

Aspect Sentiment Model for Micro Reviews

Reinald Kim Amplayo

Yonsei University

Seoul, South Korea

Email: rktamplayo@yonsei.ac.kr

Seung-won Hwang

Yonsei University

Seoul, South Korea

Email: seungwonh@yonsei.ac.kr

Abstract—This paper aims at an aspect sentiment model for aspect-based sentiment analysis (ABSA) focused on micro reviews. This task is important in order to understand short reviews majority of the users write, while existing topic models are targeted for expert-level long reviews with sufficient co-occurrence patterns to observe. Current methods on aggregating micro reviews using metadata information may not be effective as well due to metadata absence, topical heterogeneity, and cold start problems. To this end, we propose a model called Micro Aspect Sentiment Model (MicroASM). MicroASM is based on the observation that short reviews 1) are viewed with sentiment-aspect word pairs as building blocks of information, and 2) can be clustered into larger reviews. When compared to the current state-of-the-art aspect sentiment models, experiments show that our model provides better performance on aspect-level tasks such as aspect term extraction and document-level tasks such as sentiment classification.

I. INTRODUCTION

Recently, the number of short reviews is increasing as most people write concise and brief reviews, while only few enthusiasts write long and expertly-written reviews. Following this phenomenon, some websites¹ are already implementing a rule forcing users to limit their reviews to a specified number of characters, usually the same as a tweet with 140 characters. With the increase of short reviews, it is necessary to create models specific to them.

Our research goal is thus, to holistically understand the opinions of the general people, yet in the quality of expert reviews, by performing an aspect-based sentiment analysis (ABSA) on short reviews. Specifically, we pursue LDA-based approaches [1], which are previously used as a semi-supervised technique to answer several ABSA tasks [2], [3], [4]. LDA-based approaches are advantageous over other approaches since they only require a small sentiment lexicon and do not require labelled training data, unlike other recent deep learning-techniques [5], [6], [7]. However, conventional topic models are not fit, as they are designed to be used on documents that are long enough to extract meaningful discoveries. Empirical studies also show that these models do not perform well on short texts when experimented on classification and prediction problems [8]. This is because, when fed with short texts, topic models suffer from data sparsity due to the lack of co-occurrence patterns [9].

Current solutions are proposed to remedy the problem of aspect sentiment models by a simple aggregation or pooling of short documents, based on meta-information available on documents such as time slices [10] and hashtags [11]. However, there are three major problems in current methods. First, document metadata such as users and hashtags are not always available. Reviews seldom use hashtags and there are lots of anonymous users. Second, documents aggregated using these approaches tend to be topically heterogeneous [12]. This is because the assumptions made by aggregation techniques frequently violate the topic consistencies assumed by the model. Lastly, there are cold start entities without enough documents to be pooled into one large document. This makes the documents of these users less informative and may be disregarded by the topic model.

Different from normal-sized reviews, short reviews have several properties. We observe two properties on short reviews. First, we observe that short reviews consist of pairs of sentiment word and aspect word. We call these pairs **sentiment-aspect pairs**. For example, in a review “*Unprofessional service and indifferent staff. Disappointed because of nice ambience.*”, we can extract sentiment-aspect pairs *unprofessional service*, *indifferent staff*, *disappointed staff*, and *nice ambience*². Instead of generating words a word at a time, modeling sentiment-aspect pairs results to a **higher coherence**. This means that words in sentiment-aspect pairs will appear with much higher probabilities in the same aspect distribution. This improves the performance of the model on aspect-level tasks.

Second, we observe that short reviews can be grouped into clusters based on their aspect distributions. One can think of the clusters as large and unorganized list of words, which when properly organized, can be reconstructed into expertly-written reviews. For example, the sentences “*Despite the bad food, I liked the atmosphere and the kind waiters.*” and “*The pasta was really bad. They treated us like kings, though.*” can be grouped together into one cluster because of their similar positive sentiments on atmosphere and service, and negative sentiment on food. This may result into having the reviews in the cluster **share information**. This improves the detection of aspect and sentiment proportions in the document, which in turn improves the performance of the model on document-

¹Naver Movies (URL: <http://movie.naver.com/>) and Douban Movies (URL: <https://movie.douban.com/>)

²We note that a sentiment word is not restricted to adjectives as shown in the examples; this includes opinionated verbs and negated words as well.

TABLE I: Meanings of the notations in MicroASM

D	# of documents
C	# of clusters
N	# of words in a document
P	# of pairs in a document
S	# of sentiments
T	# of topics
c	cluster
w	word
s	sentiment
z	aspect
θ	aspect multinomial distribution of a cluster
ϕ	word multinomial distribution of an aspect-sentiment pair
π	sentiment multinomial distribution of a cluster
ψ	cluster multinomial distribution
α	Dirichlet prior for θ
β	Dirichlet prior for ϕ
γ	Dirichlet prior for π
δ	Dirichlet prior for ψ

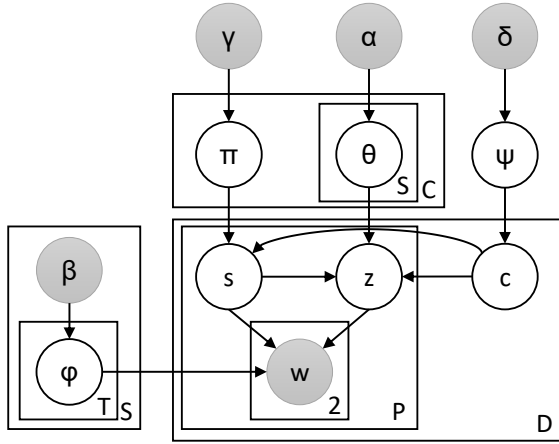


Fig. 1: Graphical representation of MicroASM

level tasks.

II. MICRO ASPECT SENTIMENT MODEL

Based on these observations, we present Micro Aspect Sentiment Model (MicroASM). We provide the source code of the model in the link for reproducibility: <https://github.com/rktamplayo/MicroASM>. The model is shown graphically in Figure 1. Notations are summarized in Table I. Instead of generating just one word at a time, MicroASM generates pairs of words using both the aspect and the sentiment latent variables z and s . Note that we include not only sentiment-aspect pairs but also all kinds of word pairs (i.e. sentiment-sentiment pairs and aspect-aspect pairs among others). This addition helps in two ways. First, it removes the necessity of doing aspect word and sentiment word detection beforehand. Second, modeling sentiment-sentiment and aspect-aspect pairs enhances sentiment and aspect learning, respectively. Moreover, outside the document layer, MicroASM introduces a new cluster layer. The cluster layer holds the sentiment and aspect distributions π and θ , respectively. In order to connect

both document and cluster layer, MicroASM introduces a new cluster latent variable c inside the document layer. Pairs of words are then generated using both z and s , both of which depend on c . Following the traditional aspect sentiment models [2], [3], we use prior sentiment lexicons by encoding it to the Dirichlet prior β . This leads to an assymmetric β vector that controls the sentiment of the word distribution.

A. Generative process

In MicroASM, pairs of words are generated using both the aspect and the sentiment latent variables z and s respectively. Both aspect and sentiment variables are dependent to the cluster variable c . The probability of a pair of words w_1 and w_2 in document d is given in Equation 1. Using the ideas on factorization and parameterization described above, we do the same as shown in Equation 2.

$$\begin{aligned}
 P(w_1, w_2) &= \sum_c \sum_s \sum_z P(c|d)P(s, z|c)P(w_1|s, z)P(w_2|s, z) \quad (1) \\
 &= \sum_d \sum_c \sum_s \sum_z \psi^{(d)} \pi^{(c)} \theta^{(c,s)} \phi_{w_1}^{(s,z)} \phi_{w_2}^{(s,z)} \quad (2)
 \end{aligned}$$

B. Inference

We use collapsed Gibbs sampling [13] to estimate the latent variables θ , ϕ , π , and ψ . We first sample the clusters of the documents. At each transition step of the Markov chain, the cluster c of the d th document is chosen according to the conditional probability shown in Equation 3, where l is the currently sampled cluster. The variable N_{ab}^{AB} represents the number of $a \in A$ and $b \in B$ assignments and the variable N^{AB} represents the sum of the number of $a \in A$ and $b \in B$ assignments for all possible a and b , excluding the current d th document.

$$\begin{aligned}
 P(c_d = l | rest) &\approx \frac{N_l^C}{N^C + C\delta} \frac{\prod_s \prod_{x=1}^{N_{ds}^{DS}} (N_{ls}^{CS} + \gamma + x - 1)}{\prod_{y=0}^{N^{DS}} (N^{CS} + S\gamma + y - 1)} \\
 &\frac{\prod_s \prod_z \prod_{x=1}^{N_{dsz}^{DST}} (N_{lsz}^{CST} + \theta + x - 1)}{\prod_{y=0}^{N^{DST}} (N^{CST} + T\theta + y - 1)} \quad (3)
 \end{aligned}$$

We then sample the aspect and sentiment of the word pairs. At each transition step of the Markov chain, the sentiment s and the aspect z of the i th word pair are chosen according to the conditional probability shown in Equation 4, where j and k are the currently sampled sentiment and aspect, respectively. The variables N_{ab}^{AB} and N^{AB} are similar to the description above, but now excluding the current i th word pair.

$$\begin{aligned}
 P(s_i = j, z_i = k | rest) &\approx \frac{N_{cjk}^{CST} + \alpha}{N^{CST} + T\alpha} \frac{N_{cj}^{CS} + \gamma}{N^{CS} + S\gamma} \\
 &\frac{N_{jkw_1}^{STW} + \beta_{sw_1}}{N^{STW} + \beta_s} \frac{N_{jkw_2}^{STW} + \beta_{sw_2}}{N^{STW} + \beta_s + 1} \quad (4)
 \end{aligned}$$

After inference is done, the approximate probability of sentiment s in document d is shown in Equation 5. The approximate probability of aspect z for sentiment s in document d is shown

in Equation 6. Finally, the approximate probability of a word w in sentiment s and aspect z is shown in Equation 7.

$$P(s|d) = \frac{N_{ds}^{DS} + \gamma}{N^{DS} + S\gamma} \quad (5)$$

$$P(z|s, d) = \frac{N_{dsz}^{DST} + \alpha}{N^{DST} + T\alpha} \quad (6)$$

$$P(w|s, z) = \frac{N_{szw}^{STW} + \beta_{sw}}{N^{STW} + \beta_s} \quad (7)$$

III. EXPERIMENTAL SETUP

In our experiments, we address the following research questions:

- Does MicroASM perform well on aspect-level ABSA tasks such as aspect term extraction? (Section IV-A)
- Does MicroASM perform well on document-level ABSA tasks such as sentiment classification? (Section IV-B)

A. Data

We use two kinds of datasets: Yelp³ and Naver⁴ reviews dataset. The Yelp dataset consists of English reviews on restaurants and shops. Since the Yelp dataset also contains longer reviews, we filter out reviews that have more than 140 characters. We choose three categories based on the amount of data available and diversity: Nightlife, Restaurant, and Shopping. The Naver dataset consists of Korean movie reviews. Reviews in this dataset are naturally short, since the website limits the users to write only 140 characters. We choose three genres based on the amount of data available and diversity: Crime, Comedy, and Fantasy. Statistics of the data is shown in Table II. We do basic preprocessing to the data by lemmatizing and retaining only nouns, verbs, adjectives, and adverbs. Stanford CoreNLP⁵ and Komoran⁶ are used as preprocessing tools for English and Korean texts, respectively. We also handle negations by adding a negating prefix (e.g. *not_*) in the neighbors (set to a window= 5) of a negating word. We also get the review ratings of each dataset for testing. We assume that reviews with less than 5 (or 3 for Yelp) stars are negative reviews, and positive otherwise. Note that we do not use this information during the training of the model.

As an aspect sentiment model, our models need a sentiment lexicon to propagate the inference of the sentiment variable. In our model, we use *Paradigm+* which consists of the sentiment oriental paradigm words from the original *Paradigm* [14] such as *good* and *bad*, and additional general affective and evaluative words such as *love* and *hate* [3]. For our Korean dataset, we use a Korean-translated *Paradigm+*. The English and Korean *Paradigm+* list is listed in Table III. We also include the negated seed words (i.e. words with *not_* prefix) to the opposite sentiment seed list. For example, we include the seed word *not_good* in the negative seed list. For domain-specific environments where the sentiment lexicon is different,

³https://www.yelp.com/dataset_challenge

⁴<http://movie.naver.com/>

⁵<https://stanfordnlp.github.io/CoreNLP/>

⁶http://www.shineware.co.kr/?page_id=835

Category	#Review	+/- Ratio	Ave. Len.	#Word
Nightlife	24,512	5.63	12.87	54,750
Restaurants	53,407	5.90	12.92	234,100
Shopping	15,615	5.45	12.38	21,076

(a) Yelp Restaurants and Shopping datasets

Category	#Review	+/- Ratio	Ave. Len.	#Word
Comedy	6,901	7.45	7.63	35,685
Crime	11,342	4.78	8.04	99,246
Fantasy	8,932	6.45	7.95	47,767

(b) Naver Movie datasets

TABLE II: Statistics of the datasets. **#Review**: the total number of reviews/documents within a category. **+/- Ratio**: the ratio between the number of positive and negative reviews. **Ave. Len.:** average length in words. **#Word**: The total number of unique words within a category.

existing work on domain-specific lexicon induction [15] can be adopted.

B. Parameter tuning

We set the Dirichlet priors to the following values: $\alpha = 0.1$, $\gamma = 1$, and $\delta = 0.1$, following [3], [16]. In the case of the β prior, we set it to different values depending on the current sentiment and the current pair of words. Moreover, following [3] for consistency, the following Dirichlet prior β is used accordingly. If both words are non-sentiment words, $\beta = 0.01$. If at least one word is a sentiment word, then if the sentiment matches the current sentiment, $\beta = 0.1$. Otherwise, β is set to zero. We set the number of clusters $C = 500$, number of topics $T = 15$, and number of sentiments $S = 2$. We limit the pairing of words into a context window that we set to 5. We then run the model with 1500 iterations.

C. Baselines

For our baseline models, we use the following four widely used aspect-sentiment models:

- **Joint Sentiment Topic (JST) Model** [2]: The most basic aspect sentiment model which extends the LDA topic model by adding a sentiment latent variable.
- **Aspect and Sentiment Unification Model (ASUM)** [3]: The most well-known aspect sentiment model which extends the JST model by adding a sentence layer in order for one sentence to capture a single aspect and sentiment. However, since the number of sentences in a short review is usually limited to one, ASUM cannot be directly used in the experiments. We instead use short phrases of size 5 to represent the sentences of ASUM.
- **Joint Aspect-based Sentiment Topic (JAST) Model** [4]: A fine-grained aspect-based sentiment topic model that separates the distribution of general sentiment terms, aspect terms and aspect-specific sentiment terms.

For fairness in comparison, all of the above use the same sentiment lexicons as used by MicroASM for both English and Korean datasets. Also, the same hyperparameter α , γ , β are used for the same purpose.

English	+	good, nice, excellent, positive, fortunate, correct, superior , amazing, attractive, awesome, best, comfortable, enjoy, fantastic, favorite, fun, glad, great, happy, impressive, love, perfect, recommend, satisfied, thank, worth
	-	bad, nasty, poor, negative, unfortunate, wrong, inferior , annoy, complain, disappointed, hate, junk, mess, dislike, unworthy, problem, regret, sorry, terrible, trouble, unacceptable, upset, waste, worst, worthless
Korean	+	좋다 (good), 우수하다 (excellent), 긍정적인 (positive), 운, 행운 (fortunate), 옳, 맞 (correct), 훌륭하다 (superior), 매력 (attractive), 대단하다 (awesome), 베스트, 짱, 최고 (best), 편안하다 (comfortable), 즐기다, 즐겁다, 재미, 재미있다 (enjoy), 기쁘다 (glad), 멋지다 (great), 행복, 행복하다 (happy), 인상적 (impressive), 사랑, 사랑하다 (love), 완벽하다 (perfect), 추천, 추천하다 (recommend), 만족, 만족하다, 만족스럽다 (satisfied), 감사하다, 고맙다 (thank), 가치, 보람 (worth)
	-	나쁘다 (bad), 더럽다 (nasty), 불쌍, 불쌍하다, 초라하다 (poor), 부정적 (negative), 불행, 불행하다 (unfortunate), 잘못되다, 잘못, 틀리다 (wrong), 괴롭다, 괴로움, 괴롭히다 (annoy), 불평하다, 불평 (complain), 실망, 실망하다, 실망스럽다 (disappointed), 싫다, 싫어하다, 미다 (hate), 쓰레기 (junk), 똥, 혼란 (mess), 싫증 (dislike), 문제 (problem), 후회, 후회하다 (regret), 미안하다, 죄송, 죄송하다 (sorry), 불편 (trouble), 당황, 당황하다, 당황스럽다 (flustered), 낭비, 낭비되다 (waste), 최악 (worst), 가치없다 (worthless)

TABLE III: The original and the Korean-translated Paradigm+ sentiment seed lists. The Korean words are accompanied with their corresponding English translations. Note that some of the English words are not translated into Korean because they are not directly translatable. The bold-faced words are the sentiment oriental paradigm words from the Paradigm list.

IV. EXPERIMENTS

A. Aspect term extraction

Each word distribution ϕ constitutes to a specific aspect and a sentiment. These distributions should contain aspect terms related to the aspect of ϕ , and aspect-specific sentiment terms related to the sentiment of ϕ . We perform evaluations to check whether these terms match with their corresponding aspect and sentiment. We use two kinds of metrics: distribution sentiment accuracy and aspect category assignment metrics.

a) *Distribution sentiment accuracy*: In order to check whether the aspect-specific sentiment terms correspond to the ϕ distribution’s sentiment, we evaluate using the accuracy of the classification of ϕ ’s sentiment based on human judgement. Given the top five terms of a distribution, we let two annotators classify the distribution with either positive, negative, or undecidable, where annotators are asked to assign the class *undecidable* to distributions unclassifiable as positive or negative. We then compare the human-annotated sentiments to the sentiment of the distribution, and calculate the accuracy. Since JAST produces three different ϕ distributions, we use the aspect-specific sentiment term ϕ distribution for comparison with other models.

The results are reported in Table IV. On all datasets, MicroASM outperforms all the other aspect sentiment models. Moreover, JST performs the worst among all the aspect sentiment models on all datasets. Interestingly, the overall results of all the models on the Naver datasets are comparatively lower than the results on the Yelp datasets. Although we can optimize for better accuracy by providing better hand-crafted seed lists and better performing tokenizers instead of machine-translated seed lists and off-the-shelf Korean tokenizers, the goal of this evaluation is to show how straightforward it is to extend to other domains and languages: MicroASM still performs the best despite problems on the Korean dataset.

b) *Aspect category assignment metrics*: We also verify whether the aspect terms are related to the aspect of the given ϕ distribution. We do this by measuring how diverse, specific, and agreeable the aspect categories judged by human

Dataset	JST	ASUM	JAST	MicroASM
Nightlife	66.7	68.3	80.0	88.3
Restaurant	65.0	66.7	68.3	83.3
Shopping	61.7	73.3	65.0	86.7
Average	64.2	69.4	71.1	86.1

(a) Yelp Restaurants and Shopping datasets

Dataset	JST	ASUM	JAST	MicroASM
Comedy	38.3	40.0	51.7	66.7
Crime	36.7	45.0	71.7	76.7
Fantasy	40.0	36.7	60.0	73.3
Average	38.3	40.6	61.1	72.2

(b) Naver Movie datasets

TABLE IV: Distribution sentiment accuracy results

annotators are. Given the top 20 terms of a distribution and a set of aspect categories which consists of specific and general aspects, we let two annotators classify the distribution with a given set of aspect categories. Only the Restaurant dataset is used because it is the only dataset with a publicly available hand-annotated set of aspect categories [17]. We use the aspect term ϕ distribution of JAST for comparison in this experiment. We then compute the following metrics: diversity, specificity, and agreeability.

Diversity measures how different the annotated labels are. The measure gives a higher value if the number of distinct annotated labels is also high, considering the number of word distributions annotated as a specific label. We use the Shannon diversity index to measure the diversity. The measure is given as follows:

$$Diversity = - \sum_{a=1}^A \frac{C_a}{N} \log \left(\frac{C_a}{N} \right) \quad (8)$$

where A is the list of aspect labels, C_a is the number of word distributions annotated as label a , and $N = T*S*2 = 60$ is the total number of word distributions.

Specificity measures how particular the annotated labels are. The measure gives a higher value if the number of distinct annotated labels are high, not considering the number of word

Metric	JST	ASUM	JAST	MicroASM
Diversity	0.584	0.541	0.436	0.702
Specificity	0.583	0.650	0.450	0.767
Agreeability	0.618	0.669	0.158	0.826
Average	0.595	0.620	0.348	0.765

TABLE V: Aspect category assignment metrics on Yelp Restaurant dataset

distributions annotated as a specific label. We use a simple metric for the specificity measure as follows:

$$Specificity = \frac{N - N(\text{general}, \text{other}, \text{none})}{N} \quad (9)$$

where $N(\text{general}, \text{other}, \text{none})$ is the number of annotations using the labels "general", "other", or "none".

Agreeability measures the agreement between annotators with regards to their annotated labels. The measure gives a higher value if annotations of annotators are the same. We use Cohen’s kappa coefficient for multiple categories [18], defined as follows:

$$Agreeability = \frac{\bar{P} - P_e}{1 - P_e} \frac{1 - \bar{P}}{N * (1 - P_e)} \quad (10)$$

where \bar{P} is the relative observed agreement among annotators and P_e is the hypothetical probability of chance agreement.

The results are reported in Table V. Overall, MicroASM significantly outperforms all the other models. The results also show that JAST performs the worst among the models. This is especially very apparent on the agreeability metric, which means that the ϕ distributions extracted by JAST is hard to interpret. This contradicts the results presented by the original paper [4]. Moreover, although JAST is generally better than JST and ASUM on other tasks such as distribution sentiment accuracy as presented above, and sentiment classification as presented in Section IV-B, it performs the worst in this experiment. This may be because the model needs a larger sentiment seed list to perform well⁷, while other models only need a very small subset of sentiment words to work well. One may consider applying larger seed lists, which can be done with insignificant increase in time and space complexity, but with the following complications. First, a good seed list must only have aspect-independent sentiments such as *good* and *bad*. When starting with a larger list, it is inevitable that aspect-specific sentiments (e.g. *fast* is positive in *fast performance* but negative in *fast battery consumption*) will be included in the list, with negative effects in the process. Second, a larger seed list may not be available in other low-resource languages such as Korean. Hence in this experiment, a smaller set of sentiment seeds is desirable for aspect sentiment models.

B. Sentiment classification

In this section, we evaluate our models in terms of sentiment classification. To determine the sentiment s of a review d , we

⁷In the JAST paper [4], they used a larger opinion lexicon [19].

Dataset	JST	ASUM	JAST	MicroASM
Nightlife	70.6	73.4	76.3	81.9
Restaurant	69.1	69.3	72.4	81.3
Shopping	68.4	67.9	70.0	79.1
Average	69.4	70.2	72.9	80.8

(a) Yelp Restaurants and Shopping datasets

Dataset	JST	ASUM	JAST	MicroASM
Comedy	57.6	71.0	80.2	87.2
Crime	57.2	70.7	70.3	77.0
Fantasy	51.5	73.5	74.9	78.8
Average	55.4	71.7	75.1	81.0

(b) Naver Movie datasets

TABLE VI: Sentiment classification accuracy results

can use Equation 5 to solve for $p(s|d)$. This will return a sentiment distribution consisting of two probabilities, one for positive sentiment and another for negative sentiment. We classify a review as positive if the positive sentiment probability is greater than the negative sentiment probability, and classify it as negative otherwise. We compare the performance of our models to JST, ASUM, and JAST.

The classification results are presented in Table VI in terms of accuracy. On all datasets, MicroASM performs the best. JST performs the worst on all datasets except for the Shopping dataset, where ASUM performs worse. One interesting finding is that ASUM, JAST, and MicroASM perform similarly on both English and Korean datasets, however JST performs worse on the Naver datasets. This may mean that on real world micro review datasets, JST performs worse than usual when it is used to normal datasets. Note that the main difference between the Yelp and the Naver datasets is that Naver reviews are forced to be written within a 140-character limit, while Yelp does not impose such constraint. This means that the Yelp dataset is synthetic and the Yelp short reviews are less natural than that of the Naver short reviews. Thus, the observations discussed above are more evident in Naver reviews than in Yelp reviews. On such datasets, MicroASM still performs better than other models, thus is a more appropriate aspect sentiment model for micro reviews.

V. RELATED WORK

Latent Dirichlet allocation (LDA) [1] has been extended to answer problems in ABSA. One such extension is the Joint Sentiment Topic (JST) model [2] by adding a new sentiment latent variable, aside from the topic latent variable. An extension to JST is Aspect and Sentiment Unification Model (ASUM) [3]. ASUM extends JST simply by adding a sentence layer, based on the assumption that grouping words into a sentence during inference provide much better aspect discovery. JST has also been extended by separately generating sentiment and aspect terms [20]. Recently, ASUM has been extended to further improve its performance by modeling the transition between two sentences using Markov chains [21] and by separating the variables of general and specific aspect and sentiment words [4], among others.

LDA has also been extended to accommodate short texts. Past works involve aggregating tweets through Twitter meta-data information such as user information [8] and through calculated scores based on Twitter trends such as burst score [22]. A more domain-independent approach is used in Biterm Topic Model (BTM) [9]. BTM removes the document layer, and instead used biterns, unordered pairs of co-occurring words, to learn topics from the documents. Another way to approach the problem is to use pseudodocuments. This approach is used in Pseudodocument Topic Model (PTM) [16]. Using pseudodocuments, short texts are implicitly aggregated to avoid the data sparsity problem.

There are several aspect sentiment models for datasets with additional metadata information such as Twitter. One such model is Twitter Opinion Topic Model (TOTM) [23], where they incorporate hashtags to extract product opinions from Twitter data. Microblog Sentiment Topic Model (MSTM) specially accounts for tokens such as hashtags to model both sentiment and topic word distribution [11]. A more problem-specific model used timeslices as additional variables to detect sentiment-aware topics from social media [10].

VI. CONCLUSION

In this paper, we proposed Micro Aspect Sentiment Model (MicroASM), an aspect sentiment model specifically for short reviews. MicroASM assumes that sentiment-aspect word pairs are the building blocks of information in short reviews and that reviews can be grouped into one information sharing cluster. We showed that our model performed better than previous generative models on two kinds of Aspect-based Sentiment Analysis (ABSA) tasks: aspect term extraction and sentiment classification. We confirmed that MicroASM extracts more meaningful aspect-specific sentiment terms and aspect terms. We showed two further applications that can be done using MicroASM: adjective-noun pair-based review summarization and preference-based review clustering. MicroASM is capable of discovering common and rare adjective-noun pairs that can be used for summaries. Moreover, we also showed that the model is able to automatically cluster the documents based on their preferences. Finally, we reported the results of sub-models, PairASM and ClusterASM created based on the individual observations, on the three ABSA tasks and showed that both observations are necessary to perform well on both aspect-level tasks such as aspect term extraction and document-level tasks such as sentiment classification.

ACKNOWLEDGMENTS

This work was supported by Samsung Research Funding Center of Samsung Electronics under Project Number SRFC-IT1701-01.

REFERENCES

- [1] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.
- [2] C. Lin and Y. He, "Joint sentiment/topic model for sentiment analysis," in *Proceedings of the 18th ACM conference on Information and knowledge management*. ACM, 2009, pp. 375–384.
- [3] Y. Jo and A. H. Oh, "Aspect and sentiment unification model for online review analysis," in *Proceedings of the fourth ACM international conference on Web search and data mining*. ACM, 2011, pp. 815–824.
- [4] S. Wang, Z. Chen, and B. Liu, "Mining aspect-specific opinion using a holistic lifelong topic model," in *Proceedings of the 25th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2016, pp. 167–176.
- [5] W. Wang, S. J. Pan, D. Dahlmeier, and X. Xiao, "Recursive neural conditional random fields for aspect-based sentiment analysis," *arXiv preprint arXiv:1603.06679*, 2016.
- [6] Y. Wang, M. Huang, L. Zhao, and X. Zhu, "Attention-based lstm for aspect-level sentiment classification," in *EMNLP*, 2016.
- [7] S. Ruder, P. Ghaffari, and J. G. Breslin, "A hierarchical model of reviews for aspect-based sentiment analysis," *arXiv preprint arXiv:1609.02745*, 2016.
- [8] L. Hong and B. D. Davison, "Empirical study of topic modeling in twitter," in *Proceedings of the first workshop on social media analytics*. ACM, 2010, pp. 80–88.
- [9] X. Yan, J. Guo, Y. Lan, and X. Cheng, "A biterm topic model for short texts," in *Proceedings of the 22nd international conference on World Wide Web*. ACM, 2013, pp. 1445–1456.
- [10] G. Kaminka *et al.*, "A joint model for sentiment-aware topic detection on social media," in *ECAI 2016: 22nd European Conference on Artificial Intelligence, 29 August-2 September 2016, The Hague, The Netherlands-Including Prestigious Applications of Artificial Intelligence (PAIS 2016)*, vol. 285. IOS Press, 2016, p. 338.
- [11] A. Ahuja, W. Wei, and K. M. Carley, "Microblog sentiment topic model," in *Data Mining Workshops (ICDMW), 2016 IEEE 16th International Conference on*. IEEE, 2016, pp. 1031–1038.
- [12] D. Alvarez-Melis and M. Saveski, "Topic modeling in twitter: Aggregating tweets by conversations," in *ICWSM*, 2016, pp. 519–522.
- [13] T. L. Griffiths and M. Steyvers, "Finding scientific topics," *Proceedings of the National academy of Sciences*, vol. 101, no. suppl 1, pp. 5228–5235, 2004.
- [14] P. D. Turney and M. L. Littman, "Measuring praise and criticism: Inference of semantic orientation from association," *ACM Transactions on Information Systems (TOIS)*, vol. 21, no. 4, pp. 315–346, 2003.
- [15] W. L. Hamilton, K. Clark, J. Leskovec, and D. Jurafsky, "Inducing domain-specific sentiment lexicons from unlabeled corpora," *arXiv preprint arXiv:1606.02820*, 2016.
- [16] Y. Zuo, J. Wu, H. Zhang, H. Lin, F. Wang, K. Xu, and H. Xiong, "Topic modeling of short texts: A pseudo-document view," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 2105–2114.
- [17] M. Pontiki, D. Galanis, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, A.-S. Mohammad, M. Al-Ayyoub, Y. Zhao, B. Qin, O. De Clercq *et al.*, "Semeval-2016 task 5: Aspect based sentiment analysis," *Proceedings of SemEval*, pp. 19–30, 2016.
- [18] H. C. Kraemer, "Extension of the kappa coefficient," *Biometrics*, pp. 207–216, 1980.
- [19] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2004, pp. 168–177.
- [20] H. Wang, "Sentiment-aligned topic models for product aspect rating prediction," Ph.D. dissertation, Applied Sciences: School of Computing Science, 2015.
- [21] M. M. Rahman and H. Wang, "Hidden topic sentiment model," in *Proceedings of the 25th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2016, pp. 155–165.
- [22] R. Mehrotra, S. Sanner, W. Buntine, and L. Xie, "Improving lda topic models for microblogs via tweet pooling and automatic labeling," in *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2013, pp. 889–892.
- [23] K. W. Lim and W. Buntine, "Twitter opinion topic model: Extracting product opinions from tweets by leveraging hashtags and sentiment lexicon," in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*. ACM, 2014, pp. 1319–1328.