ExtRA: A Framework for Extracting Prominent Review Aspects from Customer Feedback

Zhiyi Luo [1], *Bill Y. Lin* [2], *Frank F. Xu* [3], *Shi Feng* [4], *Hanyuan Shi* [5], *Kenny Q. Zhu* [6]

Department of Computer Science & Engineering
Shanghai Jiao Tong University, Shanghai, China
{jessherlock[1],yuchenlin[2],frankxu[3]}@sjtu.edu.cn,
{sjtufs[4],shihanyuan1995[5]}@gmail.com, kzhu@cs.sjtu.edu.cn[6]

***Abstract*—Many existing systems for analyzing and summarizing customer reviews about products or service are based on a number of prominent review aspects. Conventionally, the *prominent review aspects* of a product type are determined manually, but this costly approach cannot scale to large and cross-domain e-commerce platforms (e.g., Amazon.com and Taobao.com) or customer review sites (e.g., Yelp.com), where there are a large amount of product types and new products emerge almost everyday. In this paper, we propose a novel framework for extracting the most prominent aspects of a given product type from textual reviews. The proposed framework, ExtRA, extracts target prominent aspect terms and phrases in an automatic and unsupervised way. The ExtRA framework is general-purpose and can be applied to various types of product and service. We also demonstrate a downstream application of ExtRA, which benefits users to understand the customer reviews more efficiently and compare products more directly based on the prominent aspects extracted by ExtRA. Extensive experiments show that ExtRA is effective and achieves the state-of-the-art performance on a dataset consisting of different product types.**

I. Introduction

Online user review is an essential part of e-commerce. Popular e-commerce websites feature enormous amount of text reviews, especially for popular products and services. To improve the user experience and expedite the shopping process, many websites provide qualitative and quantitative analysis and summary of user reviews, which is typically organized by different *prominent review aspects*. For instance, Fig. 1 shows a short review passage from a customer on TripAdvisor.com, and the customer is also asked to give scores on several specific aspects of the hotel, such as *location* and *cleanness*.

Review analysis and summarization with some specific review aspects is defined as *aspect-based review summarization* [1], which is a more concise and effective way of representing customer feedback and thus is desired by both shop managers and potential customers. Reading enormous user reviews and summarizing the advantages and disadvantages from them can be extremely time consuming. With aspect-based reviews summary, potential customers can assess a product from various essential aspects very efficiently and directly. Also, aspect-based review summary offers an effective way to group products by their prominent aspects and hence enables quick comparison. It is evident that extracting such aspect terms of great prominence is the most fundamental and important step for building such aspect-based review analysis systems.

Fig. 1: An example user review about a hotel on TripAdvisor. The grades are organized by different prominent review aspects: *value*, *rooms*, etc.

Existing approaches for producing such prominent aspect terms has been largely manual work [2], [3]. This is feasible for web services that only sell (or review) a small number of product types of the same domain. For example, TripAdvisor.com only features travel-related products, and Cars.com only reviews automobiles, so that human annotators can provide appropriate aspect terms for customers based on their domain knowledge. While it is true that the human knowledge is useful in characterizing a product type, such manual approach does not scale well for general-purpose e-commerce platforms, such as Amazon, ebay, Taobao and Yelp, which feature too many product types, not to mention that new product and service types are emerging almost everyday. In these cases, manually selecting and pre-defining aspect terms for each type is too costly and even impractical.

Moreover, the key aspects of a product type may also change over time. For example, in the past, people care more about the screen size and signal intensity when reviewing cell phones. These aspects are not so much of an issue in present days. People instead focus on battery life and processing speed, etc.

looks good (462) beautiful screen (398)
nice functions (356) high resolution (872)
good camera (218) good value (628)

fast system (196) long battery-life (193)
good design (236) high call-quality (163)
nice functions (181) good value (282)

Fig. 2: Automatic review summarization for two mobile phones on an e-commerce website

Therefore, there is a growing need to automatically extract prominent aspects from user reviews.

A related but different task is *aspect-based opinion mining* [4], [5]. Here techniques have been developed to automatically mine product-specific aspect phrases such as those shown in Fig. 2. These are terms or phrases most frequently mentioned about a phone model along with the number of times they are mentioned in the user review text. Their goal is to get the *fine-grained*, possibly overlapping aspects about a particular product, which will then be used as the target of sentiment analysis from users reviews. We argue that such terms are not suitable for aspect-based review summarization because:

1) aspect phrases for different products of the same type are often different, making it difficult to compare them directly;
2) emotional terms in the aspect phrases can hinder potential customers to assess products efficiently from different prominent angels.

The goal of this paper is to develop an unsupervised framework for automatically extracting the most prominent review aspects for given product types from user review text. Developing such an unsupervised framework is challenging for the following reasons:

- We expect that the extracted prominent aspects are both important and representative for a given product, yet have little semantic overlap with each other.
- The expression of opinions in user reviews can be very versatile; aspect terms can be expressed either explicitly by direct mention or just implicitly through the personal experiences of customers.
- Very often, opinions including multiple aspects are covered within the same short piece of comment, and thus the topics transit from sentence to sentence much faster than other kinds of documents; this makes information extraction from user reviews significantly challenging.

Most previously proposed unsupervised approaches for the prominent aspect extraction task are different variations of topic modeling techniques [6]–[8]. The main problem of such approaches is that they typically use only word frequency and co-occurrence information, and thus degrade when extracting aspects from sentences that appear different on the surface but are actually discuss similar aspects.

Given a certain product type, our framework, ExtRA, first clusters all the review sentences into several *sentence cluster*s. Then, within each sentence cluster, we perform *topic modeling* to obtain potential aspect topics. We further cluster all the potential topics across sentence clusters to produce the final refined *topic cluster*s, based on our designed vectorial representation for word distributions (*AspVec*). Then, ExtRA can extract the most prominent aspect terms and phrases based on similarity computation.

Our main contributions in this paper are as follows:

1) We propose a novel unsupervised framework for extracting prominent aspects from customer review corpora.
2) Extensive experiments show that our framework is effective and outperforms the state-of-the-art methods by a substantial margin.
3) We release an open-source implementation of the framework and an evaluation dataset for future work in this research area.
4) We demonstrate a downstream application of our ExtRA framework, which can benefit existing aspect-based review analysis systems.

The rest of this paper are organized as follows. In Sec. II, we introduce the ExtRA framework in detail. In Sec. III we evaluate the proposed framework on a dataset consisting multiple domains. A downstream application of ExtRA is demonstrated in Sec. IV. Finally, in Sec. V, we discuss and compare our work with previous research related to aspect-based review analysis.

II. ExtRA Framework

In this section, we first state the *review aspect extraction problem*, then present the overall workflow of ExtRA, and finally talk about the framework stage by stage.

A. Problem Statement

The review aspect extraction problem is how to extract a certain number of words (or phrases) from customer reviews about a given type of product, each of which represents a prominent and distinct review aspect.

For instance, if the given product type is *hotel*, we expect a successful extraction framework to extract the aspect terms as follows: *room, location, staff, breakfast, pool, etc.*

B. Overall Workflow

The overall workflow of ExtRA framework is shown in Fig. 3, which consists of 5 stages:

1) *Sentence Representations and Clustering.* We encode review sentences into distributed representations, and then cluster them into several coarse-grained topics in an unsupervised way.
2) *Topic Modeling for Each Sentence Cluster.* In order to obtain potential prominent aspect topics, we perform the topic modeling in each sentence cluster. We do this in each cluster instead of all sentences because topic

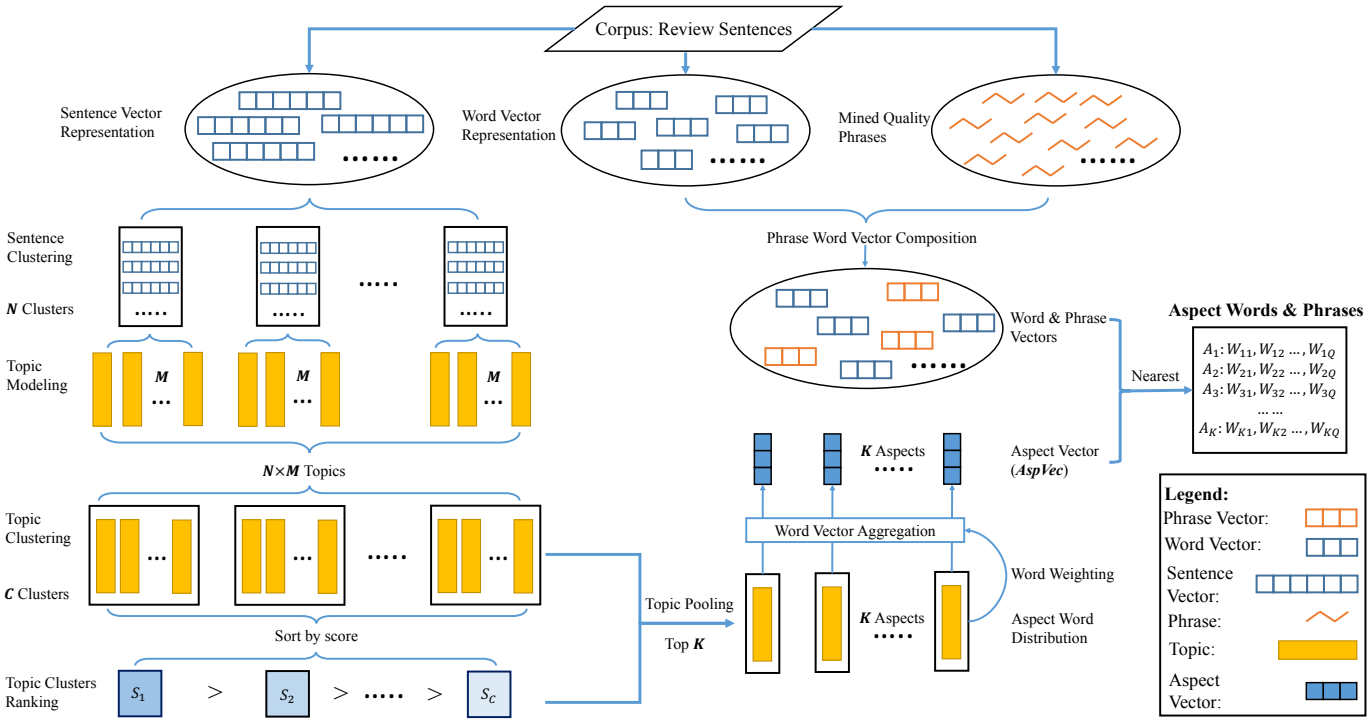

Fig. 3: Our overall framework.

modeling tends to be more accurate on a smaller and relatively cleaner corpus. Also, we treat each sentence cluster equally, so that topic modeling process is less affected by uneven distribution of aspects in user reviews.

3) *Topic Clustering.* To improve the consistency of the topics in each sentence cluster, we further cluster the topics from all sentence clusters to obtain the final purified topic clusters. This step isolates away noisy topics by redistributing the potential topics in the process of clustering.
4) *Topic Cluster Ranking.* We select the most prominent and distinct topic clusters by a ranking mechanism.
5) *Aspect Extraction from Ranked Topic Clusters.* In the final stage, we extract aspect terms from previously ranked topic clusters. We first propose a word-ranking based method, which can score each candidate term for its prominence. For extracting the aspect phrases, we further propose an encoding method. It represents each selected topic cluster with a vector, so that we can extract the most prominent aspect terms and phrases based on nearest neighbors searching in a shared vector space.

Next we will discuss each stage in more details.

C. Stage 1: Sentence Representations and Clustering

Consider the following hotel review example:

> *Pool is small and only 4 ft but refreshing. Hot tub also there. Staff were super friendly each day. Room was nothing special but clean and comfy. Lots of restaurants and bars nearby. Breakfast was great and despite being a busy weekend there was always a big selection available."*

From this example, we can conclude that topics (underlined) in reviews can shift very quickly, and adjacent sentences may refer to completely different aspects about a product. Such fine-grained semantic shifts in user reviews make it more suitable to consider the sentence as our base unit instead of a whole review document.

Motivated by this observation, we divide all review documents into sentences and then perform sentence clustering. Instead of using simplistic methods like bag-of-word representation, we leverage distributed representation which encodes words and sentences as low-dimensional real-valued vectors. Existing sentence embedding methods include averaging word vectors in a sentence, ParagraphVector (PV) [9], LSTM-based methods [10], etc. After obtaining sentence representations, we cluster them into N clusters by k-means algorithm [11]. As a result, we obtain N clusters of sentences each carrying some coarse-grained semantics.

D. Stage 2: Topic Modeling for Each Sentence Cluster

Although we performed sentence clustering based on semantic sentence representations, there still exits some sentences that carry multiple topics or contain noise. For example, the following two sentences are taken from two TripAdvisor reviews:

- Sentence 1: *"The room was clean, the staff were friendly, and I would say the price is very reasonable given the proximity to business and leisure destinations around downtown."*
- Sentence 2: *"There is a restaurant just 5 min walk away with nice Italian food, pizza was great."*

TABLE I: Topics extracted from three sentence clusters of hotel review.

Sentence cluster 1	t1: room bed bedroom size floor t2: bedroom room wall size decor t3: room bathroom shower water towel t4: room suite size view floor t5: room shower area kitchen bed
Sentence cluster 2	t1: station minute tube location bus t2: location price night place rate t3: location square station street subway t4: distance bus subway downtown shopping t5: **restaurant city food buffet place**
Sentence cluster 3	t1: price rate service money star t2: **location city star time rate** t3: price service night money city t4: price location place night city t5: **location service food price restaurant**

Sentence 1 mentions multiple aspects such as *room*, *staff* and *price*; sentence 2 is mainly about the *location* aspect of the hotel but it contains some irrelevant information about the food outside the hotel.

To address such challenges, we apply topic modeling within each sentence cluster and obtain M topics for each cluster. This gives us in total $N \times M$ topics. Each topic here can be considered as a word distribution.

We choose to use Biterm Topic Model (BTM) [12] for the topic modeling. It is a word co-occurrence based topic model that learns topics by modeling word-word co-occurrences patterns (i.e., biterms). A bi-term consists of two words co-occurring in the same context, for example, in the same short text window. Unlike word-document co-occurrence based topic modeling methods (such as LDA [13] and PLSA [14], [15]), BTM models the bi-term occurrences in a corpus, which alleviates the data sparsity problem of short documents. Therefore, BTM is much more suitable for modeling topics from our review sentence clusters.

Table I shows an example of topics modeled from hotel reviews ($N = 3$ and $M = 5$). In this example, there are five topics (t_1-t_5) extracted from each sentence cluster. We find that the core aspects for these three clusters should be *room*, *location* and *price* respectively.

However, there are still some noise in such sentence clusters: boldfaced topics are obviously irrelevant to the core aspect of their clusters. Especially, "t5" in "Sentence cluster 3" consists of multiple different aspects.

E. Stage 3: Topic Clustering

To address the above problem, we propose to take a step forward and cluster the topics across the sentence clusters. This way, we can obtained C topic clusters from the above $N \times M$ topics. Note that each topic here is represented as a word distribution. Also, if we would like to obtain the K most prominent aspects from the user reviews, then we purposely set C to be larger than K for further refinement.

Simple clustering on the word distributions is infeasible for two reasons: 1) the vocabulary size is too large to cluster the topics; 2) the semantic similarities are not utilized to cluster topics. We propose a novel approach to represent each topic: we construct a vector for each topic by summing up the word embeddings of the k most probably words in this topic. Note that the summation is weighted with $P(w|t)$ as. Such *topic vector*s represent the topic centers by integrating the topic word semantics. With k-means clustering algorithm, we aggregate the topics based on their semantic similarities. Table II shows some example topic clusters extracted from hotel reviews, where each row is a topic cluster. We sort the words with the sum of their probabilities of all topics within each topic cluster and use the top ones to represent each topic cluster. We can see that each topic cluster are now more concentrated and most words are related to each other within a single review aspect. For example, the words in the first row are all about breakfast and food, while the forth row is about the location and the transportation.

TABLE II: Topic clusters extracted from hotel reviews. Each row shows the candidate words of a topic cluster, sorted by their weights.

breakfast, meal, food, tasty, dinner, morning, coffee, tea
room, night, time, bed, day, bathroom, staff, area, place
staff, desk, service, friendly, reception, concierge, helpful
close, city, location, place, central, station, bus, street
bed, shower, spacious, room, size, bathroom, bedroom, floor
price, room, check, night, money, city, location, star, service
location, price, room, night, place, rate, money, time, city

F. Stage 4: Topic Cluster Ranking

In the previous stage, we form C topic clusters. We hereby propose a scoring method to measure the distinctiveness of each topic cluster, which captures the intuition that a more distinctive topic cluster are supposed to have smaller overlaps with other clusters.

We first introduce the notations as follows:

- we use T to denote a certain topic cluster, which is a set of topics (word distributions).
- $\mathcal{V}_T$ is the vocabulary of the core words in this topic cluster T, which is the union of the top k words with the highest probabilities in each topic within T:

$$\mathcal{V}_T = \bigcup_{t \in T} \{w_1^t, w_2^t, ..., w_k^t\} \tag{1}$$

  where w_i^t denotes the i-th word sorted by $P(w|t)$ in descending order, and $P(w|t)$ is the probability in the word distribution of the topic t.
- we use $m(T, w)$ to denote the importance of the word w in the topic cluster T, which is the average probability across all the topics in T:

$$m(T, w) = \frac{\sum_{t \in T} P(w|t)}{|T|} \tag{2}$$

TABLE III: Aspect clusters ranked by distinctiveness score. Potential aspect words are boldfaced.

staff, desk, **service**, friendly, reception, concierge, helpful
breakfast, meal, **food**, tasty, dinner, morning, coffee, tea
price, room, check, night, money, city, location, star, service
bed, shower, spacious, **room**, size, bathroom, bedroom, floor
close, city, **location**, place, central, station, bus, street
room, night, time, bed, day, bathroom, staff, area, place
location, price, room, night, place, rate, money, time, city

- we use $g(T, w)$ to denote the normalized $m(T, w)$ over the $\mathcal{V}_T$:

$$g(T, w) = \frac{m(T, w)}{\sum\limits_{w' \in \mathcal{V}_T} m(T, w')} \tag{3}$$

Our scoring method for measuring the distinctiveness of a topic cluster T is to compute the distinctiveness of each word and sum them up as follows:

$$S(T) = \sum_{w \in \mathcal{V}_T} \log \left(\frac{g(w, T)}{1 + \sum\limits_{T' \neq T} g(w, T')} \right) \tag{4}$$

In this scoring function, if a word w occurs more often in the topic cluster T but occurs less often in other topic clusters, then the term in the parenthesis is larger. Thus, accumulating such distinctiveness of each word in the $\mathcal{V}_T$, we can obtain the distinctiveness of the topic cluster T. Then, we only keep the top K topic clusters accordingly.

G. Stage 5: Aspect Extraction from Ranked Topic Clusters

We propose two approaches for extracting K most prominent aspect terms for such K ranked topic clusters.

1) Word Ranking: This is a ranking algorithm to score the words within each topic cluster,which produces a list of K aspect terms. Extracting the most prominent aspect terms from a set of topic clusters can be achieved by ranking each candidate term with their importance and distinctiveness.

For example, in Table III, the most representative words for each cluster are boldfaced. Each of such terms is a promising aspect term. However, not all of them are associated with high probability in their clusters. In order to automatically select the most prominent aspect words, we propose a novel approach to re-rank them with both their importances, $g(T, w)$, and their semantics.

Intuitively, the most prominent and representative words in each topic cluster are assumed to be the closest to centroid of the cluster. Therefore, we calculate the semantic similarities (cosine similarities of word embeddings) of each word with all other words in the cluster and use that to measure how central each word is. In order to prevent generating duplicate aspect terms from different topic clusters, we process the ranked topic clusters in a sequential order. When we calculate the score of a word w in the i-th topic cluster T_i, we also consider the scores of w in all the previous topic clusters, namely $\{T_1, ..., T_{i-1}\}$. We prevent the duplicate aspect words by subtracting the scores of other clusters from the score of cluster i. If the word w has already been assigned with a high score in previous topic clusters, then its score in the current topic cluster should decrease. Thus, we subtract the scores of w in previous clusters. We consider this as a *mechanism of decreasing the importance of words* over iterations. The score of word w in the topic cluster T_i is:

$$\text{score}(w, T_i) = g(w, T_i) \sum_{w' \in \mathcal{V}_{T_i}} \cos(\mathbf{w}, \mathbf{w}') - \sum_{k=1}^{i-1} \text{score}(w, T_k) \tag{5}$$

where $\mathbf{w}$ is the word embedding of the word w and $g(w, T_i)$ is defined in the previous stage. Such scores for each word consider both the distinctiveness across different topic clusters and the semantic similarities inside each topic cluster. The algorithm is illustrated in Algorithm 1.

Algorithm 1: Extracting aspect terms from topic clusters

Input: Topic clusters $TS = \{T_1, T_2, ..., T_C\}$, expected number of aspects K
Output: aspect term set A, including K aspect terms

$A \leftarrow \{\}$
for *each topic cluster* $T \in TS$ **do**
  $S(T) \leftarrow 0$
for *each topic cluster* $T \in \{T_1, T_2, ..., T_C\}$ **do**
  $Z \leftarrow 1$
  for $T' \in TS$ *and* $T' \neq T$ **do**
    for *each word* $w \in \mathcal{V}_{T'}$ **do**
      calculate $g(w, T')$*, the importance of* w *in* T'
      $Z \leftarrow Z + g(w, T')$
    for *each word* $w \in \mathcal{V}_T$ **do**
      $S(T) \leftarrow S(T) + log(\frac{1}{Z} * g(w, T'))$
$rankedTS = [T_1, T_2, ..., T_C]$, where $S(T_1) > S(T_2) > ... > S(T_C)$
remove last $C - K$ *topic clusters from* $rankedTS$
$rankedTS = [T_1, T_2, ..., T_K]$
for $1 \leq i \leq K$ **do**
  for *each word* $w \in \mathcal{V}_{T_i}$ **do**
    $\mathbf{w} \leftarrow$ embedding of word w
    for *each word* $w' \in \mathcal{V}_{T_i}$ *and* $w' \neq w$ **do**
      $\mathbf{w}' \leftarrow$ embedding of word w'
      $\text{score}(w, T_i) \leftarrow \text{score}(w, T_i) + g(w, T_i) * cos(\mathbf{w}, \mathbf{w}')$
    for $1 \leq k \leq i - 1$ **do**
      $\text{score}(w, T_i) \leftarrow \text{score}(w, T_i) - \text{score}(w, T_k)$
  $aspect \leftarrow \operatorname{argmax}_w \text{score}(w, T_i)$
  $A \leftarrow A \cup aspect$
return A

2) AspVec-based Extraction: We propose a method to encode words, phrases and topic clusters into the same vector

space, so that we can find the most prominent aspect terms and phrases based on similarity computation inside this same vector space. We name the vectorial representations of topic vectors as "*AspVec*".

For each topic cluster T, we sort the words in $\mathcal{V}_T$ by their $g(w,T)$ and then we extract the highest k ones as a new word set, namely $\overline{\mathcal{V}}_T$. We obtain a *AspVec* for the topic cluster T by summing up the word embeddings of the words in $\overline{\mathcal{V}}_T$ with their weights:

$$Z = \sum_{w \in \overline{\mathcal{V}}_T} \text{score(w,T)} \tag{6}$$

$$\text{AspVec}(T) = \frac{1}{Z} \sum_{w \in \overline{\mathcal{V}}_T} \text{score}(w,T)\mathbf{w} \tag{7}$$

In order to extract aspect phrases, we extend our vocabulary with high quality phrases extracted by AutoPhrase [16]. We encode phrases by averaging the embeddings of each words.

Thus, we encode topic clusters and candidate terms/phrases into a single vector space. It is natural to select the nearest term (word/phrase) to the *AspVec* for each topic cluster base on their cosine similarities. The order of extracting follows the ranked list of Stage 3 and we remove each selected term every step, so that the final K prominent aspect terms are unique.

III. Experiments

We compare the ExtRA framework with a number of baseline methods on extracting aspect terms from user reviews. We first introduce a newly built dataset, the parameter tuning of ExtRA, the baseline methods, and finally the comparisons as well as an investigation of the effectiveness of various component of ExtRA.

A. Dataset

We collect our customer review corpora of 6 kinds of product and service from three popular e-commerce websites, including Amazon, Tripadvisor and Yelp. The statistics of our corpora are shown in Table IV.

TABLE IV: Dataset summary.

Product type	Source	No. of Reviews
hotel	TripAdvisor	27,145
mobile phone	Amazon	3,716
mp3 player	Amazon	2,745
laptop	Amazon	5,471
cameras	Amazon	3,077
restaurant	Yelp	4,016

Existing published datasets [1], [17]–[19] collect fine-grained aspects from reviews, which are not suitable for evaluating the performance of extracted prominent aspect terms. Therefore, we build a new evaluation dataset particularly for this task. For each category, we ask 5 annotators who are proficient in English to give 5 aspect terms while reading the reviews corresponding to each product type. The manually created ground-truth prominent aspects for each product type are shown in Table V.

TABLE V: Ground-truth. Each row is provided by one annotator.

hotel	room price location service utility room service price food location sleep service room price location location price bedroom bath staff room price bath staff location
camera	image lens battery memory carry picture lens price battery mode image price battery design operation image lens battery focus storage image appearance lens portability battery
restaurant	location price food service cleanness food price location environment service price food quietness location staff food price service environment location food price location service environment
mobile phone	brand price quality battery screen price quality camera touch battery quality price design screen carry quality price OS battery service price quality screen battery color
mp3 player	price quality sound screen battery carry price design sound screen price quality carry earphone sound quality price battery sound carry price quality sound carry screen
laptop	price quality brand OS battery quality price battery memory CPU disk memory CPU screen keyboard price battery screen CPU performance quality price appearance battery keyword

TABLE VI: The effect of different M

M	5	8	**10**	12
Accuracy	17/25	16/25	17/25	16/25

B. Parameter Tuning

We empirically determine the parameters of our ExtRA N, M, C in sentence clustering, topic modeling and topic clustering stages respectively.

Number of Sentence Clusters (N) We expect that sentences in each sentence cluster are about a single review aspect. We use an empirical elbow method to find an optimal N by plotting the loss, which is the sum of euclidean distances from cluster center to each point within the cluster. Fig. 5 shows the changes of the loss over different number of sentence clusters. We conduct the experiments on the *hotel* reviews. We find that setting N around 10 is optimal to to achieve the lowest loss, and therefore we set N as 10 in the following experiments.

Number of Topics within Each Sentence Cluster (M) In the topic modeling stage, we need to determine the number of topics that we would like to model for each sentence cluster. As Table VI shows, the performance of our framework is insensitive to M. Therefore, we set M as 10 in our following experiments.

The Number of Topic Clusters (C) As mentioned in Sec. II-E, we obtain C topic clusters in the topic clustering

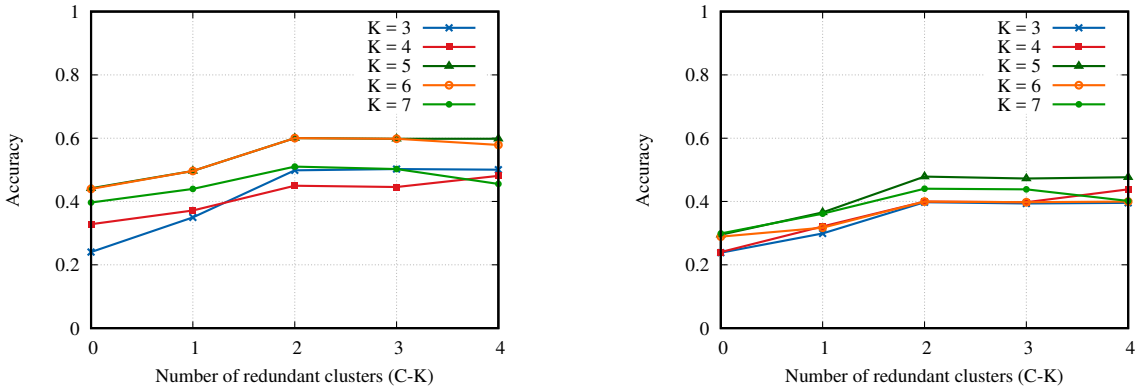

Fig. 4: The performance of different models when adjusting the number of redundant aspects ($C - K$). Left: hotel reviews. Right: mobile phone reviews.

stage, where C is larger than K for removing redundant and noisy topic cluster in the following steps. We demonstrate the effect of different C by adjusting the number of redundant topic clusters ($C - K$) with different selected K in Fig. 4. $C - K$ is in range $[0, 4]$, and K is in range $[3, 7]$. We can see that it is adequate to set $C - K$ as 2, and our framework would not benefit from removing more topic clusters.

1) Baseline Models and Our ExtRA: We introduce four baseline models for aspect extraction: LDA [13] and BTM [12] are two simple topic modeling-based methods; D-PLDA [20] is a representative for joint aspect-sentiment models; MG-LDA [21] is a representative for aspect extraction topic models. We also investigate two variations of our ExtRA model, i.e. ExtRA-LSTM and ExtRA-PV, to certify that the using PV is better than using LSTM-based sentence representation. Note that we use BTM as our topic modeling method in ExtRA.

a) LDA and BTM: We use LDA and BTM as two basic topic modeling based methods. Considering a given product type or service, we treat each review as a document and perform vanilla LDA and BTM on the reviews to extract K topics. Then, we select the words with highest probabilities in each topic as our extracted aspect terms.

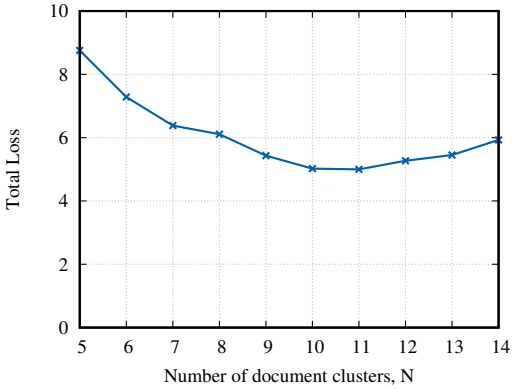

Fig. 5: Selecting N, the number of clusters for k-means. The optimal N is selected to be 10.

b) D-PLDA: D-PLDA [20], is a variant of LDA models, which is designed specifically for modeling topics from user reviews. D-PLDA only considers opinion-related terms and phrases, and nouns and phrases are controlled by two separate hidden parameters. The hidden parameters of adjectives are depended on the parameters of nouns. We use D-PLDA as a representative for such models joint extracting aspect and sentiment.

c) MG-LDA: To compare our model with a popular, well-performed model designed particularly for aspect extraction, we use MG-LDA [21] as a sophisticated baseline method. MG-LDA can also models topics at different granularities. For fair comparison among different models, the number of target aspects K is set as 5. The hyper-parameter of MG-LDA (global topics) is set to 30 with fine-tuning.

d) ExtRA models: The dimension of our sentence embeddings is set as 300. We refer our models which are fed with different sentence embeddings as ExtRA-PV and ExtRA-LSTM, respectively. Note that both of them utilize BTM as the topic modeling method in Stage 2. The parameter configurations in our experiment in Table VII are as follows: $K = 5$, $N = 10$, $M = 10$, and $C = 7$. Note that we train the sentence embeddings merely on review texts and do not use extra data.

2) Aspect Term Extraction Comparison: The labels provided by the annotators are aggregated together without removing duplicated words, so we have 25 words in total. This is to ensure that the information about the different importances of aspects is preserved. When evaluating the models, we compare the 5 aspect words generated by the models with those provided by the annotators. We calculate the portion of words among the 25 labels that are correctly generated by the model as the *accuracy* of the model.

In this evaluation, we compare the performance of our models with the baseline models mentioned above. The results are shown in Table VII. Our models outperform others in all categories. We can also see that ExtRA-PV indeed outperforms ExtRA-LSTM, which is consistent to the previous experiment results. Also, we find that BTM is better than LDA and

TABLE VII: Comparison of accuracies using different models for aspect extraction.

Types / Models	hotel	mobile phone	mp3 player	laptop	camera	restau-rant
LDA	0.36	0.40	0.28	0.32	0.36	0.24
BTM	0.40	0.40	0.32	0.36	0.36	0.32
D-PLDA	0.44	0.48	0.40	0.44	0.44	0.40
MG-LDA	0.56	0.60	0.48	0.60	0.60	0.52
ExtRA-LSTM	0.56	0.60	0.44	0.56	0.52	0.48
ExtRA-PV	0.68	0.72	0.52	0.64	0.64	0.60

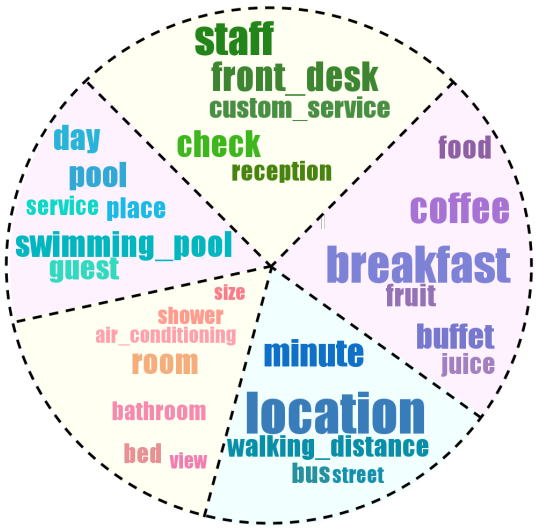

Fig. 6: Aspect Cloud: visualization of aspect words and phrases extracted from the hotel reviews.

our framework performs better than the existing state-of-the-art methods. Especially, since the accuracy of ExtRA-PV is higher than BTM, we claim that our sentence clustering indeed benefit the topic modeling process.

To qualitatively evaluate different models, we present that the extracted aspect terms (boldfaced) of each models and their candidates with highest scores in Table VIII. Our framework can be extended to extract multi-word aspects (aspect phrases) by integrating AspVec, which is called ExtRA-PV+Phrase. Fig. 6 is the visualization (using TopicCloud toolkit[1] [22]) for aspect clusters generated by ExtRA-PV+Phrase model on hotel review. The portion of area for clusters depends on their distinctiveness score. Also, the scores of the words in each cluster determine their font-sizes. With this visualization, the importance of each cluster and word is clearly presented.

3) Effect of the Number of Target Prominent Aspects (K): K reflects the different level of granularity of expected aspects. A larger K requires the framework to extract more fine-grained prominent aspects. We conduct the experiments using our framework with different K ranging from 3 to 7. From the performance shown in Fig. 7, we can conclude that our framework always outperforms other competitors given different K substantially.

[1]Open sourced toolkit is available at https://github.com/askerlee/topiccloud

TABLE VIII: Top aspect words (with phrases) for hotel reviews by different models

ExtRA-LSTM	**service**, front, desk, reception, concierge, check, gust **location**, station, minute, tube, station, bus, distance **time**, check, day, desk, charge, book, front, hour, night **food**, coffee, buffet, morning, tea, room, fruit, egg, juice **bed**, bathroom, size, floor, view, suite, king, book, decor
ExtRA-PV	**staff**, service, room, front, desk, check, concierge **food**, breakfast, bar, restaurant, coffee, morning, tea **price**, parking, night, place, rate, service, money, star **room**, bed, bathroom, size, suite, floor, view, bedroom **location**, minute, square, subway, street, block, distance
ExtRA-PV+Phrase	**staff**, front_desk, check, custom_service, reception, concierge **breakfast**, coffee, buffet, food, fruit, juice, tea **location**, minute, walking_distance, bus, street, block **room**, bed, bathroom, shower, air_conditioning, view, size **swimming_pool**, pool, guest, place, service, day
MG-LDA	**shower**, bathroom, room, floor, area, bedroom, desk, tea **time**, day, room, check, front, desk, night, service **food**, bar, service, breakfast, restaurant, staff, taxi **room**, bed, floor, place, air, night, bathroom, noise **price**, business, service, star, internet, location, staff
DP-LDA	**service**, front, desk, reception, concierge, check, guest **station**, minute, tube, location, bus, distance, street **check**, day, time, desk, charge, book, front, hour **coffee**, buffet, morning, tea, room, day, fruit, food **bed**, bathroom, size, floor, view, suite, king, book
LDA	**stay**, night, place, trip, time, weekend, night, hour **location**, square, street, place, restaurant, market, block **room**, bed, bathroom, size, tv, king, suite, pillow **staff**, service, desk, location, concierge, room, night **room**, floor, noise, view, night, water, door, bathroom

4) Effectiveness of the Word Ranking Algorithm: In the previous experiments, we always use the first approach (word ranking based) to extract the target aspect terms from the ranked topic clusters, other than the experiment with aspect phrases. We compare three different ranking setups in this experiment to show the effectiveness of our proposed ranking method.

- Without word ranking. After removing $C - K$ redundant topic clusters, we simply select the most important word w associating with highest $m(T, w)$ (mentioned in Sec. II-F) from each topic cluster T.
- Word ranking without the word importance degrading mechanism. We assign the word w within each topic cluster T_i with a score score(w, T_i) defined in Eq. (5). However, we do not decrease the importance of terms over iterations (described in Sec. II-G), which prevents our framework from extracting repetitive aspect words from different topic clusters.
- Word ranking with the word importance degrading mechanism. We rank words within each topic cluster and decrease the importance of the words that have been scored before as described in Eq. (5).

As shown in Table IX, we find that: 1) our proposed semantic similarity score for ranking words is indeed effective; 2) the word importance decreasing mechanism improves the prominence of the extracted aspect term.

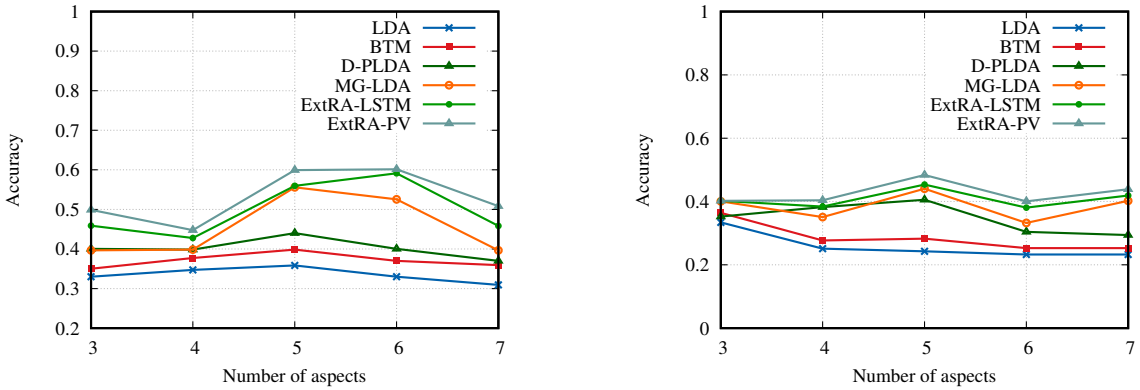

Fig. 7: The performance of different models when adjusting the number of expected aspects K. Left: hotel reviews. Right: mobile phone reviews.

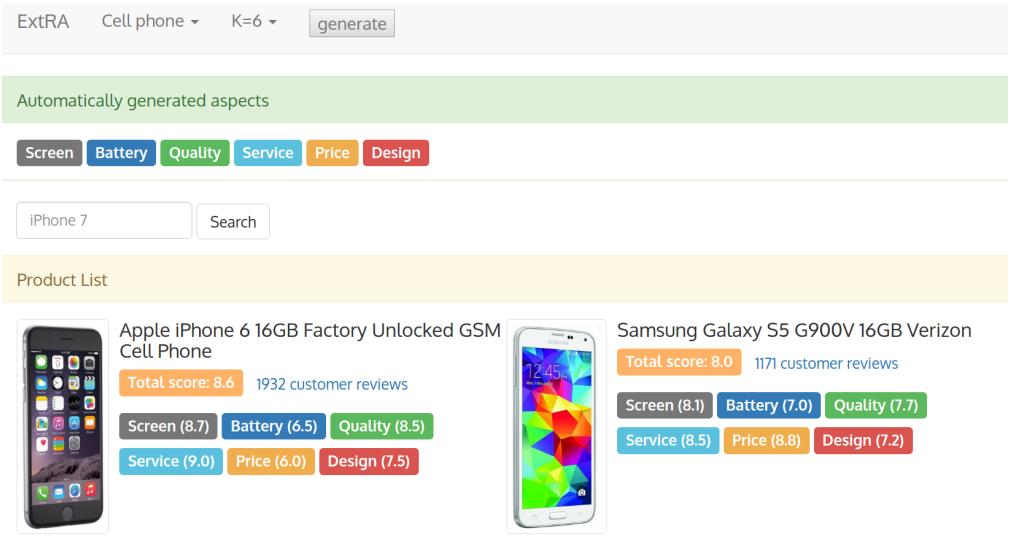

Fig. 8: Automatically generated 6 aspects for mobile phones

TABLE IX: The accuracy performance with different ranking setups.

Type \ Setup	No word ranking	Word ranking w/o. duplicate prevention	Word ranking w. duplicate prevention
hotel	0.48	0.60	**0.68**
mobile phone	0.52	0.64	**0.72**
mp3 player	0.36	0.44	**0.52**
laptop	0.44	0.52	**0.64**
camera	0.43	0.49	**0.58**
restaurant	0.48	0.56	**0.60**

IV. Application Demonstration

A most important downstream application of our extracted prominent review aspects is *aspect-based sentiment analysis on user reviews*. In this section, we demonstrate a real instance of such application, developed with the results from our framework ExtRA.

With the application, people can specify their expected number of prominent aspects K for a certain product type. Then, they can investigate the sentiment of existing customers towards any products under this type from the most prominent aspects extracted by ExtRA.

Fig. 8 shows a snapshot of our application when users are investigating the cell phone reviews based on $K = 6$ extracted prominent aspects. The topic clusters extracted from mobile phones review texts are shown in Table X. We propose a scoring method for review summarization in our demo system. As the running example shown in Fig. 8, each kind of mobile phone has rating scores summarized on each extracted prominent review aspect. The scores are computed by sentiment analysis of reviews containing the prominent aspects. We calculate the sentiment score of each aspect by analyze the sentiment of all sentences containing this aspect term. Using *Deeply Moving*[2], a well-developed deep learning model for sentiment analysis [23], we obtain a score for all

[2]https://nlp.stanford.edu/sentiment/code.html

TABLE X: The aspect clusters generated from mobile phone reviews

screen, resolution, touch, display, color, picture
battery, power, charge, day, cable, charger
quality, break, day, build, buy, control
service, buy, check, help, website, shipping
price, money, worth, cost, charge, free
design, color, metal, case, plastic, silver

useful sentences. Then, we average the sentiment scores on the sentences as the summarized score for each aspect.

Another advantage of the application is the convenience it offers in terms of comparing products of the same type on several prominent review aspects. In Fig. 8, we show an example of the comparison between iPhone 6 and Galaxy S5 on six prominent review aspects. With the summarized sentiment analysis scores on different aspects, we can easily compare the two products in different dimensions such as *screen*, *quality*, *service* and *design*.

Fig. 9 shows an example of detailed reviews of iPhone 6 with different aspects. Users who need more detailed information can click on the link of "customer reviews" and then they can read the original review sentences with the aspect words and highlighted sentiment words annotated using Stanford CoreNLP toolkit [24]. The review snippets can be further grouped by aspects by clicking on the tabs.

This application demonstrates the extracted aspects of ExtRA are effective and the downstream task based on its results can benefit the research and systems about aspect-based review analysis and summarization.

V. Related Work

The majority of existing research related to *aspect-based review analysis* is about how to mine opinion based on given aspects [4], [5] or to jointly extract the aspects and sentiment [3], [7], [25]. Their defined aspect extraction problem is to detect the aspect words in a given sentence, while ours is to extract the most prominent aspects from a large set of review sentences. Some related work focuses on extracting aspect terms while most work does not aim to extract aspect terms or phrases for a particular product type. Our work instead focus on extracting most prominent aspect terms from user reviews. We divide the existing work on this task into two types:

- *rule-based* methods, most of which utilize handcrafted rules to extract candidate aspects and then perform clustering algorithm on them.
- *topic modeling based* methods, which directly model topics from texts and then extract aspects from the topics.

1) Rule-based Methods: For rule-based methods, researchers leverage word statistical and syntactic features to manually design rules, recognizing aspect candidates from texts. Some previous work proposed carefully designed rules for aspect extraction. Poria et al. 2014 [2] uses manually crafted mining rules. Qiu et al. 2011 [3] also used rules, plus the Double Propagation method to better relate sentiment to aspects. Gindl et al. 2013 [26] cooperate the Double Propagation with anaphora resolution for identifying co-references to improve the accuracy. Su et al. 2008 [4] used a clustering method to map the implicit aspect candidates (which were assumed to be noun form of adjectives in the paper) to explicit aspects. Zeng et al. (2013) [5] mapped implicit features to explicit features using a set of sentiment words and by clustering explicit feature-sentiment pairs. Rana et al. (2017) [27] propose a two-fold rules-based model, which uses rules defined on the basis of sequential patterns mined from customer reviews. Their first fold extracts aspects associated with domain independent opinions and the second fold extracts aspects associated with domain dependent opinions. However, the model is restricted by the pre-defined rules and thus does not scale well.

2) Topic Modeling based Methods: Topic models have been used to perform extraction and clustering at the same time. Most existing work are based on two basic models, pLSA [28] and LDA [29]. Applying topic models on user reviews requires extra attention to the nature of review texts. Two features of review texts are often considered in several modifications of topic model. The first feature is the quick shift of topics between sentences, since people express multiple opinions about various aspects within a short piece of text. Sentences close to each other may talk about completely different but related topics. Phrase-based LDA The other feature is the prominence of sentiment. Since reviews express opinions, naturally there are many sentiments, and also there is a strong relationship between the sentiments and the aspects. Many variations of LDA exploits this feature to improve the mining of aspects. Lakkaraju et al. 2011 [6] models in parallel aspects and sentiments per review. Lin et al. 2009 [7] models the dependency between the latent aspects and ratings. Wang et al. 2011 [8] proposed a generative model which incorporates topic modeling technique into the latent rating regression model [30]. Moghaddam et al. 2012 [20] made a nice summarization of some basic variations of LDA for opinion mining. Our method can be thought of as a hybrid approach with topic modeling as one of its elements.

He et al. (2017) [31] propose a neural attention model for identifying aspect terms, but their proposed model is unable to automatically extract the prominent ones from the *representative terms*, which means they have to manually infer the main topics of the extracted terms. Whereas, ExtRA extracts the most prominent aspect terms without human labor, namely the "aspect labels" in their paper. Also, their model does not support mining aspect phrases, while ExtRA can mine the valuable and informative phrases as target prominent aspects.

Most previous work on aspect extraction utilizes variations of topic modeling, and aspects are modeled as topics, that is, word distributions. To the best of our knowledge, most previous work requires manual selection to choose the best word as a representative for each topic. For example in Titov et al. 2008 [21], the authors manually labeled each topic inferred

Fig. 9: Review snippets displayed by the aspects

from mp3 player reviews. On the contrary, our method is designed to automatically select the best words so that the whole process requires no human effort or labels. We argue that this is an important difference for real-world application, since selecting the best words for each product category is still too much effort for websites like Amazon and Yelp, while our method requires no manual processing and thus can be used directly for real-world applications like aspect-based review summarization. Moreover, such automatic aspect extraction enables dynamic change of the aspect words over time to reflect changing customer interest or taste.

Plus, the prominent aspect terms automatically extracted by our framework can be easily fed to downstream aspect-based review analysis and summarization frameworks. The aspect clusters generated by our model can be used for two purposes:

- The top word of each cluster can be used as the basis of review summarization.
- The clusters can be used for identifying aspects in review texts.

In Sec. IV we used a simple neural network model to predict the sentiment score for each aspect. More sophisticated neural network models for single sentence sentiment prediction [23], [32]–[34] can be used. They can form an automatic chain of software to produce more accurate, quantitative review summaries from massive online reviews.

VI. Conclusion

In this paper, we propose our unsupervised framework ExtRA for extracting most prominent aspect terms from user reviews, which is beneficial for both qualitative and quantitative review analysis and aspect-based opinion mining for various types of product or service. We find that directly performing topic modeling alone is not adequate to attack this problem because user reviews tend to switch aspects very quickly within a short text and the topics are not balanced. The proposed unsupervised framework ExtRA solves this problem by slicing the review documents into sentences and then clustering them in sentence level and topic level respectively. Finally we design word ranking algorithm and propose our AspVec to represent the semantics of each aspects, so that we can extract both aspect terms and phrases by computing the semantic similarities. Extensive experimental results show that our approach outperforms many baseline models.

As for the future work, we believe improving the representation models for sentence and topics could offer more improvement. The proposed framework ExtRA can be applied more downstream applications if more information is provided, such as detecting user communities with similar preferred review aspects. Also, ExtRA can be modified to extract aspect terms for many other domains, like question answering forums.

References

[1] 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.

[2] S. Poria, E. Cambria, L.-W. Ku, C. Gui, and A. Gelbukh, "A rule-based approach to aspect extraction from product reviews," in *Proceedings of the Second Workshop on Natural Language Processing for Social Media (SocialNLP)*, 2014, pp. 28–37.

[3] G. Qiu, B. Liu, J. Bu, and C. Chen, "Opinion word expansion and target extraction through double propagation," *Computational linguistics*, vol. 37, no. 1, pp. 9–27, 2011.

[4] Q. Su, X. Xu, H. Guo, Z. Guo, X. Wu, X. Zhang, B. Swen, and Z. Su, "Hidden sentiment association in chinese web opinion mining," in *Proceedings of the 17th international conference on World Wide Web*. ACM, 2008, pp. 959–968.

[5] L. Zeng and F. Li, "A classification-based approach for implicit feature identification," in *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*. Springer, 2013, pp. 190–202.

[6] H. Lakkaraju, C. Bhattacharyya, I. Bhattacharya, and S. Merugu, "Exploiting coherence for the simultaneous discovery of latent facets and associated sentiments," in *SDM*. SIAM, 2011, pp. 498–509.

[7] 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.

[8] H. Wang, Y. Lu, and C. Zhai, "Latent aspect rating analysis without aspect keyword supervision," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2011, pp. 618–626.

[9] Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents," in *ICML*, 2014.

[10] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[11] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Information Theory*, vol. 28, no. 2, pp. 129–136, 1982. [Online]. Available: https://doi.org/10.1109/TIT.1982.1056489

[12] X. Cheng, X. Yan, Y. Lan, and J. Guo, "Btm: Topic modeling over short texts," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 12, pp. 2928–2941, 2014.

[13] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[14] T. Hofmann, "Probabilistic latent semantic analysis," in *UAI*, 1999.

[15] ——, "Unsupervised learning by probabilistic latent semantic analysis," *Machine Learning*, vol. 42, pp. 177–196, 2001.

[16] J. Liu, J. Shang, and J. Han, "Phrase mining from massive text and its applications," *Synthesis Lectures on Data Mining and Knowledge Discovery*, vol. 9, no. 1, pp. 1–89, 2017.

[17] A.-M. Popescu and O. Etzioni, "Extracting product features and opinions from reviews," in *Natural language processing and text mining*. Springer, 2007, pp. 9–28.

[18] J. Pavlopoulos and I. Androutsopoulos, "Aspect term extraction for sentiment analysis: New datasets, new evaluation measures and an improved unsupervised method," *Proceedings of LASMEACL*, pp. 44–52, 2014.

[19] X. Ding, B. Liu, and P. S. Yu, "A holistic lexicon-based approach to opinion mining," in *Proceedings of the 2008 international conference on web search and data mining*. ACM, 2008, pp. 231–240.

[20] S. Moghaddam and M. Ester, "On the design of lda models for aspect-based opinion mining," in *CIKM*, 2012, pp. 803–812.

[21] I. Titov and R. McDonald, "Modeling online reviews with multi-grain topic models," in *WWW*, 2008, pp. 111–120.

[22] S. Li and T.-S. Chua, "Document visualization using topic clouds," *arXiv preprint arXiv:1702.01520*, 2017.

[23] R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *EMNLP*, 2013, pp. 1631–1642.

[24] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky, "The Stanford CoreNLP natural language processing toolkit," in *Association for Computational Linguistics (ACL) System Demonstrations*, 2014, pp. 55–60. [Online]. Available: http://www.aclweb.org/anthology/P/P14/P14-5010

[25] Q. Liu, B. Liu, Y. Zhang, D. S. Kim, and Z. Gao, "Improving opinion aspect extraction using semantic similarity and aspect associations." 2016.

[26] S. Gindl, A. Weichselbraun, and A. Scharl, "Rule-based opinion target and aspect extraction to acquire affective knowledge," in *Proceedings of the 22nd international conference on World Wide Web companion*. International World Wide Web Conferences Steering Committee, 2013, pp. 557–564.

[27] T. A. Rana and Y.-N. Cheah, "A two-fold rule-based model for aspect extraction," *Expert Systems with Applications*, vol. 89, pp. 273–285, 2017.

[28] T. Hofmann, "Probabilistic latent semantic indexing," in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 1999, pp. 50–57.

[29] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *The Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[30] H. Wang, Y. Lu, and C. Zhai, "Latent aspect rating analysis on review text data: a rating regression approach," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2010, pp. 783–792.

[31] R. He, W. S. Lee, H. T. Ng, and D. Dahlmeier, "An unsupervised neural attention model for aspect extraction," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, 2017, pp. 388–397. [Online]. Available: https://doi.org/10.18653/v1/P17-1036

[32] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, 2011, pp. 142–150.

[33] C. N. dos Santos and M. Gatti, "Deep convolutional neural networks for sentiment analysis of short texts." in *COLING*, 2014, pp. 69–78.

[34] D. Tang, B. Qin, and T. Liu, "Document modeling with gated recurrent neural network for sentiment classification," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 1422–1432.