RESEARCH ARTICLE OPEN ACCESS

Description Based Recommendation System

Dewanshu, Pankaj Gupta, Sahil Awasthi, Shikha Rastogi

Bharati Vidyapeeth's College of Engineering, New Delhi

Abstract

E-commerce is becoming the new normal of buying and selling goods and has been seeing exponential since the past few years. The credit of this rise can be given to the lower prices and an ample of categories of goods across various sections which are accompanied with fast shipping. Customer reviews for products are useful information that is beneficial for a future buyer as he can know the quality of goods from those who have used it. There are ample review tasks. In this paper, we present a recommendation system in which a user will describe the needs of what the user wants, and from that our model will extract the aspects and provide the most relevant products based on the customer's requirement.

1. INTRODUCTION

Online shopping involves buying goods via the internet from different sellers using applications or web browsers. Users can search the product that they wish to buy and then order the product from any listed seller based on their preferences. Preferences here refer to price, reviews, etc. While buying products, customers tend to look at other buyers' reviews before buying the product. Customers are concerned whether the product is a war gun or just a showpiece that looks good on paper. Hence potential buyer tries to find out the experience of past buyer with the interested product. Big e-commerce companies have a significant amount of reviews and it is a tedious task for a customer to go through all the reviews and deduce inferences from those reviews about the quality of the product. Further, there are many features of a product but the reviews are not classified according to the features. In addition to all, increased competition in the digital market provides various options to chose from, thereby making it confusing and tiring task to find the most suitable product.

Opinion mining refers to the process of detecting opinion of user about a particular topic or product or problem. In simple words it's a process of extracting knowledge from the judgement of the user currently using the product [1].

Sentiment analysis is the concept in machine learning that is defined as the process of mining of data, views, reviews, or sentences to predict the emotion of the sentence through the use of natural language processing (NLP). The sentiment involves the positive or negative attitude related to the context. Sentiment analysis helps to identify the statistics of like or dislike of particular product features [2].

Product reviews are mixed up as they are provided by several different users based on their experience for that product and they provide the review by comparing the product from their expectations what they were searching for and what the product offered.

A product has several features to offer and each user wants some of the features based on their usage, there may be a chance where the overall rating of the product is not good to that extent but some features are great compared to the other products available in the market, and those are the features that the user wanted, but due to the overall rating of that product, user neglects that product and go for other alternatives.

For a review, in order to extract features, we need to understand the sentence structure, and to capture inherent relation some set of defined relations are used [3-4].

To find a suitable product one possible option can be described based where the user will describe the needs and model will recommend the products which are most relevant to that description and the products are already been pre-processed based on the previous user's reviews which provide the better recommendations.

2. RELATED WORK

In [5] the author focused on aspect-based sentiment analysis, where single aspects were taken into consideration instead of complete sentences. To identify product aspects association mining was used, and then sentiment score was attached to each aspect by exploiting small seed sets of opinion words, along with antonyms and synonyms. In order to extract additional infrequent product aspects they used newly detected opinion words.

In [6] the author used the idea that for a single review to detect product aspects opinion words can be used and vice versa. In order to identify new opinion words and product aspects collection of opinion words were fused with syntactic dependencies. To detect newfound word polarity, polarities of seed words were accounted. The same polarity was assumed for opinion words expressing a sentiment towards the same aspect.

So far we have discussed solutions that are domain-independent. In addition to this, domain-specific knowledge is also used to improve accuracy.

In [7] the author targeted movie reviews for aspect-based classification, movie-related terms were provided as input (director, producer, main leads, etc.). In addition to this domain-specific terms were feed into the algorithm. As a result, a more accurate sentiment classification was obtained by including domain-specific knowledge.

In [8] new aspect related terms were identified between seed terms and candidate terms by feeding algorithm with product aspect and aspect related terms.

Our approach requires neither labelled examples nor domain-specific knowledge, thus it is domain-independent and has wider applicability.

From a point of view of a user retrieving information about a product from reviews given by other users, considering specific aspects becomes as important as overall product rating. Exploring expression of opinions and sentiments for different aspects is another crucial chore in review analysis [9].

To help users to analyse vast amount of online reviews for certain entity (e.g., mobile camera), it is essential to unveil detailed opinions on various relevant aspects of the entity (e.g., the camera was superb) [10].

A product feature can be described in many ways and its possible that several of them results in same meaning. Terms used to describe features are feature expressions [11].

3. METHODOLOGY

3.1 spaCy Parse Tree

Each review or a sentence is made up of tokens. A token is the smallest unit of a sentence. A token is any word or punctuation mark or number. A combination of tokens forms a sentence.

In a sentence, each token falls under the category of one of the Part-of-Speech and each token is dependent on some other token i.e. token dependency. We will use spaCy for our work to find Part-of-Speech (POS tag) and dependency.

spaCy is a library for advanced natural language processing. spaCy provides a way to convert a sentence into a tree structure where each token is a node in the tree and the connecting edges are the dependency between nodes or tokens. As the

sentence is converted into the tree, each node will have a child or parent or both and we can easily traverse the tree just like a normal tree.

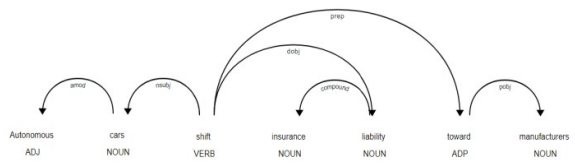

Figure 1 : Dependency Tree

In the above figure for the sentence "Autonomous cars shift insurance liability toward manufacturers" the parse tree is shown where each token i.e. word is the node is assigned a POS tag as well and each token is dependent on the other token and dependency is also shown. We can easily traverse the tree from a token we can easily go to its children if it exists and also its parent if it's not the root.

TEXT	DEP	HEAD TEXT	HEAD POS	CHILDREN
Autonomous	amod	cars	NOUN	
cars	nsubj	shift	VERB	Autonomous
shift	ROOT	shift	VERB	cars, liability, toward
insurance	compound	liability	NOUN	
liability	dobj	shift	VERB	insurance
toward	prep	shift	NOUN	manufacturers
manufacturers	pobj	toward	ADP	

Figure 2 : Tokens Head and Children

This tree will help us to find aspects based on some general rules.

3.2 VADER Sentiment Intensity Analyzer

After extracting the aspects it is important to find the polarity related to that aspect in the review, to find the sentiment we are using VADER Sentiment Intensity Analyzer. VADER stands for Valence Aware Dictionary and Sentiment Reasoner**.** It is a lexicon and rule-based sentiment analysis tool that is precisely attuned to sentiments.

It uses sentiment lexicon, a list of lexical features that are labeled according to their semantic orientation as "positive", "negative" or "neutral". Apart from Positivity and Negativity score VADER also tells us how positive or negative a sentence is based on a scale of [-1,1] where values between [-1,0) denote the sentiment is "Negative" and (0,1] denotes sentiment is "Positive", it also provides decimal values. The main advantage of using VADER, it is constructed from a generalizable, human-curated gold standard sentiment lexicon thus doesn't require any training data. We use the compound score provided by it.

positive sentiment: compound score >= 0.05

neutral sentiment: (compound score > -0.05) and (compound score < 0.05)

negative sentiment: compound score <= -0.05

Figure 3 : Compound Score Sentiment

4. EXPERIMENT

4.1 Dataset

The dataset used in this work is provided by PROMPTCLOUD. This dataset contains 413840 reviews of mobile phones sold on Amazon.com, where each row have 6 fields:-

- Product Name
- Brand Name
- Price
- Rating
- Reviews
- Review Votes.

Product Name is the name of the product, Brand Name is the name of the parent company, Price is the price of the product, Rating is rating ranging between 1-5, Reviews is the description of user experience, Review Votes are the number of people voted the review.

Several rows correspond to a single product. There are total 4410 unique smartphones and total 413840 reviews and each review corresponds to one of the phones.

In our work, we will focus on two columns Product Name, and Reviews. We will create a dictionary where the key corresponds to the Product Name and the value to the key will be the list of the reviews of that particular Product Name. During the creation of

the dictionary, we also perform general Data-Cleaning such as

- Dividing Reviews into Sentences
- Converting Reviews into lower case
- Remove Punctuation Marks
- Remove non-ASCII characters
- Lemmatization

4.2 Procedure

After creating the dictionary with keys as "Product Name" and values as "Reviews", for each review for a particular product we pass it to a function which takes a single review as an argument and returns the aspects present in the reviews (generally nouns) and there sentiment modifiers (adjectives or similar tokens).

The given function converts the review into the tree so that we can easily traverse the tree and find the dependencies of the tokens. We go through each token in the sentence and based on 7 rules (rules which defines the relation of the token with respect to the aspect of the sentiment modifier) and if the token fulfills any of the rules then based on that rule we get the aspect and its corresponding sentiment modifier.

We represent 'M' as the **Sentiment Modifier** and 'A' as the **Aspect**.

Rule 1: M is the child of A with a relationship of amod.

Rule 2: A is a child of a token with the relationship of nsubj, while M is a child of the same token with the relationship of dobj.

Rule 3: A is a child of a token with the relationship of nsubj, while M is the child of the same token with the relationship of acomp.

Rule 4: A is a child of a token with the relationship of nsubjpass, while M is a child of the same token with the relationship of advmod.

Rule 5: A is a child of M with the relationship of nsubj, while M has a child with the relationship of cop.

Rule 6: M is the interjection (POS tag), while A is the child of M with relationship nsubj.

Rule 7: A is a child of a token with the relationship of nsubj, while M is a child of the same token with the relationship of attr.

Negation is an important concern in sentiment analysis. One simple approach is cluding flipping the sentiment of a word when the word is closely located to "not" [12].

Our function also checks and handles for the negation with each rule i.e. with the presence of keywords such as "not" reverses the polarity of the sentence. E.g. "good" is Positive while "not good" is Negative whereas "bad" is Negative while "not bad" is Positive.

After passing all the reviews from this function it provides all the aspects and corresponding sentiment modifiers. We can easily check the polarity of these aspects by their sentiment modifiers with the help of VADER SIA (Sentiment Intensity Analyzer).

We consider the sentiments as Positive whose polarity_score is> 0.05 and Negative whose polarity_score is < -0.05 and neglecting the neutral ones.

We will create a final dictionary for every product, where we add the positive and negative feature count for a particular product which will help in determining the overall sentiment of the users related to that feature (aspect) for that particular product.

Till now pre-processing is done. From the user's reviews, we extracted the aspects and their related sentiments and created the dictionary.

Now we will ask the user for the input, input will be in the form of text where the user will provide a description of user requirements and we will again extract the aspects from that text based on those 7 rules which are discussed above.

Then for every aspect (feature) we will determine the polarity of that aspect and now we will traverse the dictionary which we have created earlier and for every product, we will perform the following operations:-

- If the user-specific feature is present in that product or not
- If that feature is not present we check for other features, if no feature left this product will not be recommended to the user
- If present we will check for the overall sentiment of that feature for that particular product (we are recommending product only if the feature falls above 80% i.e. for Positive: Total number of Positive Reviews for that feature / Total number of Positive + Negative Reviews for that feature >= 0.8 or similar for Negative as well).
- If the user-specific sentiment is similar to that of other user's recommendations for that feature we will recommend this product. (i.e. 85% positive reviews for that particular aspect and our user also wants that aspect in positive aspect we will add it to our list.)

5. RESULT

When the user provides the description our model will display the aspect and their sentiment what our model analyzed for that aspect to the user, and if there is any ambiguity the user can simply check the description, now our model as discussed above will recommend the products, the recommended products will be displayed in the decreasing order, decreasing order here means the number of aspects the product is fulfilling and also displaying the aspects with every product.

```
Enter your requirements :-
I am searching for a phone with awesome camera, great sound and
amazing display
camera -> awesome -> Positive
sound -> great -> Positive
display -> amazing -> Positive
```

	product	weight	features
0	5.5" Unlocked Smartphone Dual Sim Quad Core-JU...	3	(camera, sound, display)
1	5.5" Cell Phone Unlocked Dual Sim Quad Core-JU...	3	(camera, sound, display)
2	6.0" Android 5.1 Phones Unlocked Dual Sim Quad...	3	(camera, sound, display)
3	ALCATEL OneTouch Idol 3 Global Unlocked 4G LTE...	3	(camera, sound, display)
4	ALCATEL OneTouch Idol 3 Global Unlocked 4G LTE...	3	(camera, sound, display)
5	5.5" Phone Unlocked Dual Sim Quad Core-JUNING ...	3	(camera, sound, display)
6	[XMAS DEAL] Jethro [SC118] Simple Unlocked Qua...	3	(camera, sound, display)
7	5.5" Android Cellphone Unlocked Dual Sim Quad ...	2	(camera, sound)
8	5.5" Cell Phones Unlocked 3G/GSM Dual Sim Andr...	2	(camera, display)
9	Alcatel Idol 4S - Factory Unlocked Phone - Bla...	2	(sound, display)
10	Android 5.1 Smartphone Unlocked 5.5" Android 5...	2	(camera, sound)

Figure 4 : Positive Recommendations

As the input text only contains the positive modifiers of the possible aspects.

```
Enter your requirements :-
enlist phones with bad display and terrible battery
display -> bad -> Negative
battery -> terrible -> Negative
```

	product	weight	features
0	4 Inch Touch Screen Cell Phone Unlocked, Andro...	1	(display)
1	5.5" Unlocked GSM Cell Phones Android 5.1 MTK6...	1	(battery)
2	5.5" Unlocked GSM Phones Android 5.1 MTK6580 Q...	1	(battery)
3	5.5" Unlocked GSM Smartphones Android 5.1 MTK6...	1	(battery)
4	5530 XpressMusic Cell Phone w/ Games	1	(battery)
5	Apple a1549 iPhone 6 64GB T-Mobile (silver)	1	(battery)
6	Apple iPhone 3G Black, 16GB	1	(battery)

Figure 5: Negative Recommendations

As the input text only contains the negative modifiers of the possible aspects.

```
Enter your requirements :-
show phones where camera is awesome but sound is poor
camera -> awesome -> Positive
sound -> poor -> Negative
```

	product	weight	features
0	"CLEAR CLEAN ESN" Sprint EPIC 4G Galaxy SPH-D7...	1	(camera)
1	"Nokia Asha 302 Unlocked GSM Phone with 3.2MP ...	1	(camera)
2	Apple iPhone 4 16GB (Black) - AT&T	1	(sound)
3	Apple iPhone 3GS 8GB Black Factory Unlocked / ...	1	(camera)
4	Apple A1533 Unlocked iPhone 5S Smart Phone, 16...	1	(camera)
5	Android Star N7100 Unlocked 1.4 Ghz Dual Core ...	1	(camera)
6	Android 5.1 Smartphone Unlocked 5.5" Android 5...	1	(camera)
7	Alcatel PIXI 4 (6") LTE unlocked smartphone 16...	1	(camera)
8	Alcatel PIXI 4 (5) Unlocked Phone - (Black)	1	(camera)
9	ALCATEL OneTouch Pop 3 Global Unlocked 4G LTE ...	1	(camera)

Figure 6: Mixed Recommendations

As the input text contains both positive and negative modifiers of possible aspects.

6. CONCLUSION

Our model can provide the recommendations based on the textual description provided by the user, for the processing it only needs the reviews with corresponding Product Name, and it will extract the aspects based on general rules and also the overall sentiment of that aspect for the particular product, and for the recommendation, it will check for the user-specific aspects in the Products list. This model is domain-independent it can be helpful for the users to find their suitable product/services. This method of recommendation can be applied not only to the E-commerce but also to other areas as well, like Tourism, Restaurants, Real Estates, and several other areas.

REFERENCES

[1] Blei D. M., Ng A. Y., Jordan M. I., Latent dirichlet allocation. Journal of Machine Learning research, 3:993–1022, 2003.

[2] Ganu G., Elhadad N., Marian A., Beyond the stars: Improving rating predictions using review text content. In Proceedings of the 12th International Workshop on the Web and Databases, 2009.

[3] Brody S., Elhadad N., An unsupervised aspect-sentiment model for online reviews. In Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pages804–812. Association for Computational Linguistics, 2010.

[4] De Marnee M.-C., Mac Cartney B., Manning C. D., Generating typed dependency parses from phrase structure parses. In Proceedings of LREC, volume 6, pages 449–454, 2006.

[5] Hu M., Liu B., Mining and summarizing customer reviews. In Proceedings of the tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 168–177. ACM, 2004.

[6] Qiu G., Liu B., Bu J., Chen C., Expanding domain sentiment lexicon through double propagation. In Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence, volume 9, pages 1199–1204, 2009.

[7] Thet T. T., Na J.-C., Khoo C. S., Aspect-based sentiment analysis of movie reviews on discussion boards. Journal of Information Science, 36(6):823–848, 2010.

[8] Zhu J., Wang H., Zhu M., Tsou B. K., Ma M., Aspect-based opinion polling from customer reviews. IEEE Transactions on Affective Computing, 2(1):37–49, 2011.

[9] Jo Y., Oh A. H., Aspect and sentiment unification model for online review analysis. In Proceedings of the fourth ACM International conference on Web search and data mining, pages 815–824. ACM, 2011.

[10] Wang H., Lu Y., Zhai C., Latent aspect rating analysis without aspect keyword supervision. In Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 618–626. ACM, 2011.

[11] Zhai Z., Liu B., Xu H., Jia P., Clustering product features for opinion mining. In Proceedings of the fourth ACM international conference on Web search and data mining, pages 347–354. ACM, 2011.

[12] K. Eguchi and V. Lavrenko. Sentiment retrieval using generative models. In Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, pages 345–354,

Sydney, Australia, July 2006. Association for Computational Linguistics.

