Automatically Determining Review Helpfulness

Hyung Yul Choi

June 9, 2015

Abstract

Customer reviews from commerce websites have valuable information for online shoppers. They help shoppers gauge whether or not a product is worth the purchase. However, reviews vary in their quality and helpfulness. Most commerce websites have voting systems where shoppers can vote on whether a review was helpful to them or not. For popular products however, the number of reviews can be in the thousands. As a result, not all reviews will get enough attention to receive helpfulness votes even though some may contain helpful information for other shoppers. In these scenarios, it would be desirable to be able to automatically collect the most helpful reviews. This research aims to do this by finding features in the review text that are indicative of its helpfulness and training a learning algorithm that can determine review helpfulness.

1 Introduction

As online commerce is becoming more popular, the value of customer reviews is rapidly growing. Online merchants often ask their customers to review the products that they have purchased and express their preferences or concerns. Customer reviews are an increasingly important source of content as they offer information to both the customer and producers. Online shoppers usually read through reviews written by other customers to help them make their purchasing decisions. For a popular product however, the number of reviews can be in the hundreds or even thousands. It becomes increasingly difficult for shoppers to make an informed decision on whether the product is worth purchasing since it would be impossible to read through all the reviews.

Additionally, the lack of editorial and quality control causes the helpfulness of reviews to vary which also impedes the ability to objectively evaluate a product. For example, some customer reviews contain very few words, while others are lengthy but contain very few sentences that actually pertain to informative aspects about the product. Particularly for popular websites, the quality of a review of a product can range from excellent, detailed information to unhelpful, biased opinions. J. Otterbacher notes that the quality of information available in an online community is often inversely related to the number of users [1].

One of the most widely used commerce websites, Amazon, attempts to alleviate this problem by allowing their shoppers to vote on whether they think a review was helpful or not. The collection of votes is then represented as a ratio that is provided in the form of "X out of Y people found the following review helpful." The reviews can be sorted according to their number of helpfulness votes. Although this feature is certainly an improvement, there are still important issues to be addressed. Liu and Huang have noted that a newly posted review will have very few votes, and though it may be an excellent review, it will most likely be ignored by readers. This is because Amazon has a review "front page," where the reviews with the most number of votes are listed first. Even if new reviews are noticed by some readers, it will be difficult to determine their helpfulness due to the lack of votes [6]. Furthermore, reviews that are on the "front page" receive more attention by customers, and thus they get even more helpfulness votes,

further widening the gap. Liu et al. also noted this in their research of online product reviews and called it the “early bird bias.” They also discovered in their analysis a “winner circle bias,” where the more votes a review gets, the more authority and attention it would be given by shoppers [3]. As a result, there are reviews that get little attention and authority even though they may have important and helpful information for customers.

In these scenarios, a way to automatically determine the helpfulness of reviews is desired. This paper approaches the problem of evaluating the helpfulness of reviews in order to identify the most helpful reviews for the customers, regardless of how many helpfulness votes they have. This would filter the good and the bad reviews and allow a quick system that can automatically collect the most helpful reviews.

For this research, we pinpoint helpfulness by looking at preexisting reviews of products on Amazon. We use frequently voted reviews as the basis of our definition of helpfulness, and we use this definition in order to determine the features that are indicative of helpfulness. Finally, we use these features to train a machine learning algorithm that can determine the degree of helpfulness of a given review.

2 Background and Related Work

Previous work on determining the quality of a review have mostly been determining what various features make good product reviews. Liu and Huang developed models to predict the helpfulness of movie reviews collected from IMDB. The features that they based their model on were reviewer expertise, writing style, and the date the review was written on. They found that frequent reviewers of a particular genre are better predisposed to writing a good review. They also found that reviews that are written well and readable reflected a quality review [6]. Kim et al. tested structural factors such as the length of a review, lexical features such as unigrams and bigrams, and meta-data on reviews of MP3 players and digital cameras from Amazon. They found that the length of the review has the greatest impact in determining the quality of a review [4].

Liu et al. decided against using the helpfulness voting system as a determinant of a review's quality and opted to use human subjects to classify review helpfulness. They identified several features that determined a review's quality. A high quality review was informative about specific aspects of the product and usually included both pros and cons. In addition, they also found that readability was also a big influence [3]. Liu et al. also decided to use human subjects to classify reviews. Instead of determining overall features of a review, they compared which features were valued depending on the type of person. Specifically, they focused on how the helpfulness of product reviews are perceived by design engineers [5].

3 Approach and Methods

3.1 Dataset

The dataset consists of reviews of electronic products from Amazon used in McAuley and Leskovec's research [2]. There are 1,241,778 individual reviews consisting of 82,067 different products. Each review data consists of the product id, the product price, the reviewer id, the helpfulness ratio, the rating of the product, the time of the review, and the review text.

Total # of Reviews	1241778
# of Reviews, ≥ 10 Votes	167604
# of Reviews, ≥ 10 Votes and ≥ 5 sentences	116680
Average Helpfulness Ratio	0.78
Average Length of Review (words)	141

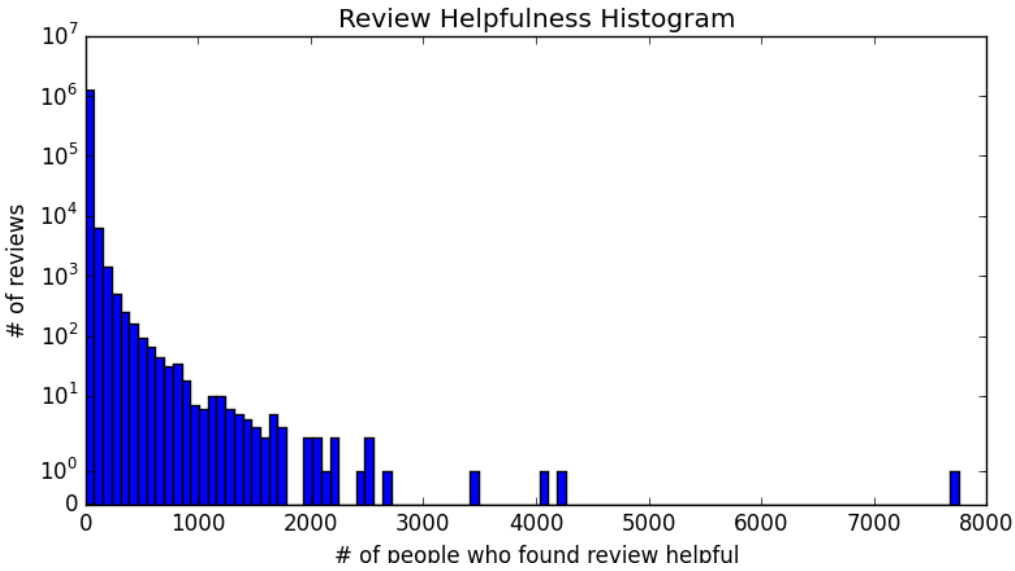

Figure 1: Histogram of all the reviews and the number of total votes. Note logarithmic scale for the number of reviews

From the initial analysis of the data, the range in the quality of reviews is obvious. The collection of reviews has an average length of 108 words, but the length ranges from 5,361 words to just two words. The average number of helpfulness votes is 8.9 and ranges from 0 to 7750 votes. The histogram of total votes shows that most reviews have little to no votes and are hidden in obscurity. More than 50% of the reviews have one vote or less. Preprocessing is required to extract out the reviews that are most useful for further analysis.

We can't test the features on all the reviews since there are reviews in the dataset that do not have enough information. A review that has almost no votes or has only a few words aren't very useful since the information it provides is so minimal for some of the features that we test. For these reasons, the reviews that we extract out to use for testing have at least 10 helpfulness votes and have at least 5 sentences. Out of dataset, there are 167,604 reviews that have 10 votes or more. Once we further extract the reviews that have at least 5 sentences, we are left with 116680 reviews to work with. Due to our large

dataset, there is still a substantial amount of reviews that can be used to test our features with. The average helpfulness ratio from the final testing dataset is 0.78 and the average length is 141 words.

3.2 Identifying Features

Once the reviews with enough information are extracted, they can now be tested for potential correlations. The two pieces of information we're mainly looking at are the review text and the helpfulness ratio. The helpfulness ratio is the number of votes where people found the review to be helpful over the total number of votes. This ratio is a number between 0 and 1 where one means all the voters found the review to be helpful and 0 means none of the voters found the review to be helpful. We compare the helpfulness ratio with a particular feature within the text to see if there are any potential correlations. We test for correlations using the Pearson's correlation coefficient which gives a number between 1 and -1, where 1 is total positive correlation, 0 is no correlation, and -1 is total negative correlation.

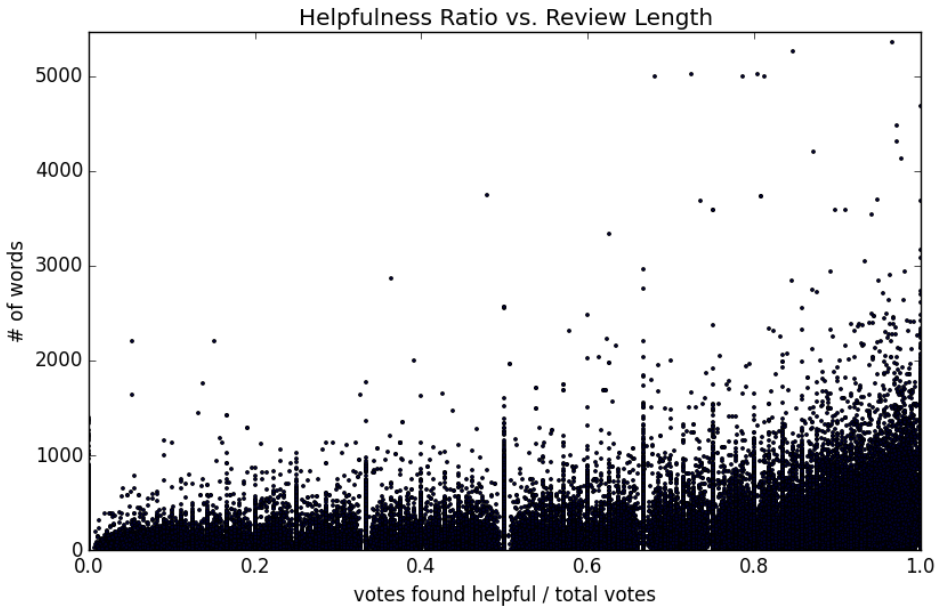

Figure 2: Comparison of the length of the review in terms of the total number words and the helpfulness ratio of the review. Correlation r = 0.258

One of the first features we tested for was comparing the length in terms of words with the helpfulness ratio. Figure 2 shows a slight upward trend in the number of words as helpfulness increases. According to Pearson's correlation coefficient score, there is a positive correlation of 0.258. Likewise, the average number of sentences per review also has a positive correlation as illustrated in Figure 3. This means that helpful reviews generally tend to be longer in length. The cause of this is likely because reviews that are longer usually have more information, and thus the review will likely be more helpful.

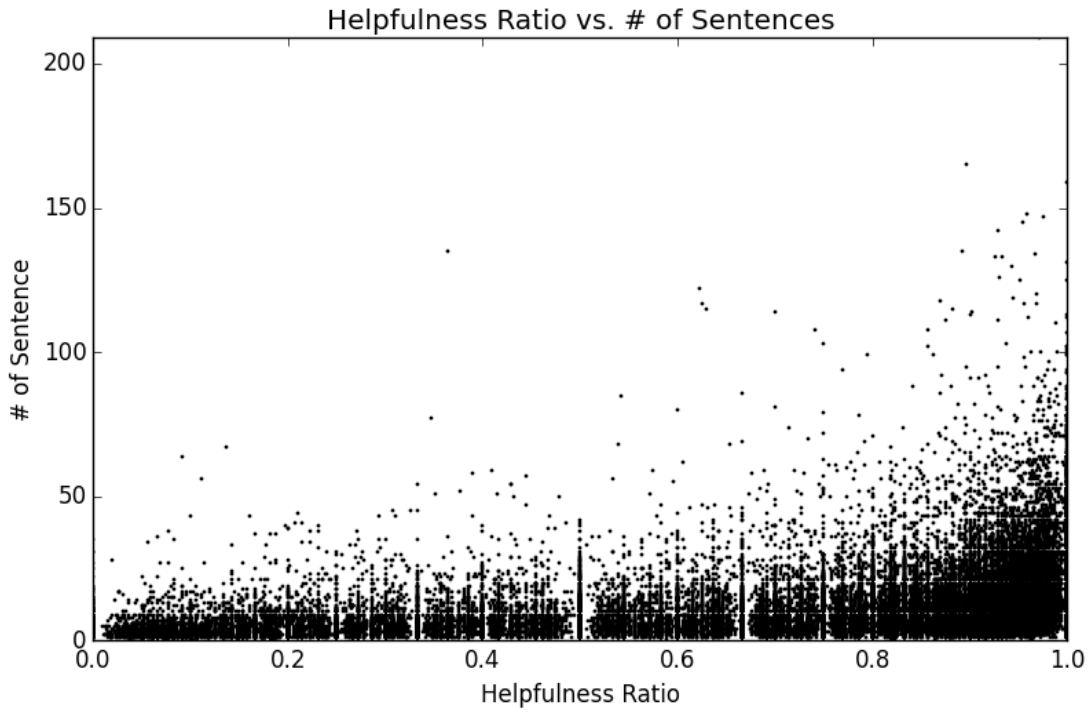

Figure 3: Comparison of the length of the review in terms of the total number sentences and the helpfulness ratio of the review. Correlation r = 0.267

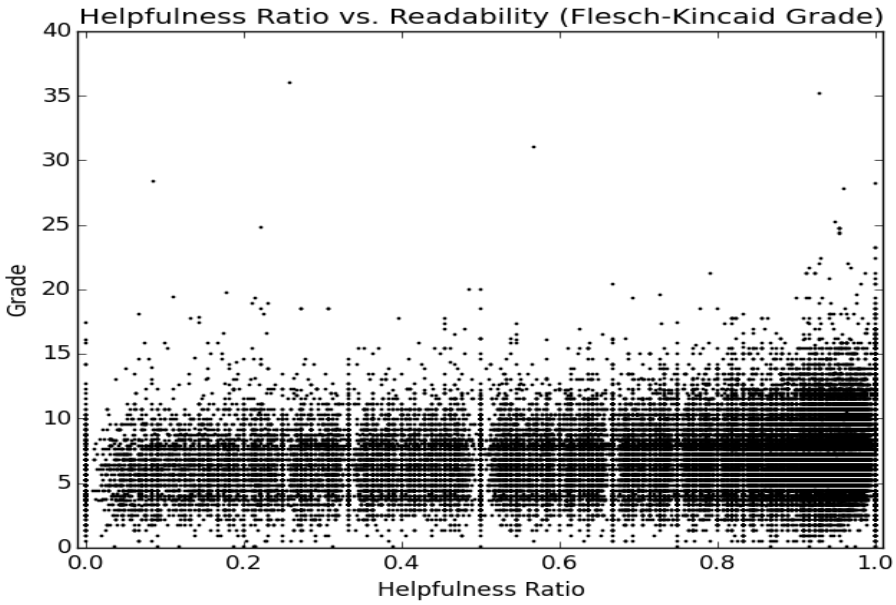

Figure 4: Comparison of the readability of the text and the helpfulness ratio of the review. Correlation r = 0.17

As part of the Python package *textstat*, the Flesch-Kincaid Grade Level Test indicates the comprehension difficulty of the review's text. Because the test requires adequate data to produce accurate results, reviews with five or more sentence had to be used. A higher score in the test means the review requires are higher level of comprehension. The test does not measure difficulty in comprehension due to poor writing; it measures the difficulty in comprehension in terms of the reviewer's writing ability. The result from the test shows that good reviews tend to be better written. Its Pearson correlation score is 0.17, indicating that there is a slight positive correlation.

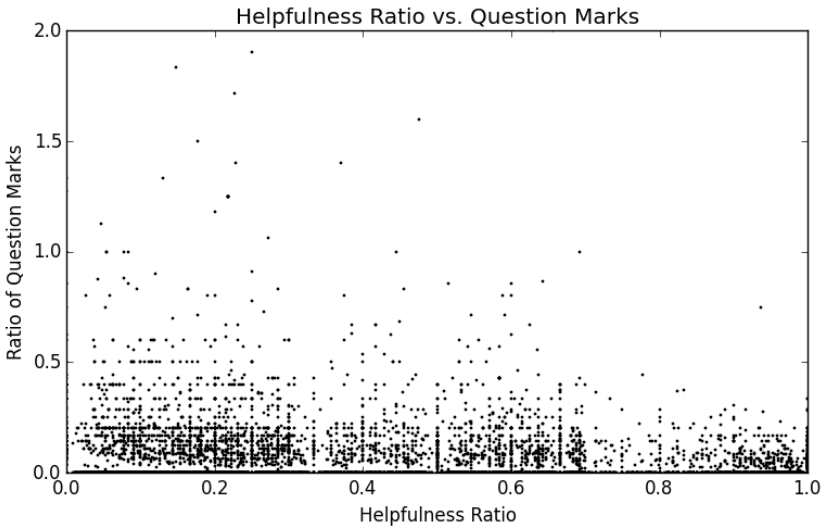

Figure 5: Comparison of the usage of question marks relative to the total length of the review and the helpfulness ratio of the review. Correlation r = -0.32

Another interesting correlation we found is the use of punctuations relative to the total length of the review. Certain extraneous punctuations add unnecessary emotion and bias to an otherwise informative review. Our tests show that the overuse of exclamation marks is an indicator of unhelpfulness as shown by its negative correlation of -0.21. Likewise, the use of question marks was also found to be an indicator of unhelpfulness. Intuitively, this makes sense because using extraneous punctuations like exclamation or question marks is generally inappropriate. Exclamations add excessive emotional sentiment while questions aren't needed since reviews shouldn't be asking questions. Helpful reviews are supposed to be informative and objective.

A final feature that we found to have a slight correlation with helpfulness is sentiment polarity. Using the tools provided by NLTK [9], we were able to produce a polarity score of how emotionally charged the language is for a given body of text. Whether it's strongly positive or negative, a higher score indicates a greater strength of the sentiment. Our tests showed a negative correlation of -0.15, suggesting that the use

of emotionally strong language is related to unhelpfulness. Helpful reviews were found to be more neutral and objective in its use of language.

Unfortunately, not all the features that we explored had correlations. For example, we tested for the average sentence length of each review. Going along with the readability and comprehension of the text, we tested for this feature because we believed that eloquent, informative sentences would be preferred to short sentences. As illustrated in Figure 6, our tests showed very little correlation with helpfulness.

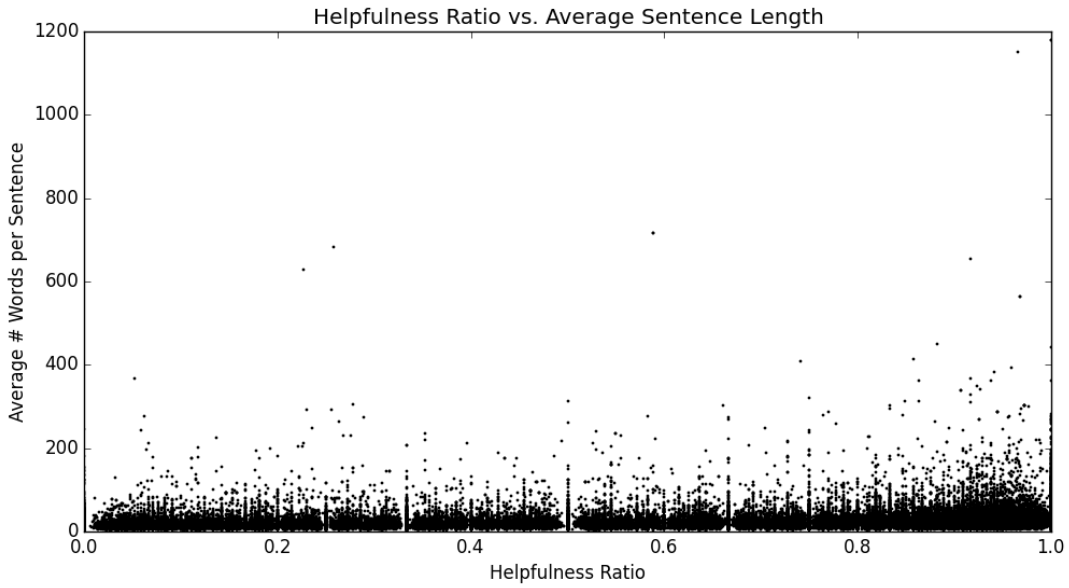

Figure 6: Comparison of the average sentence length and the helpfulness ratio of the review. Correlation r = 0.07

Finally, we tested for the use of different grammatical words in the text. The process of part-of-speech tagging was used where each word was marked up corresponding to its particular word identification. For each review, adjectives, nouns, and verbs were separately counted and normalized to the total number of words. All of our tests showed that there is almost no connection between the use of these words and helpfulness. For each test, their correlation coefficients were very close to 0. To further

analyze the feature, we separated out the differences in ratio between poor, neutral, and good reviews as shown in the tables below.

Adjectives

	Poor Reviews	Neutral Reviews	Good Reviews
Ratio	0.27	0.22	0.20

Nouns

	Poor Reviews	Neutral Reviews	Good Reviews
Ratio	0.48	0.40	0.41

Verbs

	Poor Reviews	Neutral Reviews	Good Reviews
Ratio	0.17	0.15	0.16

4 Results and Discussion

Using the features with the strongest correlations, we trained a decision tree classification algorithm and used it as our predictive model. To begin with, we trained our model to classify reviews into one of three categories: good, neutral, or poor. We used cross-validation where we partition the data into a training set

and a testing set, where we train the prediction model with the training set and validate its accuracy with the testing set. We did 10 rounds of this and averaged the accuracy results to reduce variability within any one of the set selection. We initially set our baseline accuracy to be the random selection of one of the three classifications. The average accuracy achieved from our prediction model was 43%. When compared to the random baseline accuracy of 33%, our prediction model performed slightly better.

		Predicted		
		Poor	Neutral	Good
Actual	Poor	308	146	133
	Neutral	217	201	169
	Good	192	187	208

Figure 7: Confusion matrix of one of the rounds of the test. Trained features: length (words), length (sentences), readability, sentiment polarity, punctuations (exclamation/question marks)

To try to improve upon our results, we changed some of the variables and reran our tests. For example, we tested again using different subsets of features to see if we could produce different results. One of the subsets where we saw an improvement in accuracy was when we tested without the readability feature. Because the readability feature required at least 5 sentences to produce a result, removing that feature allowed for more reviews to be included in the training set. Since poorer reviews tended to be shorter in length, more poor reviews were added to the dataset, improving the correlations for features that compared the length of the review to helpfulness. The average accuracy without the readability feature increased to 48%, and the most dramatic increase in accuracy occurred when classifying poor reviews.

Finally, we reduced the number of classifications from three to two classes in order to further improve our accuracy. Our results showed that average accuracy did increase for both classes, but the most significant increase occurred when classifying poor reviews where accuracy went up to 69%. For positive

reviews, the accuracy was only 59%, which is only slightly better than the random baseline accuracy of 50%.

		Predicted	
		Poor	Good
Actu	Poor	402	179
	Good	239	352

Figure 8: Confusion matrix with two classes. Trained features: length (words), length (sentences), sentiment polarity, punctuations (exclamation/question marks)

However, previous work done by [3] achieved a consistent accuracy result above 80% using an SVM and classifying reviews into three classes. There are still improvements to be made if we are to compare our prediction model to that of the previous work's.

5 Future Work and Conclusion

The future work would be to explore more advanced features such as finding lexical information from the text. This means figuring out whether a review is informative about the actual product or specific parts of the product, rather than informing about something unrelated. Other features include implementing aspect-based opinion mining [8], where it attempts to find associations of words with specific parts of the product. In addition, it may be possible to look at information beyond the review text. For example, frequent reviewers or highly rated reviewers tend to write more helpful reviews than someone who is writing one for the first time. By identifying the good reviewers, we can use them as guidelines to helpfulness. These features, once implemented, can be added to our current prediction model to potentially help its prediction accuracy.

In conclusion, this research was first motivated by the desire to collect the most helpful reviews for shoppers, especially for products that have too many reviews. Then, with the available data, we looked at

the possible features of reviews that are suggestive of its helpfulness by comparing the feature with the helpfulness ratio. Finally, with the most useful features, we taught a learning algorithm that could predict a review's classification in terms of its helpfulness. Our predictive model was able to achieve an accuracy that was slightly better than the random baseline accuracy but fell far short of the accuracies achieved by previous works.

References

[1] J. Otterbacher. Helpfulness in online communities: a measure of message quality. Proceedings of the 27th international conference on human factors in computing systems, ACM, Boston (MA, USA) (2009), pp. 955–964

[2] J. McAuley and J. Leskovec. Hidden factors and hidden topics: understanding rating dimensions with review text. RecSys, 2013.

[3] Jingjing Liu, Yunbo Cao, Chin Y. Lin, Yalou Huang, Ming Zhou. Low-Quality Product Review Detection in Opinion Summarization In Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL) (2007), pp. 334-342

[4] 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 (EMNLP '06). Association for Computational Linguistics, Stroudsburg, PA, USA, 423-430.

[5] Jian Jin and Ying Liu. 2010. How to interpret the helpfulness of online product reviews: bridging the needs between customers and designers. In Proceedings of the 2nd international workshop on Search and mining user-generated contents (SMUC '10). ACM, New York, NY, USA, 87-94. DOI=10.1145/1871985.1872000

[6] Yang Liu; Xiangji Huang; Aijun An; Xiaohui Yu, "Modeling and Predicting the Helpfulness of Online Reviews," Data Mining, 2008. ICDM '08. Eighth IEEE International Conference on , vol., no., pp.443,452, 15-19 Dec. 2008. DOI: 10.1109/ICDM.2008.94

[7] Ying Liu, Jian Jin, Ping Ji, Jenny A. Harding, Richard Y.K. Fung, Identifying helpful online reviews: A product designer's perspective, Computer-Aided Design, Volume 45, Issue 2, February 2013, Pages 180-194, ISSN 0010-4485, http://dx.doi.org/10.1016/j.cad.2012.07.008.

[8] Abbasi Moghaddam, Samaneh. Aspect-based opinion mining in online reviews. Diss. Applied Sciences: School of Computing Science, 2013.

[9] Bird, Steven, Edward Loper and Ewan Klein (2009), Natural Language Processing with Python. O'Reilly Media Inc.