches researchers have concentrated more on the machines learning techniques both supervised and unsupervised techniques. Using both of these techniques in a hybrid fashion to extract semantics from review sentences is the core idea behind this paper that hasn't been tried in any of the earlier technique.

Also, we would be evaluating the performance of this new approach against multiple review structure and understand how efficient or inefficient this technique is in extracting the relevant information from the review sentences.

CHAPTER 2. RELATED WORK

Idea of opinion mining is not new. With the boom in the online retail the problem of extracting semantics from the review sentences is being extensively researched. Many techniques have been proposed and implemented to solve this problem. Our work in particular is closely related to Srivastaval, Bhatia, Srivastava and Sahu's work presented in "Exploiting Grammatical Dependencies for Fine-grained Opinion Mining" [9].

Early approaches for solving this problem [7],[8] included determining the overall polarity of the review sentence. Though these approaches are good in providing a single polarity of the review sentence they fail to provide information about the different aspects present in the review sentence. For example, a review "This camera is OK. I picture quality is great but the bulky size is terrible." Would have a positive polarity but this doesn't give any insight about different aspects of the camera and different people can have different preferences. An enthusiastic photographer won't mind the bulky size of the camera as long as the picture quality is at par but an avid traveller might want to compromise of the picture quality as long as the camera is more portable. Hence, not each aspect of the product mentioned in the actual review sentence hold the same level of importance or same influence on the decision-making. In this respect our approach is different from that mentioned in [7],[8] because our approach deals with the review sentences at the level of aspects rather then taking a review sentence as a single atomic entity. This resulted into a new branch of sentiment analysis called Multi-aspect sentimental analysis where each separate aspect mentioned in the review is handled individually hence providing a more granular level of sentiment analysis.

One of the most widely used techniques from extracting the sentiments from the literature sentence involves extensive use of Machine learning. In "*Machine Learning Algorithms for*

Opinion Mining and Sentiment Classification " *Khairnar* & *Kinikar* [5] have discussed many of the machine learning approaches including Naïve Bayes & Support Vector Machines (SVM). The approach is straightforward. Models are trained on the training set consisting of millions of review manually tagged. Once the training is completed the model can predict which of the words in the review sentences are possible feature and which of the words are possible opinions. Though this technique is effective in most scenarios there are some drawbacks that make this technique not suitable for high volume opinion mining.

Most of these techniques discussed in [5] are supervised machine learning techniques involving Naïve Bayes classification & Support Vector Machines (SVMs). Both Naïve Bayes and SVM take into consideration a set of input training examples, which have already been tagged to a particular category and build a model using which it assigns a category for every input test example. The biggest issue with the supervised machines learning technique is that it requires large amount of labeled training data to start with. Most of the domains don't necessarily have available huge sets of labeled data, which puts a constraint on the usability of these techniques. On the other hand, our approach is not dependent upon an initial set of labeled training data. It uses initial corpora of reviews to get frequent nouns but it doesn't require it to be labeled as required by the supervised machines learning models. One important thing to note here is that our approach uses the initial corpora only to extract the product class of the review sentences provided. In case class of product is already know our approach wouldn't require an initial corpora to begin with, making our approach readily useful out of the box and not requiring any sort of initial requirements to be effective. Another important aspect to consider here is that, supervised machines learning techniques are highly coupled with the quality of the training data. The quality of the training data highly dictates how good would be the precision and recall during test phase. Following on the similar lines, as our approach isn't dependent upon the initial training data the performance isn't constrained. Moreover the quantity and quality of review being generated is so different in the present e commerce scenario that a training set readily loses its relevance; which brings up to another issue corresponding to this approach. Without initial training set the system is not capable of doing any prediction or tagging. This is an important concern as for any new type of review structure we need to initially generate a sample training set. Hence this technique is not readily useful.

Another technique being used is what is called Double Propagation, which was proposed by *Guang Qiu, Bing Liu, Jiajun Bu and Chun Chen* in "*Opinion Word Expansion and Target Extraction through Double Propagation*" [6]. It is an unsupervised technique for solving the problem. Because of it being an unsupervised technique it overcomes the problem that is common in the techniques based on the supervised learning. There is no need for an initial training set that would be used as a base to train the model.

It exploits certain syntactic relations of opinion words and features, and propagates through both opinion words and features iteratively. Although as mentioned in "*A Survey on Feature Level Sentiment Analysis*" [10] the performance of supervised machines learning techniques is better than that of unsupervised ones the flexibility of not having an initial training set is good enough reason to use these techniques.

Our approach has some similarities with Double propagation. Both techniques use relationships among the opinions and features in the sentences but where Double propagation is a recursive way of extracting feature/opinions starting from an initial opinion pool, our technique only applies some certain grammatical relations to the review sentences to extract the possible feature/opinion pairs. It doesn't recessively iterate through the review corpora to extract more opinions and feature based on those already found. But as pointed out in [1] double propagation alone adds lots of noise (low precision) if the size of the corpora is huge, we have also taken into consideration the sentence patterns to aid the process of feature/opinion extraction. Though our technique adds a lot of noise (false positive) when the size of the corpora is huge but based on findings from [1] it still score over Double Propagation in terms of precision. In [1] where author has provided some performance numbers for DP. For 1000 review sample, the author has calculated the average precision of 73.75%, which is less then our overall precision of 78.00%. Though the datasets are not similar this gives an estimate of performance variation between double propagation and our hybrid technique. A limitation that both Double Propagation and our approach share is that when the size of the corpora is too small [1], both techniques might end up missing some important features.

There are some more techniques used for Feature mining that involve topic modeling and clustering [11]. A serious problem with both these techniques is that although they have no difficulty in finding common features they don’t perform well in case of fine grain feature mining [1].

CHAPTER 3. PROPOSED APPROACH

As mentioned in the previous chapters there are two different insights that can be used to extract semantics from the reviews; Patterns that exist in the sentences and grammatical dependency structures. Our proposed approach involves using both of these techniques in a hybrid fashion to extract the relationships between different features and components mentioned in the product reviews and henceforth used for hierarchy generation. In this section we would go through both of these approaches and propose the algorithm for hierarchy generation.

3.1. Creating component database

A component database would consist of a general set of all the components of a particular product, which serves as an initial knowledge about the product itself. The idea is to have an initial set of feature/components of the product for which the reviews are being considered. It is evident that in most cases the features/components in a review sentence would have NN or NNS POS tagging. Hence, this knowledge can be very useful in creating a component database for our problem. One disadvantage of this approach is that not all the NN and NNS tagged words are features/components. This approach as such adds a lot of noise to the dataset generated. For this purpose nouns below a certain threshold are not included in the CD. After multiple iterations the threshold is found to be in the ballpark of 10 occurrences in the feature corpse. The important point to remember here is that while creating this CD the reviews are not domain specific but rather are the reviews picked up in general. This flexibility is really helpful as CD is not domain specific and can be used while creating hierarchies corresponding to products from different domains.

3.2. *Extracting root features*

The first step in the extraction of the relevant hierarchy is to identify the root feature or parent feature. For such identification the approach taken is to rank all the nouns present in the reviews according to their frequency. The assumption is that the root feature or the parent noun is mentioned more in a set or relevant reviews than any other noun. For identifying the nouns in the reviews and ranking them according to their frequency Stanford Maxent tagger was used.

Algorithm 1. Root feature extraction

---

1. Use Maxent tagger to tag all the words in the review according to the part of speech.
2. Extract all the words with NN or NNS tags.
3. Rank the words according to their frequency.
4. The most frequent one is parent/root feature.

Example:

Input:

I bought this camera for my friend yesterday. According to him to camera is awesome. The size of the camera is small which makes it perfect in terms of portability. The viewfinder works perfectly and takes nice pictures. Overall nice camera.

Pos Tagged Document with ranking:

Camera – 3, Friend – 1, yesterday – 1, size – 1, portability – 1, viewfinder – 1, picture – 1

Output: camera

3.3. Pruning irrelevant sentences

Not all the sentences in a review are the relevant sentences for hierarchy generation. Like in the previous example the review talks about the camera but not all the sentences provide an insight of what the person thinks about the camera's features or about the camera in general. Like the first sentence *"I bought this camera for my friend yesterday"*. In this sentence the reviewer doesn't provide any useful information about what he thinks about the camera. To reduce the complexity of the problem it is better to remove the sentences that don't provide any valid information. For our problem we have used at least *1 adjective and 1 noun as the pruning criteria* meaning we would prune the sentences that don't have at least one pair of nouns and adjective.

Algorithm 2. Removing irrelevant sentences

---

1. Use Maxent tagger to tag all the words in the review sentences
2. Count the occurrence of *JJ, NN* and *NNS* in each of the sentences
3. Remove the sentences that don't have at least one pair to *JJ & NN* or *JJ & NNS*

Example:

Input:

I bought this camera for my friend yesterday. According to him to camera is awesome. The size of the camera is small which makes it perfect in terms of portability. The viewfinder works perfectly and takes nice pictures. Overall nice camera.

POS Tagged Document:

I/PRP bought/VBD this/DT camera/NN for/IN my/PRP$ friend/NN yesterday/NN ./. According/VBG to/TO him/PRP the/DT camera/NN is/VBZ awesome/JJ ./. The/DT size/NN of/IN the/DT camera/NN is/VBZ small/JJ which/WDT makes/VBZ it/PRP perfect/JJ in/IN terms/NNS of/IN portability/NN ./. The/DT viewfinder/NN works/VBZ perfectly/RB and/CC takes/VBZ nice/JJ pictures/NNS ./. Overall/JJ nice/JJ camera/NN ./.

Output:

According to him the camera is awesome. The size of the camera is small which makes it perfect in terms of portability. The viewfinder works perfectly and takes nice pictures. Overall nice camera.

3.4. Generating features

Once the relevant sentences are know the next step in the process is to identify the features/components that are present in the review sentences. There can be two different ways in which a feature is opinionated in a review sentence.

1. **Explicit mention**:

In such cases the feature/component of the product is explicitly mentioned in the review sentence.

Example:

*The viewfinder of the camera was awesome.*In this particular review sentence the component **viewfinder** is explicitly mentioned in the review sentence.

2. **Implicit mention**:

In such cases the feature/component of the product is not explicitly mentioned but can be inferred via the sentence semantics.

Example:

The camera easily fits into the pocket and is really handy. In this case the feature that is being opinionated is **size** but it is not explicitly mentioned in the review sentence itself.

In this paper we are going to handle only those cases in which the features/components are explicitly mentioned in the product reviews. In most cases the feature/components present in the review sentences are *nouns* and hence have *NN* or *NNS*POS tags. In this paper we have discussed two different approaches to identify the features/components present in the review sentences.

3.4.1. Using grammetical dependencies

The grammetical dependencies generated by the Stanford POS tagger is based on the relationships that exists between different words in the sentences. Understanding the relationships between the words in the sentence can provide a meaningful insight about the **Feature/Components – Opinions** relatioships that we are interested in. Also we can use the grammetical dependencies to extract the Parent-Child relationship that exists in the review sentences.

The first part of the feature extraction process deals with extracting the composite features. A composite feature can be defined as a multiworded feature/component. Some of the example of the composite features can be picture quality, lens curvature, camera size and so on. One of the reasons for extracting composite features from the review sentences first is to make sure that in the subsequent run of the algorithm a standalone feature can be easily distinguished from a composite feature. Like if the review talks about picture quality rather that the picture, it makes more sence to include picture quality as the feature rather than picture itself.

Algorithm 3. Composite feature extraction

```
if((SD_Tag == nn && nn_governor is in CD && nn_dependent is not in CD)

|| (SD_Tag == nn && nn_governor not in CD && nn_dependent is in CD)

&& distance between nn_governor and nn_dependent == 0)

then

composite_feature == nn_dependent + nn_governor
```

Once the composite feature extraction is complete we can move ahead and extract the standalone features using next algorithm.

Algorithm 4. Standalone feature extraction

```
if(SD_Tag == nsubj && nsubj_governor == JJ && nsubj_dependent == NN ||
nsubj_dependent == NNS)

    then nsubj_dependent is the feature

if(SD_Tag == nsubj && nsubj_dependent is in CD && any_dobj_dependent not in CD)

    then dobj_dependent is the feature

if(SD_Tag == poss && poss_dependent == root_feature)

    then poss_governor is the feature

if(SD_Tag == nsubj && any_amod_governor not in CD && nsubj_dependent ==
root_feature)

    then amod_governor is the feature
```

The above-mentioned algorithms use grammatical dependency structures for extracting the features from the review sentences. Other then the grammatical dependency structure, some specific sentence patterns are also useful in extracting the features. The next module explains some useful sentence patterns that we have used for this purpose.

3.4.2. Using sentence patterns

In the earlier section we talked about the relevant grammetical dependencies that can be used to extract such relationships from the review sentences. In this section we would discuss some interesting sentence patterns that can be useful in out exploration.

a. Part-of Pattern

This pattern is really helpful in extracting the parent-child relationship that exist in the review sentences. The idea is that each feature/component that is mentioned in the review sentence is related to the parent product in a part-of relation.

Example:

Consider the review sentences

i. The screen quality of the camera is great.

ii. The seats with this couch are really comfortable.

iii. The headphone's cords are really strong.

In all the above mentioned review sentences the parent product is in one way or other related to child feature/component. In first one, the screen quality is connected to the camera by *of* preposition. In second one the seats are connected to couch via *with* and in third the headphones are connected to cords via *'s*. In general it can be said that the parent and child are (in most cases) connected via a prepostition. And such patterns can be really helpful in extracting these relationships from the review sentences.

For our requirements we are interested in some specific patterns [1] in review sentences that are more likely to contain a parent-child relationship or other features.

i. Noun Phrase + Prep + Noun Phrase

Such patterns are useful in extracting the parent-child relationships in the review sentences in which the parent and their corresponding child is related via a single preposition. An example would be "The size of camera is awesome." Its corresponding POS tagged structure being "DT *NN IN NN* VBZ JJ" The pattern NN IN NN is the important one as irrespective of the actual words which are filling in NN, IN and NN the relationship would hold true in most of the cases.

So, in algorithmic terms this can be explained as

Algorithm 5. Generating standalone features using NN IN NN patterns

```
if POS tagged structure contains NN IN DT NN || NN IN NN
   if last NN tagged word is root_feature
         first NN tagged word is feature
    if first NN tagged word is root_feature && second word is preposition
         second NN tagged word is feature
```

b. Composite feature pattern

i. Noun Phrase + Noun Phrase

Such patterns are useful in finding the composite features that might exist in the review sentences. Such composite features are not connected via a preposition hence the earlier mentioned pattern doesn't hold true in such cases.

An example would be "The picture quality of the camera is perfect." Its POS tagged structure boils down to "DT *NN NN* IN DT NN VBZ JJ". Consecutive nouns phrases hold an important place here, as they are in most cases an indicator of composite feature involved.

Consecutive NN tags can be really helpful in extracting composite features using the following algorithm:

Algorithm 6. Extracting composite features using sentence patterns

---

if word POS tag is NN && word + 1 POS tag is NN {

if(word is a feature && word + 1 is not a feature || word is not a feature && word + 1 is a feature)

word + (word +1) is a composite feature }

---

c. Opinionating existing feature pattern

i. Noun Phrase + Verb Phrase + Adjective Phrase + Noun Phrase

Such patterns are useful in extracting the feature in the cases when it is not directly linked to the root feature of the review sentence. A word of caution here is that this pattern might add unnecessary noise to the result as this pattern though exists in the review sentence can easily substitute for a completely different purpose. An example for such patterns can be "This camera has great viewfinder"; "This bed contains soft mattresses."

Algorithm 7. Extracting directly opinioned features

If (POS tagged structure contains NN VBZ JJ NN || NN VBZ JJ NNS && if first NN tagged word is root_feature) then

second NN OR NNS tagged word is feature

3.5. Generating opinions

In the last module we discussed how to extract features from the review sentences using both the grammatical dependency structure and sentence patterns. As the next phase of the project for all the features/components extracted in the previous step we would extract there corresponding opinion words. Once the opinions are extracted there polarity is decided be the list of positive and negative words as provided in [2].

An important thing to note here is that the polarity of the opinion words has only a binary output. Meaning the opinions words would only be categorized into liked or disliked instead of liked, very liked, disliked, very disliked. The degree of opinion is not handled in this paper.

Below mentioned algorithms are the ones used to extract opinions corresponding to the features extracted in the last phase. Just like in last module the opinion extraction is also divided into two phases. In phase one we extract the opinions corresponding to the features extracted using grammatical dependencies; in phase two we use POS tagged sentence structure to extract their corresponding opinions.

3.5.1. Using grammatical dependencies

Grammatical dependency structures are useful in extracting opinions also. Some of the important dependency structures that are useful in our case are nsubj, amod & advmod.

a. Opinion extraction using nsubj

The below mentioned algorithm uses nsubj grammatical dependency for extracting the opinion.

Algorithm 8. Extracting opinions nsubj

---

if (SD_Tag == nsubj && nsubj dependent is feature && nsubj governor is adjective)

then

nsubj governor is opinion for nsubj dependent

An example for this could be "The size of the camera is perfect" which translates to *det(size-2, The-1), nsubj(perfect-7, size-2), det(camera-5, the-4), prep_of(size-2, camera-5 cop(perfect-7, is-6), root(ROOT-0, perfect-7)*

So, the opinion corresponding to feature *size* is extracted as *perfect*.

b. Opinion extraction using amod

The algorithm below uses amod grammatical dependency structure for extracting the opinions.

Algorithm 9. Extracting opinions amod

---

if (SD_Tag == amod && amod governor is feature && amod dependent is adjective)

then

amod dependent is opinion for amod governor

An example for this can be "The camera has great viewfinder" which translates to *det(camera-2, The-1), nsubj(has-3, camera-2), root(ROOT-0, has-3), amod(viewfinder-5, great-4),dobj(has-3, viewfinder-5)*

So, the opinion corresponding to feature *camera* is extracted as *great*.

c. Opinion extraction using advmod & nsubj

The algorithm uses both nsubj and advmod dependency to generate the opinions.

Algorithm 10. Extracting opinions nsubj & advmod

---

if(SD_Tag == nsubj && nsubj dependent is feature && nsubj governor is not adjective)

then

if(SD_Tag == advmod && nsubj governor == advmod governor && advmod dependent is adjective)

then

advmod dependent is opinion for nsubj dependent

An example for this can be "The viewfinder of the camera works perfectly" which translates to *det(viewfinder-2, The-1), nsubj(works-6, viewfinder-2), det(camera-5, the-4), prep_of(viewfinder-2, camera-5), root(ROOT-0, works-6), advmod(works-6, perfectly-7)*

So, the opinion corresponding to feature *camera* is extracted as *perfectly*.

3.5.2. Using closest adjective approach

As the last resort if none of the above mentioned grammatical dependency structures are present in the sentence then the closest adjective is extracted as the possible opinion for the feature [2]. Though this approach is able to extract closest adjective but the extracted adjective might not be related to the actual feature. Because of which this approach can add a lot of noise to the solution. Also in some cases two adjective words might be at same distance from a noun feature. For this we have decided to go with the closest adjective phrase, which succeeds the feature word.

Algorithm 11. Extracting opinions using proximity adjectives

---

if POS tagged word is feature

nearest JJ tagged word is opinion

if two JJ tagged words are at equal proximity then

JJ tagged word which succeeds feature is opinion for it

CHAPTER 4. EXPERIMENTAL RESULTS

The performance of our system was evaluated based on the following aspects.

1. How accurately the system can extract the feature that exists in the review sentences.
2. How accurately the system can extract the opinion that exists in the review sentences.
3. How accurately the system can relate the correct features to correct opinions in the review sentences.

4.1. Data sets

For evaluating the performance of the system we have to come up with a golden data set that would be used as the source of truth for performance evaluation. For our use cases we have decided to evaluate performance against 5 different sets of sentence structure. Golden set for each sentence structure consists of 25 review sentences taken from Amazon.com. The reviews were categorized into 5 different sub categories based on which of the sentence structure they best suited to. An important thing to note here is that the system is not designed to only handle the mentioned structures only. The sentence structures are only used to evaluate the performance of the system. The five categories are mentioned below along with a brief description of what each category means.

4.1.1. Simple sentence structures

A review sentence falls into this category if there are no specific structure like comma separated feature, conjunction separated features etc. present in it. These are the simplest review sentences providing clear insight of person’s opinion about a specific feature in the product.

Examples

a. The viewfinder of the camera is awesome.

b. Very good little camera!

c. This is a great Camera, Love it!

d. Great camera for the price.

e. Excellent little camera

4.1.2. Comma separated features/components

As the name indicated, these review sentences are those in which we have at least one pair of features separated by a comma. A point to note here is that there can be any number of features separated by comma. As long as we have at least one pair the whole review falls into this sentence structure category.

Examples

a. Works great. Easy to use, nice pictures. Easy to upload to the computer.

b. Nice little camera, great when you are outdoor, less weight and compact.

c. It's a great camera. Fun, easy to use and takes good pics! Plus it's pink and it's perfect for a teen!

d. It is easy to use, pretty tough, and takes decent photos.

e. Great viewfinder, lens and size.

4.1.3. Negative semantics structure

Sentences, which have a negative orientation towards the feature present in the review sentence, fall under this category. Mostly but not necessarily these sentences have presence of “not” keyword preceding the opinion work for a feature.

Examples

a. Not a great camera. The viewfinder is not good.

b. The picture quality is not great just small postcard size photos.

c. The camera was good for the first photo and the rest are all just a black or white background

d. This camera is not great for photography.

e. Pathetic camera generates dull pictures.

4.1.4. Conjunction separated features/sentences

These are those review sentences in which we have at least one pair of feature separated by a conjunction. These are similar to the comma-separated sentences but in this case we have conjunctions joining the features being opinionated.

Examples

a. Nice and compact. Takes good photos, just what I needed!

b. High resolution but dull pictures

c. Excellent. Take good picture and videos and it is a user-friendly camera. Easy to use and very durable.

d. Takes great pictures and videos and we love that it's small. Love it

e. Works great. Easy to use, nice pictures. Easy to upload to the computer

4.1.5. Compound sentence structures

Compound sentences are those, which fall under multiple sentence structures. An example can be a review sentence that has both conjunctions and commas as separator for features. Another example can be sentences, which have compound features (like picture quality, lens resolution, build quality etc.) in them.

Examples

a. This viewfinder, lens and picture quality of the camera is perfect.

b. This camera has good flash quality.

c. Great picture quality but low resolution.

d. The picture quality of the camera is perfect.

e. Great build quality.

For each of the sentence structures mentioned above we had 25 examples and each were manually tagged for features/components and their corresponding opinions. In ideal scenario the system should generate results in accordance with this golden data set created.

4.2. Evaluation parameters

For each of the review types discussed above we will calculate following parameters

1. Total accuracy of feature extraction
2. Total accuracy of feature-opinion pair extraction.
3. Total precision
4. Total recall

These parameters would be calculated for all the sentence structure, which will provide us the individual performance of the system on these review types. Based on the performance of the system for individual review sentence structures the overall performance of the system can be calculated. Another important thing to note here is that there is an overlap of the sentence structure meaning a review sentence can belong to more than one sentence structure. This gives a really good understanding of how the system behaves when faced with sentences that do not strictly belong on only one structure.

4.3. Performance evaluation

Table 2. Simple sentence structures

Total features present	34
Total features extracted	40
Correctly identified features	34
Incorrectly identified features	6
Precision	34/40 = 85%
Recall	34/34 = 100%

Table 3. Comma separated features/components

Total features present	65
Total features extracted	76
Correctly identified features	59
Features not identified	6
Incorrectly identified features	17
Precision	65/76 = 85.55%
Recall	59/65 = 90.7%

Table 4. Negative semantics structure

Total features present	34
Total features extracted	41
Correctly identified features	29
Features not identified	5
Incorrectly identified features	12
Precision	34/41 = 82.92%
Recall	29/34 = 85.29%

Table 5. Conjunction separated features/sentences

Total features present	54
Total features extracted	74
Correctly identified features	49
Features not identified	5
Incorrectly identified features	25
Precision	54/74 = 72.97%
Recall	49/54 = 90.70%

Table 6. Compound sentence structures

Total features present	40
Total features extracted	60
Correctly identified features	34
Features not identified	6
Incorrectly identified features	26
Precision	40/60 = 66.66%
Recall	34/40 = 85%

4.4. System performance

Based on the results from the individual sentence structure we can estimate the overall performance of the system.

Table 7. Total system performance

Total features present	34+65+34+54+40=227
Total features extracted	40+76+41+74+60=291
Correctly identified features	34+59+29+49+34=205
Features not identified	0+6+5+5+6=22
Incorrectly identified features	6+17+12+25+26 = 86
Precision	227/291 = 78.00%
Recall	205/227 = 90.30%

In the above-mentioned tables we discussed the performance of the system against different sentence structures. Even though the system is not designed to only handle only these sentence structures, performance evaluation against these different sentence structures gave us a pretty decent understanding of the capabilities of the system. The system performed pretty decently with for all the sentence structures except compound sentence structures where precision was as low 66.66%. The most prominent reason for such low precision can be the high number of variance in which these sentence structures can be formulated. Another reason is the compound sentence structures most of the times don't follow the strict language grammar rules as followed by other sentence structures. Our approach is based on the existence of some specific grammatical pattern in the input review sentence and it's of utmost importance that the formulation of these input review sentence follow the language specific grammar rules. Deviation from those rules would result in Stanford parser generating a wrong grammar tree and hence our system would perform poorly. As rest of the sentence structures i.e. simple sentence structure, comma separated structures, negative semantic structure and conjunction separated structures have a very little room to deviate from the language specific grammatical rules the performance is much decent (precision above 70%).

CHAPTER 5. CONCLUSION & FUTURE WORK

In this paper we explored a different approach to solve the problem of semantic extraction from online reviews. Using a hybrid approach of using grammatical dependencies and pattern matching for extracting the features and also the related opinion words. We also compared our approach with the existing approaches out there most of which involved large-scale machine learning and/or pattern matching. Our approach is unique in the sense that it is mostly agnostic of the initial learning corpora, which is an integral part of approaches using supervised machine learning techniques for extracting the feature/opinion pairs from the review sentences. Using grammatical dependencies and pattern matching in a hybrid fashion helped us extract features and opinion words which we might have missed using either one of these two techniques. We also compared our technique with other techniques in terms of precision and recall and saw how not always but most of the times our approach scored over the other mentioned approaches.

In future we would like to make our approach more robust. As mentioned in the introduction section our approach; though works with multiple sentence types is not totally agnostic of sentence structure. It only handles the scenarios where the opinion word is adjective. In future we would like to extend this to extract and opinion word irrespective of the part of speech it belongs to. Also we are only extracting relationships are one level of depth. Even though most of the real world use cases are solved by extracting only first level of relationships there are some edge cases in which there would be a requirement to extract second level of relationships also. Like “The focal length of the lens of this awesome camera is great” can be a prospective candidate of review sentences where it is important to extract second order of relationships also.

Another aspect, which we would like to explore in future, is the scalability. Current implementation of the algorithms works on a single node and hence is limited by the computation capacity of one machine. In future we would like to explore how to effective use the new distributed computational techniques, probably exploiting hadoop with map-reduce framework to our advantage so that much larger review sentences can be handled effectively.

REFERENCES

[1] Lei Zhang, Bing Liu, Suk Hwan Lim and Eamonn O'Brien-Strain. 2010. Extracting and ranking product features in opinion documents [Internet]. Available from http://www.aclweb.org/anthology/C10-2167

[2] Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews [Internet]. Available from https://www.cs.uic.edu/~liub/publications/kdd04-revSummary.pdf

[3] comScore. 2007. Online consumer-generated reviews have significant impact on offline purchase behavior [Internet]. Available at https://www.comscore.com/Insights/Press-Releases/2007/11/Online-Consumer-Reviews-Impact-Offline-Purchasing-Behavior

[4] John B. Horrigan. 2008. Online Shopping [Internet]. Available at http://www.pewinternet.org/2008/02/13/online-shopping/

[5] Jayashri Khairnar, Mayura Kinikar. 2013. Machine learning algorithms for opinion mining and sentiment classification. Available at http://www.ijsrp.org/research-paper-0613/ijsrp-p18125.pdf

[6] Guang Qiu, Bing Liu, Jiajun Bu, Chun Chen. 2011. Opinion word expansion and target extraction through double propagation. Available at http://dl.acm.org/citation.cfm?id=1970422

[7] B. Pang, L. Lee, and S. Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. Available at http://www.cs.cornell.edu/home/llee/papers/sentiment.pdf

[8] S. Baccianella, A. Esuli, and F. Sebastiani. 2009. Multi-facet rating of product reviews. Available at http://nmis.isti.cnr.it/sebastiani/Publications/ECIR09c.pdf

[9] R. Srivastava, M.P.S. Bhatia, H.K. Srivastava, C.P. Sahu. 2010. Exploiting grammatical dependencies for fine-grained opinion mining. Available at http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5640438&isnumber=5640373

[10] Neha S. Joshi, Suhasini A. Itkat. 2014. A survey on feature level sentiment analysis. Available at http://www.ijcsit.com/docs/Volume%205/vol5issue04/ijcsit20140504135.pdf

[11] Bin Lu, Myle Ott, Claire Cardie and Benjamin Tsou. 2011. Multi-aspect sentiment analysis with topic models. Available at https://www.cs.cornell.edu/home/cardie/papers/masa-sentire-2011.pdf