Sentiment Analysis of Product Reviews Using LDA Method based on Customer Text Content

[1.] **Dr. P. Suresh,**
HOD, Computer Science,
Salem Sowdeshwari College, Salem

[2.] **Mr.S. MuthuKumaran,**
Asst. Prof in Computer Science,
St. Joseph's College of Arts and Science (Autonomous),
Cuddalore-1

***Abstract*-This paper proposes a new approach to aspect-based sentiment analysis. The goal of our algorithm is to obtain a summary of the most positive and the most negative aspects of a specific product, given a collection of free-text customer reviews. Our approach starts by matching handcrafted dependency paths in individual sentences to find opinions expressed towards candidate aspects. Then, it clusters together different mentions of the same aspect by using a Word Net-based similarity measure. Finally, it computes a sentiment score for each aspect, which represents the overall emerging opinion of a group of customers towards a specific aspect of the product. Our approach does not require any seed word or domain-specific knowledge, as it only employs an off the-shelf sentiment lexicon. We discuss encouraging preliminary results in detecting and rating aspects from on-line reviews of books. We investigate the efficacy of topic model based approaches to two multi-aspect sentiment analysis tasks multi aspect sentence labeling and multi-aspect rating prediction**

Keywords: aspect-based sentiment analysis, opinion mining, syntactic multi-aspect sentiment analysis, topic modeling, dependency paths

1. INTRODUCTION

Sentiment analysis is the task of detecting subjectivity in natural language. Approaches to this task mainly draw from the areas of natural language processing, data mining, and machine learning. In the last decade, the exponential growth of opinionated data on the Web fostered a strong interest in the insights that sentiment analysis could reveal. For example, companies can analyze user reviews on the Web to obtain a good picture of the general public opinion on their products at very little cost. While the first efforts in sentiment analysis were directed towards determining the general polarity (positive or negative) of a certain sentence or document, the interest has recently shifted towards a more qualitative analysis, that aims to detect the different aspects of a topic towards which an opinion is expressed.

In this paper a new algorithm for automatically detecting and rating product aspects from customer reviews. Aspectator can discover candidate aspects by simply matching few syntactic dependency paths, while other approaches [6, 14, 16, 21] require seed words in input and use syntactic dependencies or some bootstrapping technique to discover new words and the relations between them. Additionally, it does not require any domain-specific knowledge in input, but only few handcrafted syntactic dependency paths and an off-the-shelf sentiment lexicon. Consequently, the proposed system can detect and rate aspects of products in any domain, while many existing approaches [16, 21, 18] focus on domains for which machine-readable knowledge is available. Concretely, Aspectator combines a first high-recall step where candidate aspects are extracted from individual sentences through syntactic dependency paths, with second and third high-precision steps, where aspect mentions are clustered and their sentiment scores are aggregated by leveraging an external sentiment lexicon.

2. RELATED WORK

While sentiment analysis has been studied extensively for some time [10], most approaches have focused on document level overall sentiment. Recently, there has been a growing interest in sentiment analysis at finer levels of granularity, and specifically approaches that take into account the multiaspect nature of many sentiment analysis tasks. Early multi-aspect work focused on creating aspect-based review summaries using mined product features [11]–[13].More recent work [14], [15] has also began modeling implicit aspects. For example, [16] develop an aspect-based review summarization system that extracts and aggregates aspects and their corresponding sentiments. Recent work has also began to look at multi-aspect rating prediction. [17] Present the Good Grief algorithm, which jointly learns ranking models for individual aspects using an online Perceptron Rank (PRank) [18] algorithm. [19] and [20] bootstrap aspect terms with seed words for unsupervised multi-aspect opinion polling and probabilistic rating regression, respectively. [21] integrate a document-level HMM model to improve both multi-aspect rating prediction and aspect-based sentiment summarization.

.

3.SENTIMENT ANALYSIS TASKS USING MULTI-ASPECT

In this work, we consider a limited version of the aspect identification and mention extraction task, which we call multi-aspect sentence labeling. In our limited setting, we assume that aspects are fixed—e.g., food, service, and ambiance for restaurant reviews—and that it is sufficient to identify Weak Supervision with Minimal Prior Knowledge: To encourage topic models to learn latent topics that correlate directly with aspects, we augment them with a weak supervised signal in the form of aspect-specific seed words. Rather than directly using the seed words to do bootstrapping, as in [19] and [20], we use them to define an asymmetric prior on the word-topic distributions. This approach guides the latent topic learning towards more coherent aspect-specific topics, A different line of work on aspect-based sentiment analysis is based on topic models. Brody and Elhadad [3] have tried to use Latent Dirichlet Allocation (LDA) [2] to extract topics as product aspects. To determine the polarity towards each topic/aspect, they start from a set of seed opinion words and propagate their polarities to other adjectives by using a label propagation algorithm. Instead of treating aspect detection and sentiment classification as two separate problems, Lin and He [11] and Jo and Oh [8] directly integrate the sentiment classification in the LDA model, so that it natively captures the sentiment towards the topic/aspect. While these LDA-based approaches provide an elegant model of the problem, they produce topics that are often not directly interpretable as aspects, and thus require manual labeling to achieve a readable output.

The work discussed so far proposes domain-independent solutions for aspect based sentiment analysis, where also our approach is positioned. However, several works make use of domain-specific knowledge to improve their results. For instance,

Thet et al. [16] focus on aspect-based classification of book reviews, and include as input for their algorithm movie-specific terms such as the name of the book, the author and the publication. Additionally, they include some domain-specific opinion words as input for their algorithm. As expected, including domain specific knowledge yields a more accurate sentiment classification. To make an example, the word “unpredictable” has a negative polarity in general English, but in the movie domain it is often used to praise the unpredictability of a storyline.Since all relevant aspects are given as input, they exclusively focus on detecting opinions towards the given aspects by (1) capturing new opinion words through syntactic dependencies, and (2) rating the product aspects based on an external

Sentiment lexicon and some given domain-specific opinion words The second phase of multi-aspect sentiment analysis is multi-aspect rating prediction [7], [17], [20], [21]—in which each aspect of a document is assigned polar (i.e., positive, negative, neutral), numeric, or “star” (i.e., 1-5) ratings.

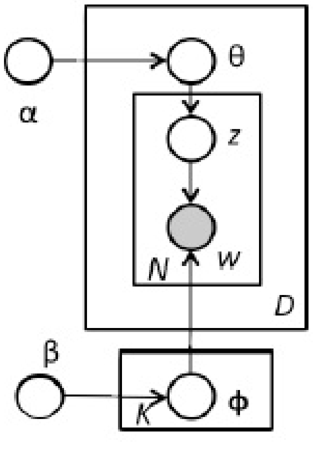
Specifically, we consider two settings: (1) multi-aspect rating prediction with indirect supervision, and (2) supervised multi-aspect rating prediction. In (1), aspect ratings are predicted based only on the text and overall rating of each review. Specifically, we train a regression model on the given overall ratings and, for each aspect, apply the model to the corresponding aspect-labeled sentences. In (2), the supervised multi-aspect rating prediction setting, we augment and compare standard supervised regression learners with features derived from unsupervised topic

4. Sentiment Analysis Using LDA

LDA, introduced by David Blei et al. [12], is a probabilistic generative topic model based on the assumption that each document is a mixture of various topics and each topic is a probability distribution over different words. A graphical model of LDA is shown in wherein nodes are random variables and edges in-dicate the dependence between nodes [12, 13]. As a directed graph, shaded and unshaded variables indicate observed and latent (i.e., unobserved) variables respectively, and arrows indicate conditional dependencies between variables while plates (the boxes in the figure) refer to repetitions of sampling Formally, each

Choose global topic proportions: _gl _ Dir(_gl)
For each sliding window v of size T:
 Choose local topic proportions: _loc
d;v _ Dir(_loc)
 Choose granularity mixture: _d;v _ Beta(_mix)
 For each sentence s:
  Choose window proportions: d;s _ Dir()
 For each word w in sentence s of document d:
Choose sliding window: vd;w _ d;s
Choose granularity: rd;w _ _d;vd;w
 Choose topic: zd;w _ f_gl; _loc d;vgrd;w
 Choose word: w _ _rd;w

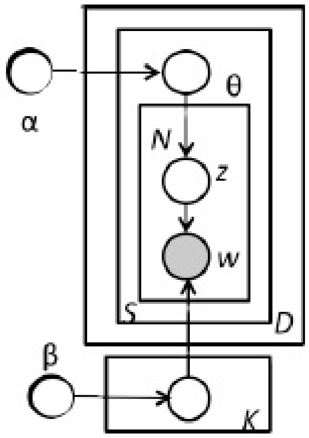
zd;wThe goal of LDA is therefore to find a set of model parameters, topic proportions and topic-word distributions. Standard statistical techniques can be used to invert the generative process of LDA, in-ferring the set of topics that were responsible for generating a collection of documents. The exact in-ference in LDA is generally intractable, and we have to appeal to approximate inference algorithms for posterior estimation. The most common approaches that are used for approximate inference are EM, Gibbs Sampling and Variational method [12, 13, 15].



(a) LDA.

Figure 1 *LDA Topic Model*

In response to limitations of standard LDA for multi-aspect work, [7] propose Multi-Grain LDA (MG-LDA). MG-LDA jointly models documentspecific themes (global topics), and themes that are common throughout the corpus intended to correspond to ratable aspects, called local topics. Additionally, while the distribution over global topics is fixed for a given document (review),local topic proportions are varied across the document according to sentence-level sliding windows



(b) Local LDA.

Figure 2. Local LDA.

Sentiment tokens and sentiment scores are information extracted from the original dataset. They are also known as features, which will be used for sentiment categorization. In order to train the classifiers, each entry of training data needs to be transformed to a vector that contains those features, namely a feature vector. For the sentence-level (review-level) categorization, a feature vector is formed based on a sentence (review). One challenge is to control each vector's dimensionality. The challenge is actually twofold: Firstly, a vector should not contain an abundant amount (thousands or hundreds) of features or values of a feature, because of the curse of dimensionality [32]; secondly, every vector should have the same number of dimensions, in order to fit the classifiers. This challenge particularly applies to sentiment tokens: On one hand, there are 11,478 word tokens as well as 3,023 phrase tokens; On the other hand, vectors cannot be formed by simply including the tokens appeared in a sentence (or a review), because different sentences (or reviews) tend to have different amount of tokens, leading to the consequence that the generated vectors are in different dimensions.

```
for every Tagged Sentences do
  for $i/i + 1$ as everyword/tag pair do
    if $i + 1$ is a Negative Prefix then
      if there is an adjective tag or a verb tag in next pair then
        NOA Phrases←$(i, i + 2)$
        NOV Phrases←$(i, i + 2)$
      else
        if there is an adjective tag or a verb tag in the pair after next then
          NOA Phrases←$(i, i + 2, i + 4)$
          NOV Phrases←$(i, i + 2, i + 4)$
        end if
```

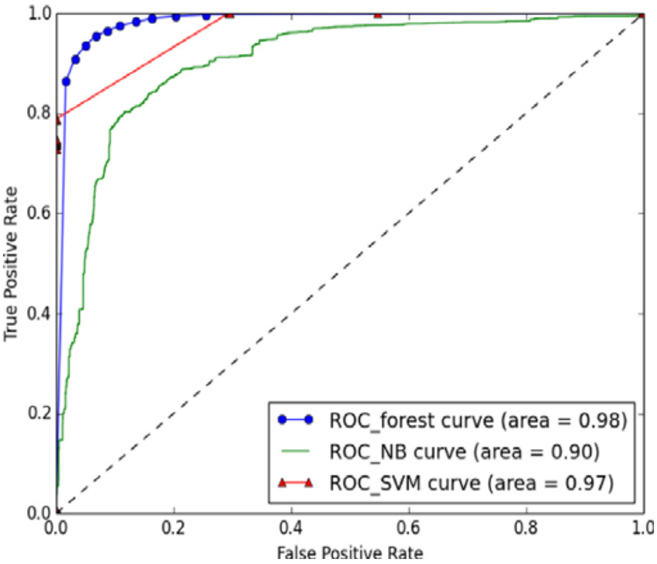
The process of sentiment polarity categorization is twofold: sentence-level categorization and review-level categorization. Given a sentence, the goal of sentence-level categorization is to classify it as positive or negative in terms of the sentiment that it conveys. Training data for this categorization process require ground truth tags, indicating the positiveness or negativeness of a given sentence. However, ground truth tagging becomes a really challenging problem, due to the amount of data that we have. Since manually tagging each sentence is infeasible, a machine tagging approach is then adopted as a solution. The approach implements a bag-of-word model that simply counts the appearance of positive or negative (word) tokens for every sentence. If there are more positive tokens than negative ones, the sentence will be tagged as positive, and vice versa. This approach is similar to the one used for tagging the Sentiment 140 Tweet Corpus. Training data for review-level categorization already have ground truth tags, which are the star-scaled ratings

5.Feature Vector Formation

Sentiment tokens and sentiment scores are information extracted from the original dataset. They are also known as features, which will be used for sentiment categorization. In order to train the classifiers, each entry of training data needs to be transformed to a vector that contains those features, namely a feature vector. For the sentence-level(review-level) categorization, a feature vector is formed based on a sentence (review). One challenge is to control each vector's dimensionality. The challenge is actually twofold: Firstly, a vector should not contain an abundant amount (thousands or hundreds) of features or values of a feature, because of the curse of dimensionality [12]; secondly, every vector should have the same number of dimensions, in order to fit the classifiers. This challenge particularly applies to sentiment tokens: On one hand, there are 11,478 word tokens as well as 3,023 phrase tokens; On the other hand, vectors cannot be formed by simply including the tokens appeared in a sentence (or a review), because different sentences (or reviews) tend to have different amount of tokens, leading to the consequence that the generated vectors are in different dimensions. Since we only concern each sentiment token's appearance inside a sentence or a review, to overcome the challenge, two binary strings are used to represent each token's appearance. One string with 11,478 bits is used for word tokens, while the other one with a bit-length of 3,023 is applied for phrase tokens. For instance, if the *i*th word (phrase) token appears, the word (phrase) string's *i*th bit will be flipped from "0" to "1". Finally, instead of directly saving the flipped strings into a feature vector, a hash value of each string is computed using Python's built-in hash function and is saved. Hence, a sentence level feature vector totally has four elements: two hash values computed based on the flipped binary strings, an averaged sentiment score, and a ground truth label. Comparatively, one more element is exclusively included in review-level vectors. Given a review, if there are *m* positive sentences and *n* negative sentences, the value of the element is Computed as: −1 × *m* + 1 × *n*.

6. Review-Level Categorization

3-million feature vectors are formed for the categorization. Vectors generated from reviews that have at least 4-star ratings are labeled as positive, while vectors labeled as negative are generated from 1-star and 2-star reviews. 3-star reviews are used to prepare neutral class vectors. As a result, this complete set of vectors is uniformly labeled into three classes, positive, neutral, and negative. In addition, three subsets are obtained from the complete set, with subset A contains 300 vectors, subset B contains 3,000 vectors, subset C contains 30,000 vectors, and subset D contains 300,000 vectors, respectively.



The experimental result is promising, both in terms of the sentence-level categorization and the review-level categorization. It was observed that the averaged sentiment score is a strong feature by itself, since it is able to achieve an F1 score over 0.8 for the sentence-level For the review-level categorization with the complete set, the feature is capable of producing an F1 score that is over 0.73. However, there is still couple of limitations to this study. The first one is that the review-level categorization becomes difficult if we want to classify reviews to their specific star-scaled ratings. In other words, F1 scores obtained from such experiments are fairly low, with values lower than 0.5. The second limitation is that since our sentiment analysis scheme proposed in this study relies on the occurrence of sentiment tokens, the scheme may not work well for those reviews that purely contain implicit sentiments. An implicit sentiment is usually conveyed through some neutral words, making judgement of its sentiment polarity difficult. For example, sentence like “Item as described.", which frequently appears in positive reviews, consists of only neutral words.

7. Results And Conclusion

With the help of the ROC curves it is clear to see that all three models performed quite well for testing data that have high posterior probability. (A posterior probability of a testing data point, *A*, is estimated by the classification model as the probability that *A* will be classified as positive, denoted as *P(+|A).*) As the probability getting lower, the Naïve Bayesain classifier outperforms the SVM classifier, with a larger area under curve. In general, the Random Forest model performs the best. Sentiment analysis or opinion mining is a field of study that analyzes people’s sentiments, attitudes, or emotions towards certain entities. This paper tackles a fundamental problem of sentiment analysis, sentiment polarity categorization. Online product reviews from Amazon.com are selected as data used for this study. A sentiment polarity categorization process (Figure 2) has been proposed along with detailed descriptions of each step. Experiments for both sentence-level categorization and review-level categorization have been performed.

Reference

1. Kim S-M, Hovy E (2004) Determining the sentiment of opinions. In: Proceedings of the 20th international conference on Computational Linguistics, page 1367. Association for Computational Linguistics, Stroudsburg, PA, USA
2. Liu B (2010) Sentiment analysis and subjectivity. In: Handbook of Natural Language Processing, Second Edition. Taylor and Francis Group, Boca
3. Liu B, Hu M, Cheng J (2005) Opinion observer: Analyzing and comparing opinions on the web. In: Proceedings of the 14th International Conference on World Wide Web, WWW ’05. ACM, New York, NY, USA. pp 342–351
4. Pak A, Paroubek P (2010) Twitter as a corpus for sentiment analysis and opinion mining. In: Proceedings of the Seventh conference on International Language Resources and Evaluation. European Languages Resources Association, Valletta, Malta
5. Pang B, Lee L (2004) A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In: Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics, ACL ’04.Association for Computational Linguistics, Stroudsburg, PA, USA
6. Pang B, Lee L (2008) Opinion mining and sentiment analysis. Found Trends Inf Retr 2(1-2):1–135
7. Turney PD (2002) Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, ACL ’02. Association for Computational Linguistics, Stroudsburg, PA, USA. pp 417–424
8. Whitelaw C, Garg N, Argamon S (2005) Using appraisal groups for sentiment analysis. In: Proceedings of the 14th ACM International Conference on Information and Knowledge Management, CIKM ’05. ACM, New York, NY, USA. pp 625–631
9. Twitter (2014) Twitter apis. https://dev.twitter.com/start
10. Liu B (2014) The science of detecting fake reviews. http://content26.com/blog/bing-liu-the-science-of-detectingfake-reviews/
11. Jindal N, Liu B (2008) Opinion spam and analysis. In: Proceedings of the 2008 International Conference on, Web Search and Data Mining, WSDM ’08. ACM, New York, NY, USA. pp 219–230
12. Mukherjee A, Liu B, Glance N (2012) Spotting fake reviewer groups in consumer reviews. In: Proceedings of the 21st, International Conference on World Wide Web, WWW ’12. ACM, New York, NY, USA. pp 191–200
13. Stanford (2014) Sentiment 140. http://www.sentiment140.com/
14. www.amazon.com

15. Go A, Bhayani R, Huang L (2009) Twitter sentiment classification using distant supervision. CS224N Project Report, Stanford. pp 1–12

16 Lin Y, Zhang J, Wang X, Zhou A (2012) An information theoretic approach to sentiment polarity classification. In: Proceedings of the 2Nd Joint WICOW/AIRWeb Workshop on Web Quality, WebQuality '12. ACM, New York, NY, USA. pp 35–40

17 Sarvabhotla K, Pingali P, Varma V (2011) Sentiment classification: a lexical similarity based approach for extracting subjectivity in documents. Inf Retrieval 14(3):337–353

18 Wilson T, Wiebe J, Hoffmann P (2005) Recognizing contextual polarity in phrase-level sentiment analysis. In: Proceedings of the conference on human language technology and empirical methods in natural language processing. Association for Computational Linguistics, Stroudsburg, PA, USA. pp 347–354

19. Yu H, Hatzivassiloglou V (2003) Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In: Proceedings of the 2003 conference on, Empirical methods in natural language processing. Association for Computational Linguistics, Stroudsburg, PA, USA. pp 129–136

20. Zhang Y, Xiang X, Yin C, Shang L (2013) Parallel sentiment polarity classification method with substring feature reduction. In: Trends and Applications in Knowledge Discovery and Data Mining, volume 7867 of Lecture Notes in Computer Science. Springer Berlin Heidelberg, Heidelberg, Germany. pp 121–132

21. Zhou S, Chen Q, Wang X (2013) Active deep learning method for semi-supervised sentiment classification. Neurocomputing 120(0):536–546. Image Feature Detection and Description

22. Chesley P, Vincent B, Xu L, Srihari RK (2006) Using verbs and adjectives to automatically classify blog sentiment. Training 580(263):233

23 Choi Y, Cardie C (2009) Adapting a polarity lexicon using integer linear programming for domain-specific sentiment classification. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2, EMNLP '09. Association for Computational Linguistics, Stroudsburg, PA, USA. pp 590–598

24 Jiang L, Yu M, Zhou M, Liu X, Zhao T (2011) Target-dependent twitter sentiment classification. In: Proceedings of the 49th, Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1. Association for Computational Linguistics, Stroudsburg, PA, USA. pp 151–160

25. Tan LK-W, Na J-C, Theng Y-L, Chang K (2011) Sentence-level sentiment polarity classification using a linguistic approach. In: Digital Libraries: For Cultural Heritage, Knowledge Dissemination, and Future Creation. Springer, Heidelberg, Germany. pp 77–87

26. Liu B (2012) Sentiment Analysis and Opinion Mining. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers

27. Hu M, Liu B (2004) Mining and summarizing customer reviews. In: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, New York, NY, USA. pp 168–177

28. Gann W-JK, Day J, Zhou S (2014) Twitter analytics for insider trading fraud detection system. In: Proceedings of the sencond ASE international conference on Big Data. ASE

29 Roth D, Zelenko D (1998) Part of speech tagging using a network of linear separators. In: Coling-Acl, The 17th International Conference on Computational Linguistics. pp 1136–1142

30. Kristina T (2003) Stanford log-linear part-of-speech tagger. http://nlp.stanford.edu/software/tagger.shtml

31. Marcus M (1996) Upenn part of speech tagger. http://www.cis.upenn.edu/~treebank/home.html

32. Han J, Kamber M, Pei J (2006) Data Mining: Concepts and Techniques, Second Edition (The Morgan Kaufmann Series in Data Management Systems), 2nd ed. Morgan Kaufmann, San Francisco, CA, USA

33 (2014) Scikit-learn. http://scikit-learn.org/stable/