Makeup reputation engine based on Social Network

ChiHung Hsieh, YiChen Hsueh

***Abstract*—E-Commence is becoming more and more popular , the number of customer reviews that a product receives grows rapidly. Popular products can get hundreds or even thousands reviews, however, some special products get none reviews if they are not launched on e-commence. This makes it difficult for a potential customer to choose them. Therefore, we proposed the method which can help customers to find their preferred product more efficiency.In this paper, we present Makeup Reviews Engine, an real-time product review analysis system that performs a review time series plot for Sephora and Twitter and predicted rating based on Twitter reviews. We also summarize aspects of product reviews. We evaluate the accuracy of the system through RMSE (root-mean-square error) and the approach in experiment. Result shows that predicted rating in Twitter is really closed to Sephora product rating. It also successfully extracts product summary from product review.**

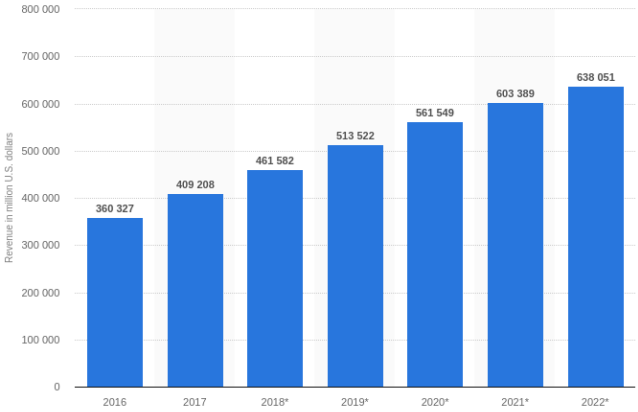
I. Introduction

The rapidly expanding e-commerce has facilitated consumers to purchase products online. More online retailer encourage customer to give the product review. As the e-commerce keep increasing,Fig.1 shows the number of customer review keep increasing. It do provide the customer good tool. However, huge collection customer reviews also bring in the some issue. In our approach is going to answer two question related to huge quantity review. First question we want to answer is, if there's no exactly the product sales on platform. More common situation is there are only few user give the reviews. The average rating will be vary a lot by few user. In many case, the user review can be manipulated by seller. Another question in the report want to answer is there's too many reviews about the product. Most the popular product will in the end achieve relative high rating. [1] Mentioned There are several reasons to believe these correlations may exist. First, people may be more inclined to vote on extreme reviews, since these are more likely to generate an opinion from the reader. Following similar reasoning, people may also be more likely to vote on reviews that are longer because the additional content has more potential to generate a reaction from the reader. Even the number of votes may be correlated with likelihood to vote due to a "bandwagon" effect. In this case, the rating can't work properly for reviewing product. In this report, we build framework for summarize the review. Based on term frequency and pos tag. We also apply liu and hu [2].

II. Related Work

Compared to our projects, [3] uses the same data source to lower the influence of spammer in a social network. The similarity is they use Twitters official policy of suspending accounts to check is it a spammer or not. Also, they examined url posted by users using URL shortening services and checked

Fig. 1. Retail e-commerce sales in the United States from 2016 to 2022 (in million U.S. dollars)



whether these URLs are the presence of blacklisted URLs or not. The difference is we would more focus on a sponsored post, which is not real makeup review. We would classify this kind of poster as a spammer.

We would use the method of extracting important aspects of a product from [4]. The goal of [4] is to automatically identify important product aspects from online consumer reviews. There is three step to achieve their goal. Doing aspect identification first by using Stanford parser and extracting the phrases to identify aspects of the candidates. Then they utilized these aspects to train SVM sentiment classifier and select sentiment terms. Finally, they use alternating optimization technique and compute the score for each aspect by integrating all the reviews. The difference is we would focus more on the reviews having a lower rating. Since most of the people would not like to give a too low rating, otherwise they would have a really bad experience with that product.

Based on data mining method, we would use NLTK package, and the concept is similar to [5]. The goal of [5] is solving product feature problem when parsing reviews. They introduce the concept of phrase dependency parsing and propose an approach to construct it.

[6] also provide a new framework for score news and web via a network. The role for determine news can be replaced by make-up product. Web score can be replaced as Twitter users. The difficult to apply the framework for 2 reasons: (A)We need initial good and bad users group. In the [6], they provide the reliable web by empirical method, bad webs from web certificate service. (B)Build network. Unlike citing web, cite the product can have so many different ways. Abbreviation, the nickname isn't easy to track. This two reason we might need to solve before we applying the same framework.

Based on twitter/Sephora reviews and time slot, we can

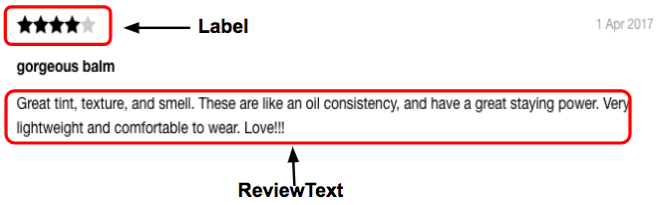


Fig. 2. Review Format

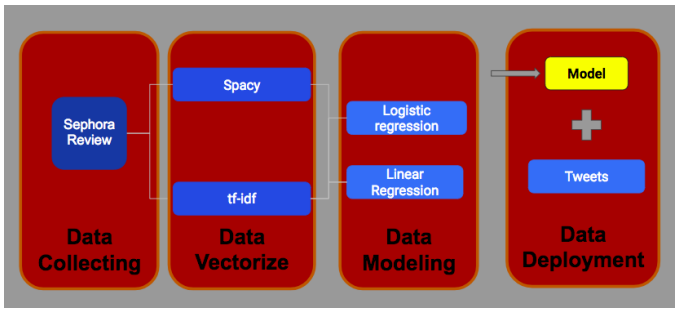


Fig. 3. Makeup Reviews Engine

do trend analysis to predict potential hottest makeup. The method is similar to [7]. The goal is using moving average convergence-divergence to predict the trends of the topic on Twitter. They also take three main factor into account, key users, key words and topic interaction, which utilize chi-square to examine.

III. Approach

Our approach performs in two ways to mine customer reviews: (1) product rating (2) auto summary

A. Rating

Fig.3 gives the architectural overview of our Makeup Review Engine. Scraped Sephora product and review first to the system, and put them in the review database. Then it does vectorizing with these reviews, which can extract frequent features in opinion words. In the last two step, all the features are analyzed in data modeling and a final predicted rating is produced.

Regarding review vectorizing, we used two ways to analyze token: (1)tf-idf (2) Spacy. The first method stands for term frequency-inverse document frequency, which is used to evaluate how important a word is to a document in a collection or corpus.Fig.4 gives the tf-idf formula. The second method, Spacy, which can leverage statistical information by training only on the nonzero elements in a word-word co-occurrence matrix, rather than on the entire sparse matrix or on individual context windows in a large corpus. [8] Fig.6 gives the Spacy architectural overview.

In the data modeling, we present two method to train data: (1)Logistic Regression (2)Linear Regression. The goal of the first method is classifying products into good product or bad product based on their reviews. Based on Fig.5, we used 4 star to be rating threshold, so if a product got high than 4 star,

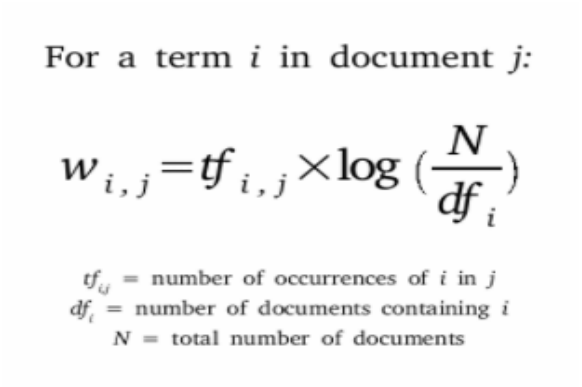


Fig. 4. tf-idf formula

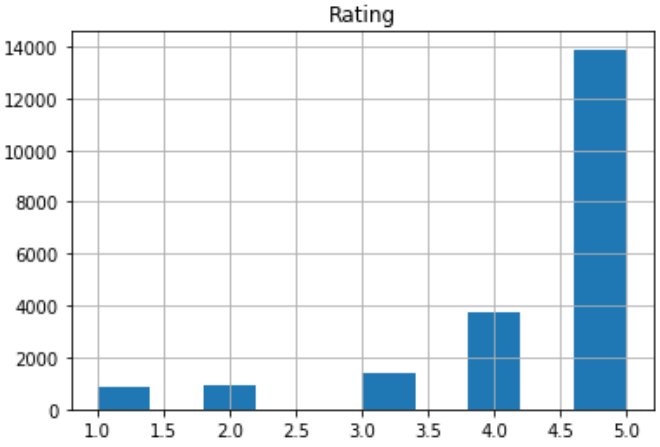


Fig. 5. Sephora Rating Distribution

it is considered as good product. However, the performance is extremely good that F1 reaches 0.9, which doesn't need to improve. Therefore, the remaining part we would only focus on the second method to explore more interesting thing. The goal of the second method is predicting rating of the product if the product isn't launched in Sephora.

B. Auto summery

There's five step to generate Auto summery. Most of implementation are based on rahulreddykr's work [9].

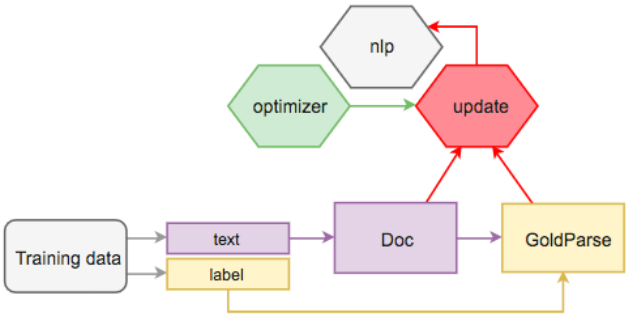


Fig. 6. Spacy architecture

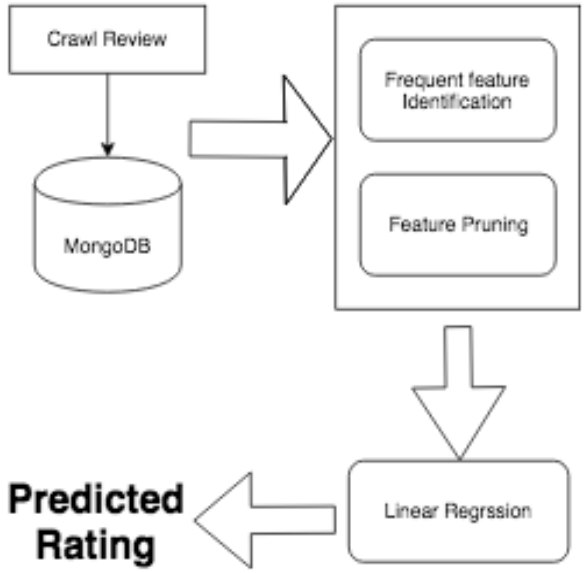


Fig. 7. Experiment flow

1) Tokenize the reviews into sentences.
2) Create a set of nouns in each sentence. Extract the most common nouns in all of the reviews using apriori algorithm. This gives us the most frequent features/aspects of the product.
3) Classify each sentence containing a product feature as positive, negitive or neutral using the Liu and Hu opinion lexicon.
4) Extract opinion phrases from sentences using nltk chunking. We identify the most frequent opinion phrases using frequency distribution.
5) Remove the positive onion phrase from negative opinion phrase set if there are the same. In the evaluation.

IV. Experiment

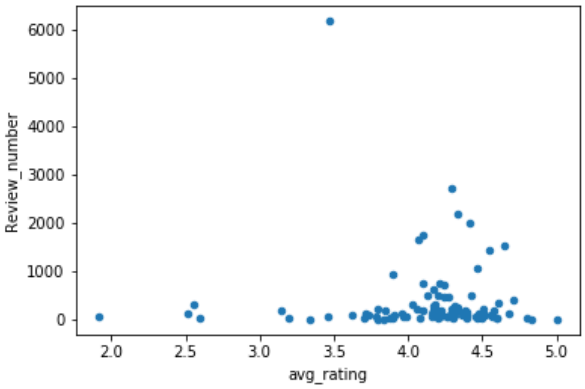
In this section we first describe the datasets, classifier and settings used for our experiments. Then, we present the best setting comparing our baseline, and using the best setting to perform predicted rating in Twitter. Finally, we present our auto summary result.

A. Dataset

In order to evaluate the result. We collect rating and review from Sephora and Amazon first. Both of them have structured data.

The biggest problem with Amazon is fake suppliers. Fig.9 shows a really common situation. The ”L’Oreal Paris” is the real supplier. ”L’Oreal” is a highly possible fake seller. We can guess by the number of products, however, it doesn’t work every time. Some of the real suppliers might not have enough products items. It’s no ground truth to set the threshold to eliminate fake supplier. It brings in the score under those fake supplier might affect the actual product ranking. Another problem comes from different supplier might use different title structure.Table I shows that no.8 and no.9 product are actually

Fig. 8. Amazon data set scatter plot



the same product. Except those cons, we still want to use amazon dataset. Fig.8 shows that the data set pretty balanced. For the most reviewed products, the average rating is around 3.5. We can get both positive onion and negative onion on it. Unlike well-picked products on Sepheora platform, which most product are satisfied for customer. The purpose for auto summery is to collect most of reasons will affect product quality. Amazon customer reviews seems to be better on this task.

TABLE I
Amazon Duplicate Product

	product
0	Maybelline New York The Blushed Nudes, 0.34 Ounce
1	Maybelline Makeup The 24K Nudes Eyeshadow Pale...
2	Maybelline New York Eyestudio ColorTattoo Meta...
3	Maybelline New York The Rock Nudes Palette, 0...
4	Maybelline Total Temptation Eyeshadow + Highli...
5	Maybelline New York Expert Wear Eyeshadow, Ear...
6	Maybelline Makeup The City Mini Eyeshadow Pale...
7	Maybelline The 24K Nudes Eyeshadow Palette, 0....
8	Maybelline Expert Wear Eyeshadow, Linen, 0.08 oz.
9	Maybelline Expert Wear Eyeshadow Quads, Mocha ...
10	Maybelline 24 Hour Eyeshadow, Bad To The Bronz...

Sephora is mature makeup online store which focuses more on beauty field. They select a high quality make-up product. It provides limited but reliable data. Both brand and product name are easy to analyze.

Twitter data is the final goal for this project. We assume the bias of comments is less than shop platform. It also needs more treatment of raw data. We collect data by simulating the ordinary user scroll the mouse since the free Twitter API doesn’t allow to track 7 days before. We extract the product name form previous two resources. Search all tweets contain exact phrase of a product name. Also, most of the people like to share their feeling on social media like twitter when they use the latest product rather than on E-commerce platform mall, therefore twitter data can help us to predict potential hottest makeup. Table II shows data source summary.

Regarding the experiment, we only focus on Sephora eyeshadow product and Twitter which mentioned these product, which are described in Table III We used Linear Regression to be our classifier. To compare various strategies, we used

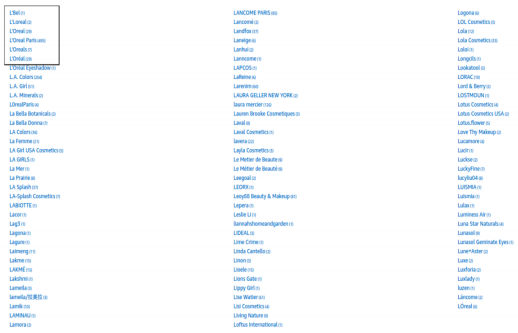


Fig. 9. Amazon contains many fake brand

TABLE II
DATA SOURCE SUMMERY

	Pons	Cons
Twiiter	Provide almost every product reviews	No product Catogooy
Amazon	Structured Score system	(1)Fake product (2)Combo set for selling. Can't rate for single product.
sephora	Structured Score system	Total product coverage is compare less

RMSE. The tokenizer we used is NLTK library, stemmer and tokenize come from SnowballStemmer and TweetTokenizer, respectively. For the train and test set, we got 13957 train set and 6875 test set in Sephora dataset. We present our result in the next section.

B. Evaluations on Predicting Rating

Regarding our baseline, we use tf-idf and uni-gram setting in tf-idf Vectorizer. We also another Vectorizer, Spacy to analyze our reviewText. However, based on result in Fig.IV, it is not really great. Therefore, we start to examine how Linear Regression would behave when it is tuned using the best parameter settings. To find out, we tuned hyper-parameter min_df i and ngram_range first. Specifically, min_df can ignore terms that appear in more than certain percentage and ngram_range can decide boundary of the range for extracting contiguous sequence. Also, rounding predict rating is another important step to change the result since Sephora review rating is integer number. Table.V and Fig.10shows results of various setting, and we finally got the best setting for Linear Regression that combing ngram_range is 1 to 3 and rounding predict rating, which represents that mining contiguous sequence of 1 to 3 words has positive influence on our result.

TABLE III
NUMBER OF COLLECTION

	# of Product(Eye Shadow)	# of Review
Twiiter	63	10726
Sephora	63	20842

TABLE IV
BASELINE

Vectorizer	RMSE
tf-idf	0.78
Spacy	0.62

TABLE V
TUNING HYPER-PARAMETER

index	hyper-parameter value	RMSE
[1]	min_df=5%	0.66
[2]	min_df=5% + ngram_range[1-3]+Rounding predict rating	0.60
[3]	Spacy+Rounding predict rating	0.57
[4]	ngram_range[1-3]	0.46
[5]	ngram_range[1-3]+Rounding predict rating	0.40

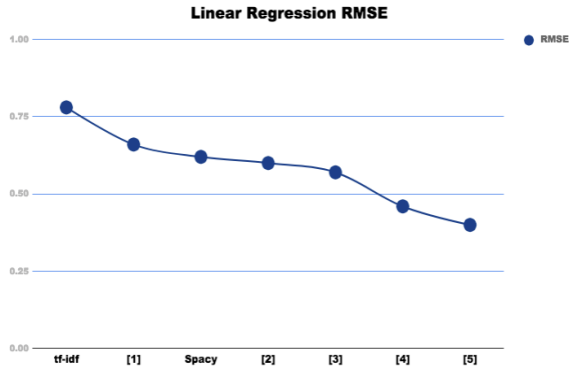


Fig. 10. RMSE comparison (use Table V index)

After getting the best setting for the model, we use this model to analyze Twitter reviews. Before predicting Twitter dataset, we drop duplicate review first, since there are too many advertisement that posting the same tweet. After that, we predict Twitter rating and got RMSE 0.42. It is really closed the RMSE value for Sephora testing set. In order to improve RMSE, we did the second data preprocessing, we drop reviews if a user post over 3 times reviews in each product. And the RMSE is decreasing a little from 0.42 to 0.39. Interestingly, Fig.11 shows that Sephora top 5 rating except no.1 doesn't match Twitter ranking. The reason we considered is the restriction of total number of Twitter review in each product. We scraped only 300 tweets in each product because we want to shorten waiting time when user use our model to get Twitter predicted rating.

p_id	Twitter_pred_rating	Rating	Twitter_Pred_Rank	Sephora_Rating_Rank
P427600	4.889017	4.769231	1	1
P411834	4.278297	4.733333	47	2
P309813	4.269807	4.689655	49	3
P410549	4.530885	4.671233	27	4
P397923	4.414549	4.656357	34	5

Fig. 11. Sephora vs Twitter ranking

C. Case Study on Auto summery

In the auto summery section. It's unsupervised learning process. We can't find the easy way to evaluate overall performance. We deliver part of key word to show the improvement we modified [9]'s code.

We add last step to remove non-relevant opinion phrase. Fig.IV-C and Fig.IV-C show that for most case in positive phrase is correct, however, in the negative case, it contains lot of positive opinion.

Fig. 12. Positive Opinion Phrase

```
Postive Opinon phrases:
[('great product', 983), ('great color', 502), ('great priced', 478), ('beautiful color', 372), ('great quality', 3
10), ('nicely color', 213), ('great pigmented', 199), ('long timely', 192), ('highly pigmented', 137), ('natural lo
ok', 134), ('neutral color', 127), ('great deal', 125), ('first timely', 118), ('great buying', 118), ('ever usin
g', 117), ('different color', 116), ('great palette', 109), ('little bit', 109), ('long way', 109), ('eyeshadow pri
mer', 108), ('pretty color', 107), ('excellent product', 105), ('brightnes color', 100), ('many color', 97), ('nice
ly product', 95), ('pretty good', 94), ('long last', 88), ('perfecting condition', 87), ('high ended', 87), ('favor
ite eye', 87), ('different shade', 83), ('really good', 83), ('great eye', 82), ('great primer', 82), ('great base
d', 81), ('great value', 81), ('fair skinned', 78), ('sensitive skinned', 76), ('blue eye', 75), ('natural color',
 73), ('reasonable priced', 72), ('skinned tone', 71), ('nicely pigmented', 70), ('really nicely', 70), ('high qual
ity', 70), ('perfecting color', 70), ('well pigmented', 67), ('great coverage', 66), ('great condition', 63), ('gre
at eyeshadow', 63)]
```

Fig. 13. Negative Opinion Phrase

```
Negitive Opinion phrases:
[('great product', 89), ('urban decay', 84), ('little bit', 79), ('first timely', 68), ('sensitive skinned', 66),
 ('dark brown', 66), ('great quality', 65), ('oily skinned', 65), ('great review', 61), ('great color', 55), ('high
ended', 49), ('long timely', 47), ('came', 42), ('dark circle', 42), ('highly pigmented', 39), ('never using', 39),
('different color', 38), ('pretty color', 35), ('great pigmented', 34), ('eyelid primer', 34), ('eyeshadow primer',
33), ('expensive brand', 33), ('brown haired', 32), ('dark skinned', 31), ('pale skinned', 30), ('bubble wrapped',
 29), ('beautiful color', 29), ('nicely color', 29), ("n't last", 29), ('great coverage', 28), ('great priced', 2
7), ('high quality', 27), ('many color', 27), ('dry skinned', 26), ('completely', 26), ('much product', 25), ('skin
ned tone', 24), ('broken eyeshadow', 24), ('long last', 24), ('different shade', 24), ('great primer', 24), ('littl
e', 23), ('high hoped', 22), ('great deal', 22), ('next day', 22), ('great thing', 22), ('hard timely', 22), ('real
ly wanted', 22), ('coastal scent', 21), ('eyeshadow based', 21)]
```

Once remove the duplicate phrase both in negative and positive collection.

Fig. 14. Negative Opinion Phrase after treatment

```
[('really disappointed', 16),
 ('real product', 13),
 ('intense color', 13),
 ('fake product', 11),
 ('cheap brand', 10),
 ('took forever', 9),
 ('highly disappointed', 9),
 ('negative review', 9),
 ('complete melted', 9),
 ('never knew', 8),
 ('extremely disappointed', 8),
 ('never received', 8),
 ('cheap stuff', 7),
 ('complete different', 7),
 ('expensive makeup', 7),
 ('bad quality', 7),
 ('complete messed', 6),
 ('extra money', 6),
 ('cheap packaging', 6),
 ('local drug', 6),
```

D. Case Study for Predict Rating

V. Conclusion and Future Work

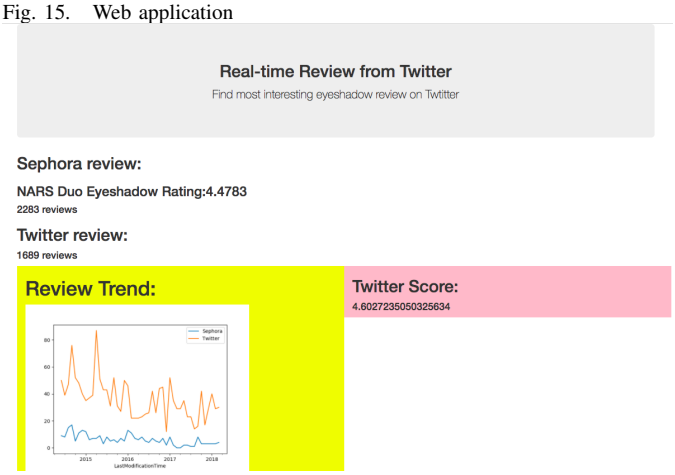
In the evaluation section, we propose the f-1 value for identifying positive and negative phrase. The result is not satisfied. [2] provide the algorithm to do aspect-level sentiment analysis. I our implement, we apply limited negation words list. And there's no weight among the positive words and negative words. Since whole sentence sentiment analysis is key for nest steps. Lot of positive sentence are identified as negative sentence. It bring in the potential issue, that's is the positive words are common in collection of negative sentence.

There is a possible solution to overcome the issue. We can collecting one star rating and five star rating only. To prevent those neutral review, which contain both positive opinion phrase and negative opinion phrase.

VI. Deployment

In this section, we present our flask web application and explain what's feature in it. User can search the product name and get estimated rating from Twiiter. After request send, the Scrapy framework will start to scrape the top 300 tweets. All the data transaction is on the MongoDB on the Amazon EC2 server. The framework provide the potential for multiple user request different item in the same time. We also provide rough trend for user. Based on the review number related to time.

Fig. 15. Web application



References

[1] Susan M Mudambi and David Schuff. Research note: What makes a helpful online review? a study of customer reviews on amazon. com. *MIS quarterly*, pages 185–200, 2010.

[2] Minqing Hu and Bing Liu. 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.

[3] Saptarshi Ghosh, Bimal Viswanath, Farshad Kooti, Naveen Kumar Sharma, Gautam Korlam, Fabricio Benevenuto, Niloy Ganguly, and Krishna Phani Gummadi. Understanding and combating link farming in the twitter social network. In *Proceedings of the 21st international conference on World Wide Web*, pages 61–70. ACM, 2012.

[4] Jianxing Yu, Zheng-Jun Zha, Meng Wang, and Tat-Seng Chua. Aspect ranking: identifying important product aspects from online consumer reviews. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1496–1505. Association for Computational Linguistics, 2011.

[5] Yuanbin Wu, Qi Zhang, Xuanjing Huang, and Lide Wu. Phrase dependency parsing for opinion mining. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3*, pages 1533–1541. Association for Computational Linguistics, 2009.

[6] Eugenio Tacchini, Gabriele Ballarin, Marco L Della Vedova, Stefano Moret, and Luca de Alfaro. Some like it hoax: Automated fake news detection in social networks. *arXiv preprint arXiv:1704.07506*, 2017.

[7] Rong Lu and Qing Yang. Trend analysis of news topics on twitter. *International Journal of Machine Learning and Computing*, 2(3):327, 2012.

[8] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[9] rahulreddykr. Review summarization aspect based opinion mining. https://github.com/rahulreddykr/Review_Summarization-Aspect_based_opinion_mining, 2013.