Recommendation of High Quality Representative Reviews in e-commerce

Debanjan Paul
Indian Institute of Technology
Kharagpur
debanjan0910@gmail.com

Sudeshna Sarkar
Indian Institute of Technology
Kharagpur
sudeshna@cse.iitkgp.ernet.in

Muthusamy Chelliah
Flipkart Internet Pvt Ltd
Bangalore
muthusamy.c@flipkart.com

Chetan Kalyan
Flipkart Internet Pvt Ltd
Bangalore
chetan.k@flipkart.com

Prajit Prashant Sinai Nadkarni
Flipkart Internet Pvt Ltd
Bangalore
prajit.pn@flipkart.com

ABSTRACT

Many users of ecommerce portals commonly use customer reviews for making purchase decisions. But a product may have tens or hundreds of diverse reviews leading to information overload on the customer. The main objective of our work is to develop a recommendation system to recommend a subset of reviews that have high content score and good coverage over different aspects of the product along with their associated sentiments. We address the challenge which arises due to the fact that similar aspects are mentioned in different reviews using different natural language expressions. We use vector representations to identify mentions of similar aspects and map them with aspects mentioned in product features specifications. Review helpfulness score may act as a proxy for the quality of reviews, but new reviews do not have any helpfulness score. We address the cold start problem by using a dynamic convolutional neural network to estimate the quality score from review content. The system is evaluated on datasets from Amazon and Flipkart and is found to be more effective than the competing methods.

KEYWORDS

Review Recommendation; Content scoring; Product aspect; E-commerce

1 INTRODUCTION

Most e-commerce portals facilitate reviews and ratings by the consumers. The reviews may discuss different aspects of the product and may contain different opinions about the product or the various aspects of the product. Such reviews form an important source of information for potential buyers who use the reviews to learn about the product features and the reviewers' opinions on them and use this information to make more informed purchase decisions. However, having a large number of reviews also poses a potential challenge for the user if she has to process each review and extract useful information from them. The e-commerce system may therefore facilitate the user by recommending a small subset of reviews to the user. This subset should ideally be representative of the entire set of reviews and cover different aspects of the product along with the right balance of positive and negative opinion on them. Many e-commerce websites facilitate users to rate the review by other users by providing helpfulness score. This helpfulness score can be used as an estimate of the review quality. However, this suffers from cold start problem since new reviews do not have associated ratings. We have addressed this challenge by training a neural network to predict the helpfulness score of a review from its content. We have incorporated the representative subset computation along with the use of quality scores in selecting a subset of reviews.

2 RELATED WORK

[1] provides a survey on work that extract product aspects and their associated opinions from user reviews of products. Past works have addressed the extraction of representative summaries from review texts ([15], [4], [2], [3], [6], [14], [17], [22]). The main aim of these methods is to create a summary of the reviews of a product that is both comprehensive and statistical in nature, representing both the positive and negative sentiments on different product aspects. But these summaries lack the linguistic structure of reviews written by real users. It has been found that users give more importance to reviews written by actual customers who have used the product than some automatically generated statistical summaries while reading reviews of a product in order to make a purchase decision.

In order to recommend a set of reviews written by real users instead of statistical summaries, some methods have been proposed by ([5],[8],[19],[21]) to assign a score to each review based on helpfulness of the review, the number of aspects it covers and other parameters. These methods then produce ranked lists of reviews based on these parameters and recommend the top k reviews to the users.

But the recommended set of reviews may ignore the less popular aspects of a product. For example, all the top reviews of a mobile phone may be focused on a limited number of attributes like its camera specification and processor speed but these reviews may not address attributes like screen size or look. In order to address the above issues, another set of approaches ([10], [18]) deal with the method of representing a subset of reviews that cover as many product aspects as possible and both positive and negative opinion

RecSys '17, August 27–31, 2017, Como, Italy.
© 2017 ACM. 978-1-4503-4652-8/17/08...$15.00
DOI: http://dx.doi.org/10.1145/3109859.3109901

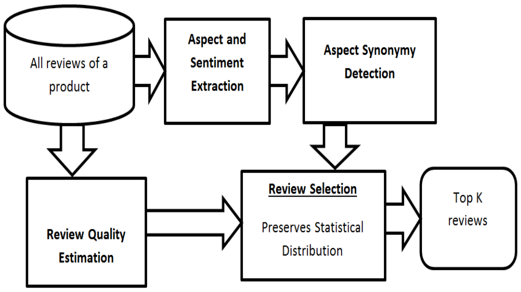
All reviews of a product
Aspect and Sentiment Extraction
Aspect Synonymy Detection
Review Quality Estimation
Review Selection
Preserves Statistical Distribution
Top K reviews

Figure 1: System Overview

sentiments associated with these aspects. The subset of reviews selected by these methods are able to represent a diverse set of aspects and opinions. However, they fail to accurately capture the proportion of opinions in the underlying corpus. This has been addressed by the method proposed by ([9]) that recommends a subset of reviews for a product which maintains the statistical distribution of product aspects along with their associated sentiments on the entire review set.

3 MOTIVATION

We wish to develop a recommendation system to recommend a small subset of reviews to the users which covers different product aspects and sentiments. We find that similar aspects are expressed in different natural language expressions in different reviews. For example, a review may speak about picture quality of a mobile phone, whereas another review may speak about the camera specifications of the same phone. However, we should be able to recognize that they are talking about the same aspect.

Most of the review recommendation systems ignore the textual content of the review and do not use review quality for recommendation. We believe that the quality of the textual content of the review should be used for review recommendation. [20] has used review helpfulness score to estimate review quality. But new reviews do not have any helpfulness score. In order to address the cold start problem, we want to develop a model to estimate the quality score of the reviews, even if they do not have any helpfulness score.

4 PROPOSED METHODOLOGY

We wish to develop a recommendation system to recommend a subset of k high quality representative reviews. The system architecture is shown in figure 1. The *Aspect and Sentiment extraction* module extracts aspect phrases (and corresponding sentiments) from review texts. It is discussed in section 4.1. Similar aspects can be mentioned by using different natural language expressions. The *Aspect Synonymy Detection* module identifies phrases that correspond to similar aspects. This is explained in section 4.2. The *Review Quality Estimation* module scores a review and is discussed in section 4.3. Section 4.4 combines our quality estimation module with the Characteristic Review Selection algorithm from [9] in order to recommend a subset of reviews to the users. Apart from recommending high quality reviews, our recommendation system also maintains the statistical distribution of product aspects along with their associated sentiments.

4.1 Aspect and Sentiment Extraction

The main task of our aspect extraction phase is to extract important aspects mentioned in the reviews along with their corresponding opinion sentiments. We extract the product aspects and opinion associated with them in a review using the double propagation method proposed by [16]. The output contains tuples of the form <review id, aspect, sentiment polarity> (for example, <1,photo quality,+>). Each aspect is a natural language phrase extracted from a review.

4.2 Aspect Synonymy Detection

Similar aspects may be mentioned using different natural language expressions in various reviews (e.g. "camera", "photo quality" and "picture quality" refer to the same product aspect "camera") and are treated as different product aspects. The main aim of synonymy detection is to identify all the similar aspects expressed in different natural language phrases and merge them together into a single aspect.

We make use of product catalogues to identify a standard set of features or aspects for each product type and try to map the different aspect phrases to one of the standard catalogue aspects. For this task, we have used the catalogue of the corresponding category from Flipkart though we have applied this to data from Amazon.

Word Vectors. We link the aspects obtained from textual expressions to the catalogue aspects by computing semantic similarity between them. We use vector representation of words to compute the semantic similarity of different aspect phrases. We train our model on the publicly available Amazon dataset ([11], [13]) for our experiment.

Merging similar aspects together . The input to this phase are the aspect phrases output from *Aspect and Synonymy Extraction* (denoted by A) and the catalogue aspects denoted by C. Algorithm 1 gives the details of Merge (A, C), which maps the extracted aspects from each review in set A with a particular catalogue aspect in set C. For each aspect $extrctd\_aspect$ in set A, if the similarity between $extrctd\_aspect$ and catalogue aspect c is maximum and is greater than a minimum threshold value $epsilon$, then $extrctd\_aspect$ is mapped to c in C. We have used $epsilon = 0.7$. The function $word2vec\_sim(extrctd\_aspect, c)$ in line 9 returns the semantic similarity between vector representation of aspect word $extrctd\_aspect$ and catalogue aspect c. Given two aspect words $extrctd\_aspect$ and catalogue aspect c, we obtain their vectors $\theta_{\text{extrctd\_aspect}}$ and θ_c from the pre-trained word vectors and we use cosine similarity to calculate the semantic similarity.

Comparing of results before and after merging similar aspect phrases. The percentage distribution of 10 out of the 35 aspects obtained from 125 reviews of "Nokia 216 (Black)" are shown in Table 1. We notice that many attributes like "audio", "sound", "speaker" are treated as distinct though they refer to the same feature of the phone. The results obtained after merging similar aspects are shown

Camera	Support	Display	Touch Screen	Price	Bluetooth Headset	Picture	Packaging	OS	Smartphone	...
10%	9%	7%	7%	6%	6%	5%	5%	4%	4%	

Table 1: Results before merging similar aspects. We obtained 35 aspects (top 10 are shown here) from 125 reviews of "Nokia 216 (Black)"". Other aspects are : Audio Quality,Video Recording, Sound Quality, photo, RAM, memory, battery life, clock speed, frequency, sim, network, internet, model, ROM, resolution, pixel, LED, button, power, configuration, processing, color, thing, formats, disk.

Algorithm 1 Merge_Similar_Aspects

```
procedure MERGE($A, C$)
Input: Extracted aspect set $A$ and catalogue aspect set $C$
Output: Aspect (and corresponding opinion) distribution for each review
    for each tuple $a \in A$ do
        $review\_id \leftarrow a[1]$
        $extrctd\_aspect \leftarrow a[2]$
        $opn\_pol \leftarrow a[3]$
        $max\_sim \leftarrow \Phi$
        for each aspect $c \in C$ do
            $sim \leftarrow word2vec\_sim(extrctd\_aspect, c)$
            if $sim > max\_sim$ then
                $max\_sim \leftarrow sim$
                $ctlg\_aspect \leftarrow c$
            end if
        end for
        if $max\_sim > \epsilon$ then
            if $opn\_pol$ is positive then
                map $extrctd\_aspect$ to $ctlg\_aspect^+$
            else
                map $extrctd\_aspect$ to $ctlg\_aspect^-$
            end if
        end if
    end for
```

General	Display	Procressor	Memory	Camera	Connectivity	Multimedia	Charging
56%	47%	16%	6%	29%	49%	34%	36%

Table 2: Results after merging similar aspects. 29 out of 35 aspects obtained from 125 reviews of "Nokia 216 (Black)" have been mapped to 8 catalogue aspects.

Method	Percentage	Average rank
CRS	15%	2.175
CNN	25%	2.2125
COMB	60%	1.6125

Table 3: Comparison of three methods as obtained by human annotation

in Table 2. We have mapped 29 out of 35 initial aspects to 8 catalogue aspects, which results in a better percentage distribution of aspects.

Algorithms	Transactions	PPV
DEFAULT	1777	313776
COMB	1781	308782

Table 4: Comparison of COMB method against default recommendation system at Flipkart.com. Product-page visit (PPV) is defined as the number of users visiting a particular product page. Transactions (Txns) is defined as the number of users who bought the product after visiting the product page.

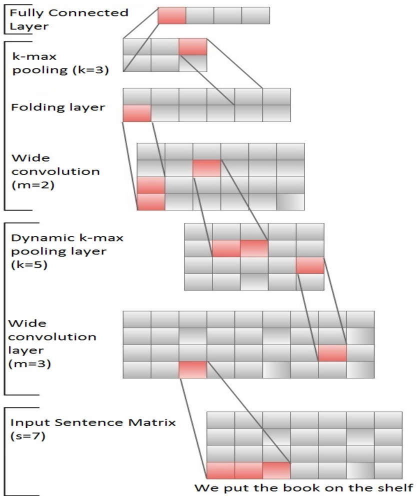
Fully Connected Layer
k-max pooling (k=3)
Folding layer
Wide convolution (m=2)
Dynamic k-max pooling layer (k=5)
Wide convolution layer (m=3)
Input Sentence Matrix (s=7)
We put the book on the shelf

Figure 2: Dynamic convolutional neural network model for estimating review quality

4.3 Predicting review quality score from review content

We use a dynamic convolutional neural network from [7] to predict the quality score of reviews. [7] proposed this model and used this for sentiment prediction and question classification. We have used the model for estimating review quality score from review texts. The architecture is described below.

The **Input Layer** contains the reviews for a particular item. Each word in a review sentence of length s is embedded as $w_i \in \mathbb{R}^d$ corresponding to the i th position of the word in the sentence. Thus, each review is represented as a two dimensional matrix $S \in \mathbb{R}^{d \times s}$ which is given as input to the convolutional layer.

Convolutional Layer : A convolution matrix of weights $m \in \mathbb{R}^{d \times m}$ is applied to the input matrix S to obtain the next layer. We have applied one dimensional wide convolution operation to each row in the input matrix S. The resulting matrix obtained denoted by C has dimensions $d \times (s + m - 1)$.

Dynamic k-max pooling : This layer selects the k most active features from the matrix C. k is determined dynamically as a function of the length of the sentence and depth of the network i.e

$$k_l = \max(k_{\text{top}}, (L - l) \times s/L)$$

where l is the number of current convolutional layers to which the pooling is applied and L is the total number of convolutional layers in the network.

The operation described above consists of one convolution layer followed by one dynamic k-max pooling layer to obtain a first order feature map. These operations are repeated in order to obtain a higher order feature map and a network of increasing size. We have repeated the above operations 2 times in order to obtain a second order feature map and a two layer network architecture.

Folding : After a convolution operation and before dynamic pooling operation, folding is applied to reduce the dimension of the matrix. The elements in two rows of the matrix are added component wise in order to obtain a reduced matrix of dimension $d/2$.

Fully Connected Layer : After extracting the multiple features representing each sentence, the features are passed to the fully connected softmax layer where the output is probabilistically distributed and calculated using multinomial regression technique.

Training datasets and parameters. We have used 2 million reviews from six different product domains (electronics, books, shoes, musical instruments and gourmet food products) in Amazon dataset ([12], [11], [13]) to train the model. Filter sizes of 10 and 7 have been used with 6 filters in the first convolutional layer and 12 filters in the second convolutional layer. Dropout value of 0.5 has been used in the fully connected layer.

4.4 Review Selection

The main aim of this module is to recommend a subset of reviews with high content score and a proper and representative coverage of the product aspects, issues and sentiments. The ranked review list from the previous module is the input to this phase. We have used a greedy iterative algorithm from [9] for selecting a subset of reviews to have the same statistical distribution of aspects (and corresponding opinions) as that of the entire review set for the product. This algorithm greedily selects one review at a time. That review is selected next so that its inclusion makes the recommended set better representative of the entire review set.

For example, let R be the entire review set for a particular product. Let S_k be the subset of reviews which has already been formed by k iterations of the algorithm. The algorithm picks up a review $r^{\mathrm{k+1}}$ in the $k + 1$ th iteration such that $S_k \cup r^{\mathrm{k+1}}$ has proportion of aspects and opinions closest to that of R.

5 EXPERIMENTAL RESULTS

5.1 Datasets

We have used reviews from the publicly available Amazon dataset ([12], [11], [13]). It includes 2 million products and 35 million reviews reviewed by 6 million users. Each review item include product title, user information, review rating, and a plaintext review. We have chosen reviews from popular product domains like Cell phone and electronics products due to the availability of a large number of products and reviews written by the users in these domains.

5.2 Comparison with existing methods

We consider the recommendation of three sets of reviews for eight items based on the three methods below:
1. **CRS**: Recommends a subset of reviews based on Characteristic Review Selection (CRS) algorithm suggested by [9].
2. **CNN**: Recommends a subset of reviews based on the quality score of each review as estimated by the DCNN model.
3.**COMB**: Recommends a subset of reviews by our combined approach as discussed in section 4.

For each of the above 8 items, we have recommended a subset of 5 reviews using the above methods. We have used 10 human annotators for this. The annotators were given the three sets of reviews (in a randomized order) and were asked to rank the methods from 1 (best) to 3 (worst). For each of the 8 items, we have given the percentage of both positive and negative opinions on each aspects of the product. This helps them to get an overall idea about the product quality and judge the representativeness of the recommended subset of reviews in a better way.

COMB method has clearly outperformed the other methods as shown in Table 3. It has been declared as the best method in 60% of the reports obtained from human annotators. Moreover, COMB approach has got a lower rank than the competing methods.

5.3 Live Evaluation on Flipkart based on Transactions

We have compared the CNN and COMB review recommendation method against the default review recommendation system implemented at a popular e-commerce website (Flipkart.com). We considered reviews from 10 random products in the mobile phone category. We have considered the top 100 sorted reviews recommended by COMB method. These reviews are shown in the most helpful tab of Flipkart product page while all other tabs shows the reviews recommended by Flipkart default algorithm. This experiment is run live for a couple of days in order to understand the impact of our methods on live users compared to their default algorithm. COMB method has clearly performed better than default recommendation system of Flipkart as seen in Table 4. The Flipkart default recommendation system resulted in 1777 product transactions out of 313776 product-page visit for 25 products from the mobile phone category. The COMB method has shown an increase in product transactions to 1781 conversions out of 308782 product-page visit, thereby increasing product sales by 1.85%.

6 CONCLUSION

We have reported the results of one preliminary evaluation of our review recommendation system. The preliminary results are encouraging but we need to improve the method for estimating review helpfulness.

REFERENCES

[1] Li Chen, Guanliang Chen, and Feng Wang. 2015. Recommender systems based on user reviews: the state of the art. *User Modeling and User-Adapted Interaction* 25, 2 (2015), 99.

[2] Kavita Ganesan, ChengXiang Zhai, and Jiawei Han. 2010. Opinosis: a graph-based approach to abstractive summarization of highly redundant opinions. In *Proceedings of the 23rd international conference on computational linguistics*. Association for Computational Linguistics, 340–348.

[3] Kavita Ganesan, ChengXiang Zhai, and Evelyne Viegas. 2012. Micropinion generation: an unsupervised approach to generating ultra-concise summaries of opinions. In *Proceedings of the 21st international conference on World Wide Web*. ACM, 869–878.

[4] Shima Gerani, Yashar Mehdad, Giuseppe Carenini, Raymond T Ng, and Bita Nejat. 2014. Abstractive Summarization of Product Reviews Using Discourse Structure.. In *EMNLP*. 1602–1613.

[5] Anindya Ghose and Panagiotis G Ipeirotis. 2007. Designing novel review ranking systems: predicting the usefulness and impact of reviews. In *Proceedings of the ninth international conference on Electronic commerce*. ACM, 303–310.

[6] Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 168–177.

[7] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188* (2014).

[8] Soo-Min Kim, Patrick Pantel, Tim Chklovski, and Marco Pennacchiotti. 2006. Automatically assessing review helpfulness. In *Proceedings of the 2006 Conference on empirical methods in natural language processing*. Association for Computational Linguistics, 423–430.

[9] Theodoros Lappas, Mark Crovella, and Evimaria Terzi. 2012. Selecting a characteristic set of reviews. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 832–840.

[10] Theodoros Lappas and Dimitrios Gunopulos. 2010. Efficient confident search in large review corpora. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 195–210.

[11] Julian McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems*. ACM, 165–172.

[12] Julian McAuley, Rahul Pandey, and Jure Leskovec. 2015. Inferring networks of substitutable and complementary products. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 785–794.

[13] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 43–52.

[14] Xinfan Meng and Houfeng Wang. 2009. Mining user reviews: from specification to summarization. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*. Association for Computational Linguistics, 177–180.

[15] Thanh-Son Nguyen, Hady W Lauw, and Panayiotis Tsaparas. 2015. Review synthesis for micro-review summarization. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*. ACM, 169–178.

[16] Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. 2011. Opinion word expansion and target extraction through double propagation. *Computational linguistics* 37, 1 (2011), 9–27.

[17] Kazutaka Shimada, Ryosuke Tadano, and Tsutomu Endo. 2011. Multi-aspects review summarization with objective information. *Procedia-Social and Behavioral Sciences* 27 (2011), 140–149.

[18] Panayiotis Tsaparas, Alexandros Ntoulas, and Evimaria Terzi. 2011. Selecting a comprehensive set of reviews. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 168–176.

[19] Oren Tsur and Ari Rappoport. 2009. RevRank: A Fully Unsupervised Algorithm for Selecting the Most Helpful Book Reviews.. In *ICWSM*.

[20] Nana Xu, Hongyan Liu, Jiawei Chen, Jun He, and Xiaoyong Du. 2014. Selecting a representative set of diverse quality reviews automatically. In *Proceedings of the 2014 SIAM International Conference on Data Mining*. SIAM, 488–496.

[21] Zhu Zhang and Balaji Varadarajan. 2006. Utility scoring of product reviews. In *Proceedings of the 15th ACM international conference on Information and knowledge management*. ACM, 51–57.

[22] Li Zhuang, Feng Jing, and Xiao-Yan Zhu. 2006. Movie review mining and summarization. In *Proceedings of the 15th ACM international conference on Information and knowledge management*. ACM, 43–50.