Mining Product Opinions with Most Frequent Clusters of Aspect Terms

Chukwuma Ejieh
Watson Financial Services, IBM Canada,
Toronto, ON
ejiehc@ca.ibm.com

C.I. Ezeife*
School of Computer Science,
University of Windsor,
Windsor, ON N9B 3P4
cezeife@uwindsor.ca

Ritu Chaturvedi
School of Computer Science,
University of Guelph,
Guelph, ON N1G 2W1
chaturvr@uoguelph.ca

ABSTRACT

This paper addresses the problem of more accurately mining product aspect opinions from Twitter posts, in the presence of spam and noisy posts, by proposing an algorithm called Microblog Aspect Miner (MAM). MAM takes a three step approach of classifying the microblog posts into subjective and objective posts using opinion scores of words from SentiWordNet. MAM then represents frequent nouns of subjective posts as vectors in such a way that nouns semantically similar to the products have a similar vector value using the WordVec algorithm. K-Means clustering algorithm is used to obtain the cluster of aspects relevant to the product to separate the noisy aspects so that the most relevant aspects are ranked using proposed Aspect-Product Similarity Threshold based on cosine similarities. Experiments show that this improves accuracy of obtaining relevant aspects of products from microblog posts in comparison to such existing aspect based opinion mining (ABOM) systems as Twitter Aspect Classifier (TAC).

CCS CONCEPTS

• **Information systems → Enterprise applications**; **Content analysis and feature selection**; *Association rules*;

KEYWORDS

Aspects, Opinion Mining, Data Mining, Sentiment Analysis, Text Mining, Social Media, Microblogs, Spam post exclusion, Information Retrieval

ACM Reference Format:
Chukwuma Ejieh, C.I. Ezeife, and Ritu Chaturvedi. 2019. Mining Product Opinions with Most Frequent Clusters of Aspect Terms. In *Proceedings of ACM SAC Conference (SAC'19).* ACM, New York, NY, USA, Article 4, 4 pages. https://doi.org/doi.org/10.1145/3297280.3297550

1 INTRODUCTION

Aspect-based Opinion Mining (ABOM) [2] focuses on recognition of opinions about a product and the aspects to which they refer. ABOM systems take as input a body of text about a product and outputs all the aspects or features of the product from the body of text and its respective opinion polarity. Examples of ABOM systems are Red Opal and Opinion Digger Moghaddam and Ester [8].

The major steps taken in ABOM are: (1) Post Collection: Obtain a collection of posts about a product from product review sites, news articles, blogs and any other online forums. In this research, Microblogs (Twitter short posts) are the source. (2) Candidate Aspect Collection: This is mainly done using association rule mining to get the frequently occurring words as candidate aspects of a product [6]. (3) Candidate Aspect Pruning: Here, the "true" aspects of the product are found. Existing systems use techniques like compactness pruning, TF-IDF and PMI to narrow down the candidate aspects to relevant aspects that are parts or features of the product. This step is where this paper's study provides an improvement on. (4) Aspect Opinion Detection – the opinion polarity of the discovered aspects is obtained at this stage. The huge volume of microblog posts that mention some companies' names collected from Twitter on the 9th of September 2015 within a 12-hour window is 115,000 posts mentioned Apple, 60,000 Google, 44,000 Microsoft and 39,000 Samsung showing the importance of this work.

1.1 Contributions and Outline

The main contributions of this paper are: To mine the aspects of a product in such a way that frequent "noise" in microblog posts does not affect the accuracy of the system. Noise includes: (i) spam - posts, (ii) advertisements, (iii) Buzz posts that are frequently repeated due to a popular event or happening, e.g., "Obama call with an iphone", (iv) Competitor's Products: posts that mention competitor's products along with the product. e.g., "Samsung upgrades Android on galaxy in light of iPhone release". This paper contributes to solving this problem by proposing an algorithm called Microblog Aspect Miner (MAM) which takes in raw microblog posts about a product as the input and outputs the relevant aspects of the products and their opinion polarity.

1.2 Outline

Section 2 provides relevant related work. Section 3 proposes a new Microblog Aspect Miner (MAM), section 4 provides example application of MAM to a problem. Section 5 reports the results of our experiments and section 6 provides concluding remarks and future work.

2 RELATED WORK

Existing ABOM systems include Red Opal [9] that identifes features and assigns a score to each feature or aspect of a product. Li

*This research was supported by the Natural Science and Engineering Research Council (NSERC) of Canada under an Operating grant (OGP-0194134) and a University of Windsor grant.

SAC'19, April 8-12, 2019, Limassol, Cyprus

ACM ISBN 978-1-4503-5933-7/19/04.
https://doi.org/doi.org/10.1145/3297280.3297550

and Li [5] proposed a Twitter Summarization Framework that uses TF-IDF to extract aspects of products. Opinion Digger [8] uses an approach similar to Hu and Li [3] for feature extraction and suffers similar limitations. Spina et al. [10] worked on extracting words from microblogs that are relevant to the product by combining the following formulae: TF-IDF (term frequency - inverse document frequency), LLR (log likelihood ratio) and PMI. TAC (twitter aspect classifier) [4] uses PMI to obtain opinion polarity of microblog posts about a product. TAC uses Google search to find relevant aspects of a product and to calculate PMI and that is a serious limitation since Google searches vary by geographic locations and by user. Existing systems suffer from limitations that impact the performance and accuracy of extracting aspects from microblogs.

3 THE PROPOSED MICROBLOG ASPECT MINER (MAM)

The words that are closest to the product will fall into the cluster containing the aspects of the product. The proposed Microblog Aspect Miner (MAM) algorithm accepts input data of an entity product name that serves as a search query to the Twitter API and produces the output of a ranked list of relevant aspects of the product. MAM has 3 main steps:
1. MAM collects all Twitter posts, P about a product, e within a period of time. MAM calls the preprocessing module with P to remove all noisy posts from P and keep only the Subjective Posts, Sp
2. MAM calls the Aspect Mining Module (AMM) with Sp to obtain the relevant aspects of the Product, P
3. MAM calls the Aspect Opining Mining Module to get the opinion polarity of the obtained relevant aspects.

3.1 Step 1 of MAM: Preprocessing of Posts to Produce Subjective Posts

This pre-processing module step cleans up the input posts to remove every text that is not likely to be an opinion such as URL links (that are for advertisements, spam, news headlines etc.), texts (i) beginning with ‘RT’ that are retweets, the foreign characters (ii) ‘@’ that may be for user names, (iii) ‘#’ that may be for hash tag. Then, a subjectivity module is introduced to detect if a post expresses an opinion or not and to collect only subjective posts. Posts that do not express opinions are discarded. Subjective posts express a positive or negative opinion while an objective post does not express any opinion. For example, ‘I love the earth’ is a subjective statement but ‘The earth is round’ is an objective statement. The subjectivity module takes in pre-processed tweets and uses opinion scores from SentiWordNet [1] to obtain the subjective posts. SentiWordnet is a lexicon used for opinion mining in which each word is assigned a positive, negative and objective score. For example, on SentiWordNet ‘Happy’ has a positive score of 0.875 and a negative score of 0.0 and an objective score of 0.125. Only the positive and negative scores are used by the subjectivity module to determine that a post is subjective if the average of total of the negative and positive scores of all its words is greater than 0.5 as outlined below.
Steps to Compute Subjective Posts (SP) from Cleaned Posts:
1. FOR post, p in cleaned posts CP, DO:
2. Get the positive score and negative score of each word, w from SentiWordNet.
3. Get the subjective score (SSc) for p, SSc_p as the average of the subjective scores of all the words(w) in post (p) having number n words. The formula for getting the score SSc of post p is:
$SSc_p = (\sum_{i=1}^{n} w_i's(positivescore + negativescore))/n$
4. IF the subjective score $SSc_p \geq 0.05$, then post p is addeed to the collection of subjective posts, Sp.

3.2 Step 2 of MAM: Obtaining Aspects of the Product

An Aspect Mining Module (AMM) used to obtain the aspects of the product from subjective posts (with noisy microblog posts removed.) is presented next.
Aspect Mining Module (AMM) for Mining Aspect Terms from Subjective Posts
1. FOR each post, p in the subjective post SP DO:
2. Tokenize p, remove stop words with tools like Twokenizer.
3. POS (Part-of-Speech) tag each word, w in p using NLTK POS tagger.
4. Obtain all w with Noun and Plural Noun POS tags in p as nouns.
5. Apply Appriori frequent pattern mining algorithm on nouns.
6. Mine frequent nouns as words with minimum support of 1%
7. Build the language model as: FOR each word, w in frequent nouns, obtain its vector representation using word2vec algorithm [7]. /* The purpose of the language model is to obtain the vector representations of words. Each word is represented by a 200-dimension vector (moderate word embedding size to capture semantic similarity) in such a way that words that have similar meanings have similar vector representations. For example, the vector representation of water will be more similar to the vector representation of liquid than the vector representation of house. The steps taken are:
i. Obtained 2 Billion tweets from the Stanford NLP Group. ii. The corpus from the step above is used as input in the word2vec algorithm [7] which was implemented with the Genism Toolkit. A language model which contains 1.2 million unique words mapped to vectors of dimension 200 each is obtained as the output. The vector representations of the frequent nouns are looked up in the language model and obtained. In lines 7-9, each frequent noun is checked to see if it is in the vocabulary of the language model, if the similarity between it and the product is greater than 0.4 and if it is longer than 2 characters in length. The frequent nouns that do not meet all these 3 criteria are dropped. In line 9, we obtain the vector form of the remaining frequent nouns. At this stage, each word is represented as a vector of length 200. */
Thus, DO:
8. IF word w is in vector V (Vocabulary of words in our language model) and Similarity (product, w) (i.e. 1- Cosine Similarity(product, w)) is greater than 0.4 THEN:
9. w is a candidate aspect of product, e.
10. Obtain vector representation of w.
11. Add vector representation of w to the collection of candidate aspects.
12. Select 2 arbitrary points (words) including e in candidate aspects as centers.

13. Calculate the Euclidean distance between each word w in candidate aspects and the centers using the equation: $Euclidean - distance(x, y) = \sqrt{(\sum_{i=1}^{n}(X_i - y_i)^2)}$
14. Assign w to the nearest center to form a cluster, c.
15. For every cluster, c obtain the center by getting the mean of all w in the cluster.
16. Repeat steps 13-15 until centers have a constant value.
To obtain the relevant aspects of a product from the cluster, we introduce a term called the Aspect-Product Similarity Threshold (APSM). This is the threshold at which the cosine similarity between a product and its aspect falls. From experiments, this threshold is observed to be 0.7. Candidate aspects that are above this threshold are mostly competitor's products or parent companies of the product and are therefore not treated as aspects of the product. Thus, to get the relevant aspects of the product, we select every word in the candidate aspect that falls below the APST threshold. The relevant aspects are also ranked using the cosine similarity. The closer a candidate aspect is to the APSM, the higher it is ranked as an aspect of the product.

3.3 Step 3 of the MAM Algorithm: Obtaining the Opinion Polarity of the Aspects

The Aspect Opinion Mining module is used to obtain the polarity of the aspects of the products from the Tweets. The opinion polarity are classified as positive, negative or neutral using the Aspect Opinion Mining algorithm (AOM) which accepts the aspect terms as its input and returns the opinion on that product aspect as either positive, negative or neutral based on the percentage of tweets that held positive, negative or neutral opinion on that aspect. The summary of the AOM algorithm is given next. The verb or adjective to the discovered aspect in the microblog post is explored by comparing it with a set of predefined positive words and negative words. If the verb or adjective is more similar to the set of positive words, the aspect is classified as positive in that particular post, otherwise if it more similar to negative words, it is classified as negative. If the similarity is equal (up to 4 decimal places), the aspect is classified as neutral. Also, this is toggled if a 'not' is found just before the adjective or verbs. The number of posts that the aspect was classified as positive, negative or neutral are summarized to get the percentage polarity of the aspect.

4 AN EXAMPLE APPLICATION OF PROPOSED MAM ON TWITTER POSTS

Assuming we have a collection of 300,000 microblog posts about the iPhone, the task is to obtain the aspects of iPhone that people are frequently talking about as well as the overall opinion of people on each of the aspects. Table 1 shows a few randomly selected microblog posts used for discussion. Step 1 of the main MAM algorithm will clean the original posts in Table 1 to remove url and foreign characters and stop words to generate the cleaned posts. Step 2 obtains the subjective posts by running the preprocessed posts through the subjectivity post computation Algorithm. For our small sample data, the result of this module is the same as the subjective posts.
Step 3 of MAM system, tokenizes first to obtain:
1 'cant', 'conect', 'iphone', '6', 'android', 'moto', '360', '.', 'help', 'please', ':'
2 'iphone', '6', 'pain', 'phone', 'cases', 'i',
3 'definitely', 'need', 'get', 'iphone', 'screen', 'fixed'

SN	Microblog Posts (Tweets)
1	@Android i cant conect my iphone 6 with the android moto 360. Help me please.
2	iPhone 6 are a pain for phone cases ÃřÅÿËIJâĂŽ I
3	Definitely need to get this iPhone screen fixed!!

Table 1: Sample Microblog Posts

STEP 4: Each of the word tokens in the subjective posts are assigned a part-of-speech tag (POS Tag) and the Nouns and Plural Nouns are chosen. Some of the nouns and plural nouns in the sample subjective posts and their frequency of occurrence found after this step are: 'iphone':10, 'phone':3, 'help':1. STEP 5: In this step, we prune off the list of nouns by selecting only nouns that occur with a minimum support of 1% in the subjective posts as our frequent nouns. Some of the semantic similarity between each frequent noun and the entity are: (help:0.3306), (iphone: 1.0000), (screen: 0.5685), (periscope: -0.0737). Our frequent noun list becomes: {battery, back, lol, iphone, get, screen, phone, cases, Android, charger}.
STEP 6: We apply K-Means clustering algorithm to this pruned frequent noun list to divide them into two clusters: Cluster 1 = {get, back, lol} Cluster 2 = {android, cases, iphone, phone, screen, battery, charger}. We select the cluster 2 that has the entity term (iphone in this case) as our candidate aspects.
STEP 7: To obtain the relevant aspects of the iphone, we drop any word in the candidate aspect that do not fall below the Aspect-Product Similarity Threshold. For example, the words iphone, android and phone with APST score higher than the threshold of 0.7 are pruned. Therefore, the Aspect Mining Module gives the following with APST scores below 0.7 as the aspects of the entity, iPhone: {screen, charger, battery, cases}. These are ranked according to their similarity with the iphone.
STEP 8: Using the discovered aspects, the next step is to get the opinions of people on each of these discovered aspects to know if they are positive, negative or neutral by running them through the Aspect Opinion Mining (AOM) module. We look up the subjective posts to get the posts in which these discovered aspects were mentioned following summary of the opinions of each aspect as the final output. SN Frequent Nouns Similarity with Entity
1. cases Negative (100%); Positive (0%); Neutral (0%)
2. screen Negative (100%); Positive (0%); Neutral (0%)
3. battery Negative (100%); Positive (0%); Neutral (0%)
4. charger Negative (100%); Positive (0%); Neutral (0%)

5 EXPERIMENTAL AND PERFORMANCE ANALYSIS

The data set consists of over 300,000 tweets from 4 products and brands from different product categories as our text corpus. The products are iphone, starbucks, xbox and sony. We obtained English tweets from Twitter over a period of 1 month and test our

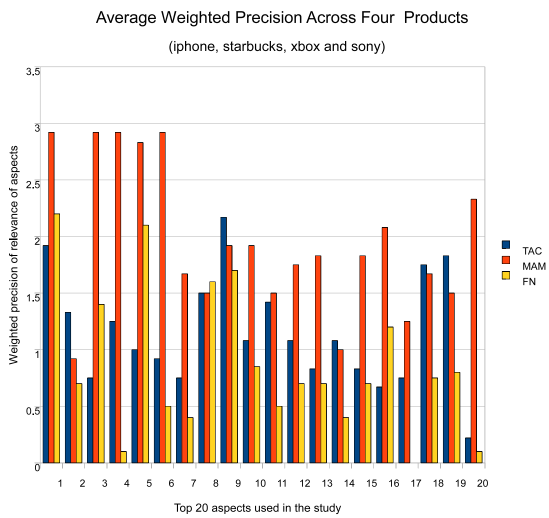

Figure 1: Average Weighted Precision for Four Products

MAM system against other Aspect-based Opinion Miners (i) TF-IDF [10], (ii) TAC [4]. We implemented these systems to get the aspects of the products from collected microblog posts about the products. We gave five human judges who know the products and have used them before to rate how relevant the aspects are to the products on a scale of 0 to 3. The rating scale is as follows: 0 (not a relevant aspect of the product), 1 (vaguely relevant to the product), 2 (slightly relevant aspect of the product), 3 (relevant aspect of the product). We compute the agreement of the judges using the Cohen's and Fleiss' Kappa scores to see if the rating of the 5 judges are in agreement. For all five products, the computed Fleiss, Cohen and kappa's scores suggest that the 5 judges are in moderate agreement about their decision on each aspect. We evaluate the top 20 most relevant aspects (that have a score of 3 by the judges) for each of the systems using the following evaluation metrics:
Precision: This measures the amount of relevant aspects that were among the top 20 aspects.
Weighted Precision: This measures the relevancy of the aspect to the product based on the scoring of the judges where R(a) is the average rating given by the judges for each aspect, a.
Weighted Precision = $(\sum_1^k R(a))/k$, where k is 20.

5.1 Results

The average weighted precision was got using our proposed solution (MAM), TAC which uses the PMI method [4] and just by counting the top 20 most frequent nouns (FN) for each aspect which serves as the baseline [10]. The results were averaged across the 4 products and are shown in Figure 1. It can be seen that all the methods did quite well in choosing a relevant aspect of the products as the first aspect. However, 70% of the aspects predicted by FN had a weighted precision score of less than 1. MAM shows very good accuracy in predicting relevant top 6 aspects and the accuracy drops after then. Only one of the predicted aspects had a weighted precision that was less than 1. MAM outperforms TAC on average in predicting the top 20 aspects across the 4 products. Furthermore, defining 'relevant aspects' of products as aspects that were given a perfect score of 3 by the judges, the number of relevant aspects obtained for each dataset for the three algorithms (TAC, MAM and FN) are respectively: (1, 8, 1) for iphone, (4, 11, 4) for starbucks, (2, 4, 2) for Xbox and (6, 3, 2) for sony. This indicates that the MAM at the middle identifies most relevant aspects that the judges identified.

6 CONCLUSIONS AND FUTURE WORK

This paper proposes Microblog Aspect Miner (MAM) for mining aspects of products from microblogs by dealing with the noisy posts in microblogs and using a variety of data mining, text mining and information retrieval approaches including classification, frequent pattern mining and clustering approaches to get relevant aspects of a product. Previous research have considered this problem but their accuracy in determining the aspects of a product is affected by spam posts and also they do not explore the semantic similarity. Experimental results show that the proposed technique performs better in terms of accuracy of getting the relevant aspects of a product. Furthermore, obtaining the opinions on these discovered aspects can give business owners an insight into what customers think of their business. Future work may consider: improving on methods for setting various thresholds done mostly through experimentations, extending this approach of ABOM to other languages, to cluster multi-word aspects (e.g. hard disk), aggregation of all posts on Microblogs, Blogs, News Articles and Product Reviews, and rating different aspects of similar products using posts from twitter to aid customers in making better purchasing decisions.

REFERENCES

[1] Andrea Esuli and Fabrizio Sebastiani. 2007. SentiWordNet: a high-coverage lexical resource for opinion mining. *Evaluation* (2007), 1–26.
[2] Ronen Feldman. 2013. Techniques and applications for sentiment analysis. *Commun. ACM* 56, 4 (2013), 82–89.
[3] Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 168–177.
[4] Hsiang Hui Lek and Danny CC Poo. 2013. Aspect-based Twitter sentiment classification. In *Tools with Artificial Intelligence (ICTAI), 2013 IEEE 25th International Conference on*. IEEE, 366–373.
[5] Yung-Ming Li and Tsung-Ying Li. 2013. Deriving market intelligence from microblogs. *Decision Support Systems* 55, 1 (2013), 206–217.
[6] Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies* 5, 1 (2012), 1–167.
[7] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.
[8] Samaneh Moghaddam and Martin Ester. 2012. Aspect-based opinion mining from product reviews. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*. ACM, 1184–1184.
[9] Christopher Scaffidi, Kevin Bierhoff, Eric Chang, Mikhael Felker, Herman Ng, and Chun Jin. 2007. Red Opal: product-feature scoring from reviews. In *Proceedings of the 8th ACM conference on Electronic commerce*. ACM, 182–191.
[10] Damiano Spina, Edgar Meij, Maarten De Rijke, Andrei Oghina, Minh Thuong Bui, and Mathias Breuss. 2012. Identifying entity aspects in microblog posts. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*. ACM, 1089–1090.