# Seeing Stars from Reviews by a Semantic-based Approach with MapReduce Implementation

Pengfei Liu, Xiaojun Qian and Helen Meng
Department of Systems Engineering and Engineering Management
The Chinese University of Hong Kong, Hong Kong SAR, China
{pfliu,xjqian,hmmeng}@se.cuhk.edu.hk

## Abstract

This study concerns the problem of aspect-level opinion (sentiment) mining from online reviews. The problem consists of two fundamental sub-tasks: aspect extraction (identify specific aspects of the product from reviews), and aspect rating estimation (offer a numerical rating for each aspect). Solving this problem is important and useful for many applications, e.g., providing aspect-level review summaries to consumers for better decision making, and for product manufacturers to collect summarized user feedback. Our objective is to propose a semantic-based approach for aspect level opinion mining from massive amounts of reviews in a scalable fashion. The MapReduce implementation for this approach obtains much runtime reduction compared with the single-process implementation. Experimental results show that the runtime reductions by the MapReduce implementation are almost linear to the number of mappers, e.g., around **7.4** times reduction with 10 mappers on the TripAdvisor dataset and **2.6** times reduction with 4 mappers on the Yelp dataset. The number of mappers and reducers can be configured on demand to handle very large datasets in a scalable fashion. Moreover, the semantic-based approach obtains good performance for aspect rating estimation on the TripAdvisor dataset, with the MAE score of around 1.0 on all aspects, which means that the average deviation between the human rating and the estimated rating is around 1 star. The source code of our implementation for the sentiment-based approach can be downloaded from https://github.com/ppfliu/aspect-opinion.

## 1 Introduction

E-commerce websites are rapidly growing with numerous customer reviews contributing to Big Data. A single product may have thousands of reviews, which are too voluminous for human reading and analysis. Opinion mining from reviews is a very active research area in recent years because more and more reviews are available and a lot of valuable "treasures" are buried in these rich, informative reviews. The problem of opinion mining has at least three different levels: the document (review) level, sentence level and aspect level. Document level and sentence level opinion mining are usually too coarse, as they cannot explicitly point out people's opinions on specific aspects of a product, such as the weight, battery life, screen size, and camera quality of a smartphone.

*Aspect-level opinion mining* extracts customers' opinions on specific aspects of a product from reviews, and is thus more fine-grained. It consists of two major subtasks: aspect extraction (identifying specific aspects of the product from reviews), and aspect rating estimation (offering a numerical rating for each aspect). It extracts customers' opinions on specific aspects of a product from reviews. These extracted aspect-level opinions are very useful in several applications, e.g., in recommender systems that show aspect-level opinions from customers about a product and thus help potential customers make better purchase decisions. They can also help product manufacturers collect customers' opinions on specific aspects of their products.

In this paper, we study the problem of aspect-level opinion mining from reviews. Solving this problem is important and useful for many applications, e.g., providing aspect-level review summaries to consumers for better decision making, and for product manufacturers to collect summarized user feedback. Our objective is to propose a semantic-based approach for aspect level opinion mining from massive amounts of reviews in a scalable fashion. We try to extract specific aspects of a product from reviews and rate each aspect with a numerical value (1-5 stars) automatically by sentiment analysis. The rest of this paper is organized as follows: Section 2 discusses related work; Section 3 presents the semantic-based approach with the MapReduce implementation; and Section 4 shows the experimental datasets, evaluation metric and experimental results. We conclude the paper and propose future work in Section 5.

## 2 Related Work

Lexicon-based methods have been proposed for sentiment analysis in several papers. Hu and Liu [4] used association mining to find frequent itemsets (i.e., sets of co-occurring words) as candidate frequent aspects (or features), and then pruned meaningless and redundant aspects. To prevent ignoring infrequent but interesting features, they extract these features by finding the nearest noun/noun phrases around the opinion words. They then utilized the adjective synonym set and antonym set in WordNet [3] to estimate the semantic orientations (positive or negative) of adjectives. Their method maintains a list of seed adjectives with known sentiment orientations, and searches both WordNet and the seed list for each target adjective word to predict its orientation and updates the seed list with new adjective and its orientation. Moghaddam and Ester [10] first extracted the nearest adjectives to each aspect and then adopted a $K$ Nearest Neighbor algorithm to estimate the sentiment of each extracted adjective, using WordNet to compute the similarity between adjectives.

Classification-based methods need labeled training data and

 ISBN: 978-1-62561-003-4 

are usually domain-dependent. Pang et al. [12] examined the effectiveness of applying machine learning techniques to determine whether a review is positive or negative for the movie domain and obtained good classification accuracy. Jin et al. [5] proposed a Hidden Markov Model based framework to extract product entities and their associated opinion orientations, by integrating linguistic features such as part-of-speech tags, lexical patterns and surrounding words/phrases. Shariaty and Moghaddam [13] used the Conditional Random Fields model to identify aspects and opinions in the sentence, by learning from the datasets with labels on aspects, opinions and background words.

Topic models are also widely used in aspect-based opinion mining. Titov and McDonald [14] presented a multi-grain topic model by extending the Latent Dirichlet Allocation (LDA) model to extract ratable aspects and cluster them into coherent topics. Lin and He [6] proposed an unsupervised LDA-based joint sentiment/topic model to detect aspect and sentiment simultaneously from reviews. Lu et al. [8] applied topic models to estimate aspect ratings from all reviews (super-review) on the item. Moghaddam and Ester [11] summarised the LDA-based models for aspect-based opinion mining.

The semantic-based approach described in this paper is similar to [4] and [10] in terms of adjective-noun word pair extraction and sentiment estimation using a lexical database. However, we focus on an scalable algorithm for very large datasets. Moreover, we intend to estimate personalized aspect ratings for each individual user from his/her review on the item (hotel or restaurant). This is different with [8], which excludes super-reviews with fewer than 10 reviews, as topic models usually need more reviews to estimate a reliable topic distribution. However, our semantic-based approach can work well even on a single sentence.

# 3 The Semantic-based Approach

## 3.1 Overview

Figure 1 is the block-diagram of the semantic-based approach.

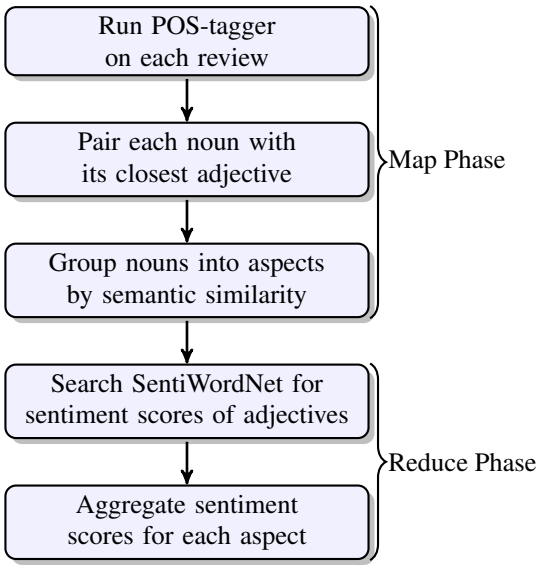


Figure 1: Block diagram of the Semantic-based Approach for Aspect-level Opinion Mining.

> My wife took me here on my birthday for breakfast and it was excellent. The weather was perfect which made sitting outside overlooking their grounds an absolute pleasure. Our waitress was excellent and our food arrived quickly on the semi-busy Saturday morning. It looked like the place fills up pretty quickly so the earlier you get here the better. Do yourself a favor and get their Bloody Mary. It was phenomenal and simply the best I've ever had. I'm pretty sure they only use ingredients from their garden and blend them fresh when you order it. It was amazing. While EVERYTHING on the menu looks excellent, I had the white truffle scrambled eggs vegetable skillet and it was tasty and delicious. It came with 2 pieces of their griddled bread with was amazing and it absolutely made the meal complete. It was the best "toast" I've ever had. Anyway, I can't wait to go back!

Figure 2: A Restaurant Review Taken from the Yelp Dataset with Adjectives in Red and Nouns in Green.

We illustrate how our approach works using the example review in Figure 2. Suppose we want to extract the customer's opinions on the four pre-defined aspects (*food, service, decor and place*) of a restaurant. We describe the three major steps as follows:

**(I) Extract adjective-noun word pairs**

For each review, we first run a part-of-speech (POS) tagger[1] to obtain the POS tags of its words, using the Penn Treebank tagset [9], e.g., *NN, NNS, NNP, NNPS* for nouns, *JJ, JJR, JJS* for adjectives. As 60-70% of aspects are explicit nouns [7], our current implementation pairs a noun with its nearest (left or right) adjective. Then, for each noun (e.g., *breakfast*) in each sentence of the review, we search for the corresponding adjective within a certain *search range*. For example, the *search range* of 4 means that we search left and right within 4 words away from current noun. Once an adjective (e.g., *excellent*) is found, the search stops and returns the noun with the adjective as an adjective-noun word pair. Otherwise, the nouns (e.g., *wife, birthday*) will not be included in the word pairs.

From the example review, this step extracts 16 adjective-noun word pairs: *[excellent breakfast], [perfect weather], [absolute grounds], [absolute pleasure], [excellent waitress], [excellent food], [semi-busy morning], [fresh garden], [excellent menu], [white truffle], [white egg], [white vegetable], [tasty skillet], [amazing bread], [complete meal], [best toast].*

The search range should not be too small or too large. For example, if the distance is only 2 words away, the pair *[excellent breakfast]* will not be extracted; when the distance is 4 words away, the pair *[fresh garden]* is incorrectly extracted. If two nouns in one sentence are both close (within the search range) to the same adjective, the extracted word pairs may be incorrect. For example, in the sentence of "Our waitress was excellent and our food ...", the pair *[excellent food]* is incorrectly extracted, as "excellent" actually modifies the word "waitress".

**(II) Group word pairs into aspects**

This step first filters out irrelevant word pairs if they are not

[1] http://nlp.stanford.edu/software/corenlp.shtml

similar to any of the pre-defined aspects and then groups relevant word pairs into their corresponding aspect. A word pair is grouped into an aspect if the semantic similarity between the noun of the pair and the aspect word is above the *similarity threshold*. We adopt the Java library WS4J (WordNet Similarity for Java)[2] to compute the semantic similarity between words by the Wu and Palmer [15] metric, which has the similarity range from 0.0 (dissimilar) to 1.0 (identical).

In the example, the irrelevant word pairs (*[perfect weather], [absolute grounds], [absolute pleasure], [semi-busy morning]*) are filtered out first as none of them are similar to any of the four aspects. Then, the pair *[excellent waitress]* is grouped into the *service* aspect. All other pairs are grouped into the *food* aspect as their nouns are semantically similar to food.

**(III) Estimate an average rating by aggregation**
This step first searches the SentiWordNet [1] database to get a sentiment score for the adjective in each word pair, and then aggregates the scores of all the word pairs of an aspect to get an average aspect score (converting to 1-5 star rating) per review given by a user on an item.

For the example review, only two aspects are reviewed: *food* and *service*. The adjectives *"excellent, tasty, amazing and best"* are strongly positive and therefore the step estimates 5-star rating for both aspects. The other aspects, e.g., *decor* and *place* are not available in the example review.

## 3.2 MapReduce Implementation

To handle large amounts of reviews in parallel, we have implemented the semantic-based approach on the Hadoop framework. Hadoop is an open-source Java implementation of MapReduce, a parallel programming model proposed by Google [2], to handle large datasets with two phases: the *map phase* and the *reduce phase*. In the map phase, *mappers* process a large dataset in the form of key/value pairs and generate a set of intermediate key/value pairs. Then in the reduce phase, *reducers* merge all intermediate values associated with the same key and output the result.

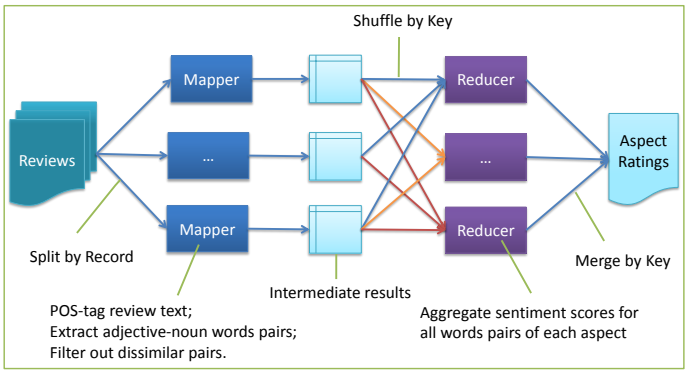


Figure 3: MapReduce Implementation for the Semantic-based Approach.

As illustrated in Figure 3, reviews are first split by *record* into several sets of records. A *record* consists of the mandatory user id and item id to ensure that it will be *atomically* processed by its assigned mapper, as well as review, review date and overall rating. A mapper receives the records as input, generates POS-tags for each review, filters out dissimilar pairs if their similarity with any of the pre-defined aspects is below the *similarity threshold*, groups relevant adjective-noun pairs into their corresponding aspect, and emits the key-value pairs with the format of (**Key**: userid, itemid; **Value**: adjective-noun pair, aspect). A reducer gets the intermediate key/value pairs from the mapper, searches the SentiWordNet to get a sentiment score for the adjective of each adjective-noun pair, aggregates all corresponding sentiment scores for each aspect, and finally outputs the key-value pairs with the format of (**Key**: userid, itemid; **Value**: ratings of each aspect). The output key-value pair for the example review is like this: *[userid, itemid; 5, 5, 0, 0]*, meaning that the aspects of *food* and *service* have 5-star rating, while the other two aspects (*decor, place*) have no ratings.

# 4 Experiments

## 4.1 Datasets and Evaluation Metric

We conduct experiments on the TripAdvisor dataset[3] for the *hotel* domain, and the Yelp's Academic dataset[4] for the *restaurant* domain. The TripAdvisor website has designed several aspects (e.g. room, location, cleanliness, service, etc.) of a hotel for customers to rate, and therefore the TripAdvisor dataset has human aspect ratings. However, the Yelp dataset has no human aspect ratings. Table 1 lists the number of users, items and reviews, excluding the data records with no reviews. We use a subset of the Yelp dataset that contains only the restaurant category.

Table 1: Statistics of Experimental Datasets.

| | No. of Users | No. of Items | No. of Reviews |
|---|---|---|---|
| TripAdvisor | 148,450 | 1,850 | 246,365 |
| Yelp | 36,472 | 4,503 | 158,424 |

We manually defined 5 aspects: *location, room, service, value and cleanliness* for TripAdvisor and 4 aspects: *food, service, decor and place* for Yelp. The metric of *Mean Absolute Error* (MAE) is used to evaluate the accuracy of aspect ratings estimation, as defined in Equation 1. $N$ denotes the number of ratings, $r_i$ and $\hat{r}_i$ are the real and estimated ratings, respectively.

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|r_i - \hat{r}_i| \quad (1)$$

## 4.2 Experimental Results

The semantic-based approach allows us to check and interpret the intermediate results (e.g., adjective-noun word pairs and their ratings), and then tune the parameters of *search range* and *similarity threshold*). Table 2 lists some example phrases and their ratings extracted from TripAdvisor and Yelp. The adjective-noun word pairs for ratings 4 and 5 contain strong positive adjectives while the pairs for ratings 1 and 2 contain strong negative adjectives.

[2]https://code.google.com/p/ws4j/

[3]http://sifaka.cs.uiuc.edu/ wang296/Data/LARA/TripAdvisor/

[4]https://www.yelp.com/academic_dataset/

Table 2: Phrases from TripAdvisor/Yelp Extracted by the Semantic-based Approach.

| Ratings | Extracted Phrases from TripAdvisor | Extracted Phrases from Yelp |
|---|---|---|
| Rating 1 | disappointing hotel, outdated lobby, smelly place, crappy shower, nostalgic atmosphere, disappointing service | disappointing broth, infuriating part, disappointing meal, smelly store, scary dinner, crappy service |
| Rating 2 | poor service, stupid stance, other hotel, worse accommodation, negative aspect, quirky breakfast, sleepless night | bad service, diabetic coma, bad food, bad experience, poor substitute, rancid smell, pathetic staff, bad pasta |
| Rating 3 | free breakfast, fresh fruit, quiet accommodation, small room, above average, modern decor, inexpensive price | raw radish, hot pepper, small salad, hot sauce, hot chocolate, casual environment, yellow curry, low price, small fries |
| Rating 4 | better service, nice option, gracious hospitality, beautiful place, better fruit, best hospitality, beautiful room, lovely hotel | best toast, healthy eating, delicious chicken, beautiful decoration, best taco, praiseworthy service, delicious pork |
| Rating 5 | relaxing stay, wonderful vacation, perfect place, complimentary breakfast, quaint room, happy atmosphere, excellent stay | yummy bonus, perfect texture, pleasant environment, amazing ambiance, excellent service, perfect cheesecake |

We implement the semantic-based approach with two programs: a single-process Java program and a distributed MapReduce program. Table 3 compares the runtime performance (measured in minutes) between them. We configured 10 mappers and 6 reducers for TripaAdvisor, 4 mappers and 6 reducers for Yelp. Experimental results show that the runtime reductions are almost linear to the number of mappers (**7.42** times for 10 mappers and **2.62** times for 4 mappers). The mappers take most of the computation time due to POS-tagging of massive amounts of reviews. The number of mappers and reducers can be configured on demand, so that we can handle very large datasets in a scalable fashion.

Table 3: Runtime Comparison between the Single-process Program and the Distributed MapReduce Program.

| | Single-Process (mins) | MapReduce (mins) | Reduction (times) |
|---|---|---|---|
| TripAdvisor | 297 | 40 | **7.42** |
| Yelp | 97 | 37 | **2.62** |

We aim to provide a *personalized aspect rating estimation* for each individual user on a particular item, and we have found no previously published results under this setting. Figure 4 shows the percentage of estimated aspect ratings from reviews on TripAdvisor and Yelp, with the search range of 3. The higher the similarity threshold, the lower the percentage of estimated ratings. For both datasets, the percentage is stable until the threshold is at around 0.3 and then drops gradually. The top two aspects with highest percentage in TripAdvisor are *Room* and *Service*, while in Yelp they are *Service* and *Food*. This makes sense as customers usually pay more attention to these aspects, about which they are more likely to share their opinions.

On the other hand, it is quite common for a user to comment only on a few aspects in the review. This leads to many missing ratings for aspects not mentioned in the reviews. A method to alleviate this situation is to combine all reviews on the item as one super-review like [8]. However, this will lose the advantage of providing personalized aspect rating estimation for each user.

Figure 5 shows the MAE score based on the estimated ratings for TripAdvisor, which is quite stable (around 1.0) for both similarity threshold and search range. Higher similarity threshold extracts better qualified but fewer similar adjective-noun word pairs for each aspect, which thus leads to lower MAE in general but unstable MAE for very high threshold. Setting the similarity threshold to 0.3 and the search range to 3, the MAE scores for all the aspects are around 1.0, which means that the average deviation between the human rating and the estimated rating is around 1 star.

Figure 6 shows the distributions of human ratings and estimated ratings in TripAdvisor. We can see that their distributions are quite similar, with the most ratings on 4-star and 5-star (more than 60%), and the least ratings on 1-star and 2-star (less than 20%). This confirms the performance of the semantic-based approach from the perspective of aspect ratings distribution.

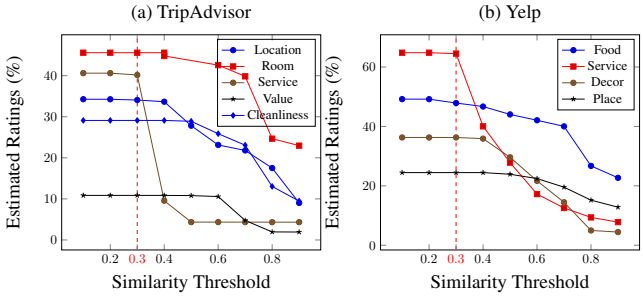


Figure 4: Percentages of Estimated Ratings for Different Similarity Threshold in TripAdvisor and Yelp.

## 5 Conclusion

This paper presents a semantic-based approach for aspect-level opinion mining from large amounts of reviews, which is an important research topic and useful for many applications, e.g., providing aspect-level review summaries to consumers for better decision making, and for product manufacturers to collect summarized user feedback. We use a semantic-based approach that applies POS-tagging on each review, extract adjective-noun word

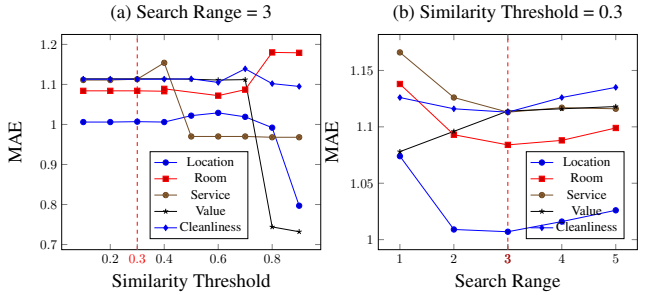


Figure 5: MAE Score on TripAdvisor for Different Similarity Threshold and Search Range.

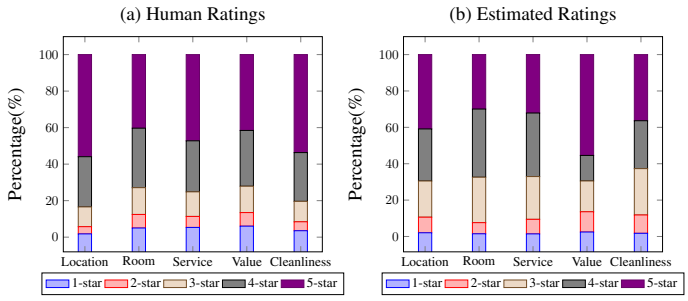


Figure 6: Distributions of Human Ratings and Estimated Ratings (similarity threshold: 0.3, search range: 3) in TripAdvisor.

pairs, aggregate sentiment scores of all the word pairs of each aspect, and finally output the aspect ratings per review given by a user on an item.

To address the scalability issue, we have implemented the approach with the MapReduce framework. Fortunately, the two sub-tasks (i.e., aspect extraction and aspect rating estimation) of aspect-level opinion mining are naturally compatible with the *map* phase and the *reduce* phase of the MapReduce framework, respectively. The challenge lies in how to design key-values to fit the MapReduce framework and parallelize the time-consuming operations. Compared with the single-process Java program, the runtime reduction by the distributed MapReduce program is almost linear to the number of mappers. The scalability advantage of the MapReduce implementation enables us to handle very large datasets by increasing the number of mappers on demand. Moreover, the semantic-based approach enables us to check and interpret the extracted phrases and their corresponding ratings for performance tuning. These phrases can also be used for related tasks, e.g., aspect-level review summaries. In addition, the semantic-based approach obtains good performance for aspect rating estimation on the TripAdvisor dataset, with the average deviation of around 1 star between the human rating and the estimated rating. The source code for the sentiment-based approach and all the experiments can be downloaded from https://github.com/ppfliu/aspect-opinion.

For future work, we shall investigate how to adapt our approach to a new domain (e.g., the laptop domain). One of the key issues to solve is how to learn ratable aspects automatically (e.g., topic modeling techniques) for a new domain, rather than having to pre-define aspects manually.

# Acknowledgements

This work is affiliated with the Big Data Decision Analytics Research Center of The Chinese University of Hong Kong.

# References

[1] S. Baccianella, A. Esuli, and F. Sebastiani. SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *LREC*, volume 10, pages 2200–2204, 2010.

[2] J. Dean and S. Ghemawat. MapReduce: Simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.

[3] C. Fellbaum. *WordNet: An electronic lexical database*. Language, Speech, and Communication. MIT Press, 1998.

[4] M. Hu and B. Liu. Mining and summarizing customer reviews. In *Proceedings of SIGKDD*, pages 168–177. ACM, 2004.

[5] W. Jin, H. H. Ho, and R. K. Srihari. A novel lexicalized HMM-based learning framework for web opinion mining. In *Proceedings of ICML*, pages 465–472. Citeseer, 2009.

[6] C. Lin and Y. He. Joint sentiment/topic model for sentiment analysis. In *Proceedings of CIKM*, pages 375–384. ACM, 2009.

[7] B. Liu. *Web data mining: Exploring hyperlinks, contents, and usage data*. Springer, 2007.

[8] B. Lu, M. Ott, C. Cardie, and B. K. Tsou. Multi-aspect sentiment analysis with topic models. In *Data Mining Workshops (ICDMW)*, pages 81–88. IEEE, 2011.

[9] M. P. Marcus, M. A. Marcinkiewicz, and B. Santorini. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993.

[10] S. Moghaddam and M. Ester. Opinion digger: an unsupervised opinion miner from unstructured product reviews. In *Proceedings of CIKM*, pages 1825–1828. ACM, 2010.

[11] S. Moghaddam and M. Ester. On the design of LDA models for aspect-based opinion mining. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 803–812. ACM, 2012.

[12] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of EMNLP*, pages 79–86. ACL, 2002.

[13] S. Shariaty and S. Moghaddam. Fine-grained opinion mining using conditional random fields. In *Data Mining Workshops (ICDMW)*, pages 109–114. IEEE, 2011.

[14] I. Titov and R. McDonald. Modeling online reviews with multi-grain topic models. In *Proceedings of WWW*, pages 111–120. ACM, 2008.

[15] Z. Wu and M. Palmer. Verbs semantics and lexical selection. In *Proceedings of ACL*, pages 133–138. ACL, 1994.