Aspect-Based Sentiment Analysis with Minimal Guidance*

Honglei Zhuang† Timothy Hanratty‡ Jiawei Han§

Abstract

Aspect-based sentiment analysis is an important tool to understand user opinions in a fine-grained manner. Although extensively studied, developing such a tool for a specific domain remains an expensive process. Most existing methods either rely on massive labeled data for training or external language resource and tools which are not necessarily available or accurate. We propose to study the aspect-based sentiment analysis with only a small set of aspect and sentiment seed words as guidance on a target corpus. We first expand the aspect and sentiment lexicons from the given seed words by features created by frequent pattern mining. Then, we develop a generative model to characterize the aspect and sentiment mentions based on their word embedding, and infer the sentiment polarity for sentiment words accordingly. The effectiveness of our method is verified by experiments on two real world data sets.

1 Introduction

Understanding massive text data automatically is highly demanded in various industries. Identifying and classifying sentiment within documents is one of the most important sub-tasks, empowering numerous applications in recommendations [2, 3], stock prediction [23, 19] *etc.*

Aspect-based sentiment analysis [22] provides a fine-grained view of sentiment within documents. It aims to extract different aspects of the entity being reviewed, and determine sentiment corresponding to each aspect individually. For example, "*The hotel has a reasonable price but the room is small*" presents two aspects: "price" and "room", with positive and negative sentiment respectively.

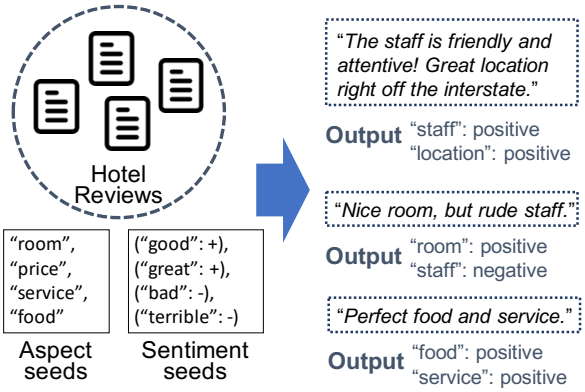


Figure 1: An example of aspect sentiment analysis with minimal guidance.

While many studies on this task adopt supervised methods [9, 14, 29, 6, 4], aspect-level sentiment labels are usually expensive to obtain. This triggers another series of research effort on weakly or distant supervised aspect-based sentiment analysis. However, most of the previous work utilizes external language resource or tools such as thesaurus information [5, 18, 8, 13, 20] or dependency parser [21, 11]. In reality, such resource is not always available or accurate in new domains or low-resource languages.

Therefore, we propose to study aspect-based sentiment with minimal user guidance. The goal is to perform aspect-based sentiment analysis without direct or indirect usage of training data or external language resource, but only a massive target corpus and very little user effort. More concretely, users are only required to provide a small set of seed aspect words and a small set of seed sentiment words with sentiment polarity. The objective is to output identified aspect mentions from each review document, as well as their corresponding sentiment polarity (positive or negative).

EXAMPLE 1. *Figure 1 shows an example. With a massive set of hotel reviews, users only need to provide a small set of seed aspect words like* {*"room", "price", ...*} *as well as a small set of seed sentiment words* {*"good", "terrible", ...*}*. Users also need to provide sentiment*

*Research was sponsored in part by U.S. Army Research Lab. under Cooperative Agreement No. W911NF-09-2-0053 (NSCTA), DARPA under Agreement No. W911NF-17-C-0099, National Science Foundation IIS 16-18481, IIS 17-04532, and IIS-17-41317, DTRA HDTRA11810026, and grant 1U54GM114838 awarded by NIGMS through funds provided by the trans-NIH Big Data to Knowledge (BD2K) initiative (www.bd2k.nih.gov). Any opinions, findings, and conclusions or recommendations expressed in this document are those of the author(s) and should not be interpreted as the views of any U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

†Department of Computer Science, University of Illinois at Urbana-Champaign (hzhuang3@illinois.edu)

‡US Army Research Lab (timothy.p.hanratty.civ@mail.mil)

§Department of Computer Science, University of Illinois at Urbana-Champaign (hanj@illinois.edu)

polarity labels for the seed sentiment words (e.g. "good" is positive and "terrible is negative"). For each review document, the algorithm should be able to identify aspect mentions within the document and output sentiment polarity label for each identified aspect mention. For example, for a review document "Nice room, but rude staff"*, the algorithm should output two identified aspect words "room" and "staff", even if "staff" is not a seed aspect word. The algorithm should also assigns positive label to "room" and negative label to "staff".*

This setting is usual when one needs to develop an aspect-based sentiment analysis tool for a new domain, such as online forums of a certain field. It is expensive to produce sufficient labeled data for training a supervised model. External language resource is usually only created for general field and thus does not necessarily have good coverage. Existing NLP tools are often inaccurate on specific domain without extra learning, especially for informal corpus like user generated content. However, our setting is much more realistic as users only need to specify around 10 seed aspect and sentiment words.

The challenges in this problem setting are: 1) how to leverage the limited user guidance to identify aspects as complete as possible; 2) how to classify sentiment for each aspect in a document with limited user guidance. The only signals we can leverage for both tasks are the user provided seed sets and the corpus.

We make the following contributions. First, we develop a method to expand the aspect and sentiment lexicons from given set of seed words based on features extracted by frequent pattern mining. Second, we propose a generative model to characterize the generation of aspect and sentiment mentions, represented by their word embedding. By inferring the model, we can assign sentiment polarity to words in the sentiment lexicon. Finally, we can accordingly perform aspect-based sentiment analysis on each document. We verify the effectiveness on two real world data sets.

We present the detail of our work below.

2 Preliminaries

In this section, we formalize the research problem, and briefly introduce the framework of our pipeline.

2.1 Problem formalization. We denote a set of documents as $\mathcal{D} = \{d_i\}_{i=1}^n$. A document d_i consists of a sequence of sentence $d_i = (s_1, \cdots, s_{|d_i|})$. Each sentence s_j can be represented as a sequence of tokens $s_j = (w_1, \cdots, w_{|s_j|})$, where each w_k takes a value from a vocabulary $\mathcal{V}$.

For each word w in the vocabulary $\mathcal{V}$, one can derive an embedding vector from word embedding technique (*e.g.* [17]). More precisely, word embedding provides a function $f : \mathcal{V} \mapsto \mathbb{R}^\nu$, where ν is the number of dimensions of the embedding space. The semantic proximity between two words should be reflected by the similarity of their embedding vectors. A popular similarity measure is cosine similarity, defined as:

$$\mathrm{sim}\big(f(w), f(w')\big) = \frac{f(w) \cdot f(w')}{\|f(w)\| \times \|f(w')\|}$$

Notice that each $w \in \mathcal{V}$ is not necessarily a unigram word, but may also refer to a multigram phrase (*e.g.* "air conditioner", "mini bar"), or a subword like "n't" in "don't". We will use the term "word" to refer to any elements from the vocabulary $\mathcal{V}$ unless otherwise noted.

Each document d_i usually contains a few aspect mentions $\mathbf{a}_i = \{a_1, \cdots, a_{|\mathbf{a}_i|}\}$, where each aspect mention is a token in d_i. Each aspect mention is associated with a sentiment label, which can be represented as $\mathbf{y}_i = \{y_1, \cdots, y_{|\mathbf{a}_i|}\}$ collectively, where $y_k \in \mathcal{Y}$ represents the sentiment label for aspect mention a_k. In this paper, we only focus on a binary setting, namely $\mathcal{Y} = \{0, 1\}$, where 0 stands for *negative* sentiment, and 1 stands for *positive* sentiment.

Users can provide guidance as seed aspect and sentiment words, which are essentially subsets of the vocabulary, denoted as $\mathcal{V}_A^{(0)}, \mathcal{V}_S^{(0)} \subset \mathcal{V}$ respectively. Moreover, for each seed sentiment word $w \in \mathcal{V}_S^{(0)}$, users also provide its sentiment label, denoted as $r_0(w) \in \mathcal{Y}$.

The problem can be formalized as:

PROBLEM 1. *Given a corpus $\mathcal{D}$, a small set of seed aspect words $\mathcal{V}_A^{(0)}$ and a small set of seed sentiment words $\mathcal{V}_S^{(0)}$ with their labels $r_0 : \mathcal{V}_S^{(0)} \mapsto \mathcal{Y}$, we aim to identify the aspect mentions $\mathbf{a}_i$ as well as their sentiment labels $\mathbf{y}_i$ for each $d_i \in \mathcal{D}$.*

Notice that there are studies in which an aspect is defined as a set or a distribution of aspect words. Although our output is only formalized as aspect mentions, we can always perform a clustering method to aggregate aspect words into more abstract aspects. However, this is beyond the scope of this paper and we would address this in our future work.

2.2 Framework. We tackle the problem in two steps. First, we expand the aspect and sentiment lexicons from very small sets of seed words. Second, we identify the aspect and sentiment mentions in each document and classify the sentiment polarity.

Lexicon expansion. Aspect and sentiment lexicons serve as strong tools to identify the most essential signals for aspect-based sentiment analysis.

For a given domain of reviews, the aspect lexicon $\mathcal{V}_A \subset \mathcal{V}$ should contain all the words characterizing

possible factors of the entity to be reviewed, such as "location", "price", "service" in hotel reviews.

The sentiment lexicon $\mathcal{V}_S \subset \mathcal{V}$ should have all the words that express or imply an attitude or emotion. General sentiment words include "good", "great", "terrible", while some sentiment words can also be domain dependent. For example, words like "renovated", "air conditioned" imply positive sentiment in hotel reviews but not necessarily in other domains.

The goal of this stage is to construct the aspect lexicon $\mathcal{V}_A$ and the sentiment lexicon $\mathcal{V}_S$ from a given review corpus $\mathcal{D}$ as precise and complete as possible merely from the seeds $\mathcal{V}_A^{(0)}$ and $\mathcal{V}_S^{(0)}$.

Sentiment classification. With the aspect and sentiment lexicons available, we can identify aspect and corresponding sentiment mentions from documents. However, we do not have sentiment polarity labels for most of the sentiment words in $\mathcal{V}_S$. Hence, we need to perform sentiment polarity assignment before we can conduct sentiment classification of documents.

The objective is to derive a mapping function $r : \mathcal{V}_S \mapsto [0, 1]$, which assign sentiment polarity rating $r(w)$ for all the words $w \in \mathcal{V}_S$. Again, only sentiment words in the seed set $\mathcal{V}_S^{(0)}$ are given polarity labels as input by $r_0(\cdot)$, while other sentiment words mined from the previous stage remain unlabeled.

Once we obtain the sentiment polarity for the entire sentiment lexicon, we can generate sentiment labels $\mathbf{y}_i$ for aspect mentions $\mathbf{a}_i$ in each document d_i.

3 Lexicon Expansion

In this section, we introduce a method based on frequent pattern mining to expand aspect and sentiment lexicons from given seed words $\mathcal{V}_A^{(0)}$ and $\mathcal{V}_S^{(0)}$.

The general idea is to select a few context pattern as features to characterize each word w. Then, we can train classifiers to extract new aspect and sentiment words iteratively.

There are three modules of our method:

- *Pattern feature mining.* Mining and selecting pattern features that can effectively characterize aspect and sentiment words.
- *Aspect lexicon expansion.* Using pattern features and currently mined sentiment lexicon to expand aspect lexicon.
- *Sentiment lexicon expansion.* Using pattern features and currently mined aspect lexicon to expand sentiment lexicon.

Notice that our method does not rely on NLP parsing tools such as Part of Speech (PoS) tagging or dependency parsing. Although NLP tools may provide strong syntactic signals in this task, they are usually trained on a more general corpus and often suffer from higher error rate on user generated review data. Furthermore, applying NLP pipeline is much slower than frequent pattern mining.

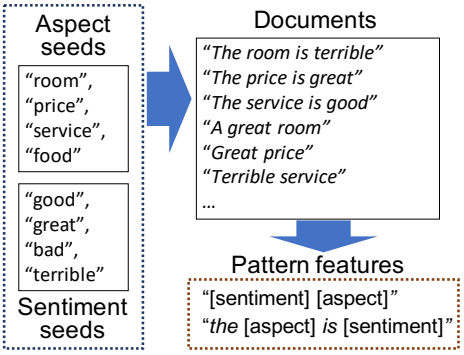


Figure 2: An example of mining pattern features.

We will introduce each module in detail.

3.1 Pattern feature mining.

An important category of signals to characterize whether a word w is in aspect/sentiment lexicon are the frequencies of specific context patterns around w. For example, if the pattern "*the w is great*" occurs sufficiently frequent, w will probably be in the aspect lexicon. Similarly, if the pattern "*a very w bed*" is frequent, then w should be in the sentiment lexicon.

Precisely, a pattern feature p is an ordered sequence of tokens or aliases, represented as $p = (t_1, \cdots, t_l)$, where t_i is either a word from $\mathcal{V}$ or an alias that could be substituted by any from a set of words. In our setting, possible aliases are "[aspect]" or "[sentiment]", which can be substituted by any currently known aspect or sentiment words respectively.

It is intractable to enumerate all the possible context patterns of a word w, and most of them are not necessarily informative in determining whether w is in the aspect/sentiment lexicon. Apparently, patterns such as "a w" or "w of" are not very useful. Therefore, we adopt the following mechanism to mine and select informative pattern features, denoted as $P = \{p_1, p_2, \cdots\}$.

Frequency. Intuitively, frequent patterns containing both (currently known) aspect and sentiment words would be a good candidate pool.

In order to mine such frequent patterns, we build a subset of sentences $\mathcal{S}$ containing both aspect and sentiment words from the seed set. As shown in Figure 2, only sentences containing words from both the aspect seeds and the sentiment seeds are selected.

Moreover, we treat all aspect (sentiment) words as a unified alias ("[aspect]", "[sentiment]") and turn all

the words into lower case, so similar patterns can be merged together. As an example in Figure 2, the first three sentences "*The room is terrible*", "*The price is great*" and "*The service is good*" would all be converted into the same pattern "the [aspect] is [sentiment]".

We then perform a contiguous sequential frequent pattern mining algorithm based on the derived set $\mathcal{S}$. The relative minimum support is set to $\theta = 0.005$, namely only patterns with frequency larger than or equal to $\theta|\mathcal{S}|$ would be mined.

Representativeness. Some frequent patterns do not contain any aspect or sentiment aliases (*e.g.* "is a"), which cannot serve as pattern features in the following modules. Therefore, we only keep the patterns containing both a sentiment alias and an aspect alias.

Patterns crossing multiple clauses usually do not serve as good features. Sometimes patterns like "[sentiment], [aspect] is" would be mined as frequent patterns while they do not necessarily reflect any interplay between the aspect and the sentiment word. We simply remove any patterns containing non-alphabetic tokens.

Concordance. Sometimes a pattern is frequent only because it has a frequent sub-pattern and does not necessarily provide novel information. For example, "[sentiment] [aspect]" is frequent and informative, but "[sentiment] [aspect] on" may be redundant even if it is frequent. The latter becomes frequent only because the former is so frequent that a co-occurred random word would produce another frequent pattern.

Therefore, we perform a test of independence to filter such redundant patterns. For each mined frequent pattern p, we test against the null hypothesis that it is merely generated randomly by attaching a word w to its immediate sub-pattern p'. By "immediate", we mean p and p' only differs by one word at the boundary, namely $p = p' \oplus w$ or $p = w \oplus p'$, where $\oplus$ means concatenation. We calculate a z-score as a test of independence [15]:

$$z(p, p') = \frac{c(p) - c(p')\mathbb{P}(w)}{\sqrt{c(p')\mathbb{P}(w)(1 - \mathbb{P}(w))}} \tag{3.1}$$

where $\mathbb{P}(w)$ is the relative frequency of w in $\mathcal{D}$.

We filter pattern p and all of its super-patterns if there exists any sub-pattern p' such that $z < \Phi^{-1}(1-\alpha)$ where Φ^{-1} is the probit function (*i.e.* inverse cumulative distribution function of a standard Gaussian distribution). We set $\alpha = 10^{-3}$ in our experiments.

Only patterns satisfying all the above criteria will be selected as pattern features in P.

3.2 Aspect/Sentiment Lexicon Expansion.

The basic idea of this module is to utilize the mined pattern features P to build a classifier to determine if a word is an aspect/sentiment word. We can then run the aspect lexicon expansion and sentiment lexicon expansion iteratively until convergence.

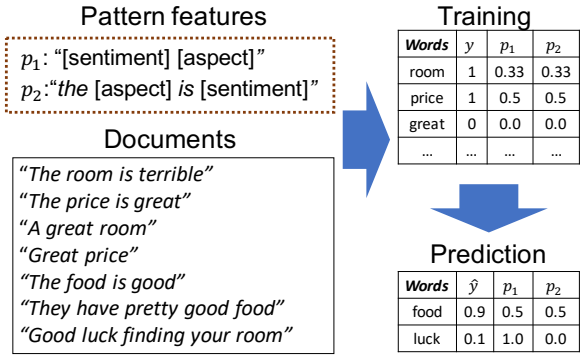


Figure 3: An example of expanding aspect lexicons from pattern features. Notice that y and $\hat{y}$ here represents whether a word is in the aspect lexicon. It is *not* its sentiment polarity label.

Both lexicons can be expanded in a similar mechanism. We only focus on describing aspect lexicon expansion below, while the sentiment lexicon expansion is symmetric.

Suppose we are at the t-th iteration and with mined aspect and sentiment lexicons $\mathcal{V}_A^{(t)}$ and $\mathcal{V}_S^{(t)}$. Our first step is to extract a set of candidate aspect words $\mathcal{U}_A^{(t+1)}$, where each candidate aspect word occurs at least once with a pattern feature $p_i \in P$, *i.e.* $\exists p_i \in P, c(p_i(w)) > 0$, where $c(\cdot)$ counts the frequency in $\mathcal{D}$ and $p_i(w)$ is a pattern by substituting w into "[aspect]" ("[sentiment]" for sentiment lexicon expansion) in p_i.

Then we adopt a supervised classifier to determine which candidate words in $\mathcal{U}_A^{(t+1)}$ should be an aspect word. We build feature vectors $\mathbf{x}_j$ for each word w_j by counting relative frequencies of all the patterns p_i in P with regard to w_j, namely $\mathbf{x}_j = [\frac{c(p_i(w_j))}{c(w_j)}]_{p_i \in P}$.

We treat words in $\mathcal{V}_A^{(t)}$ as positive samples and words in $\mathcal{V}_S^{(t)}$ as negative samples to train a random forest classifier. Then we apply the classifier to words in $\mathcal{U}_A^{(t+1)}$ to obtain the predicted value $\hat{Y}_A^{(t+1)}$, where $0 \leq \hat{y}_j \leq 1$ is the classifier's confidence value that candidate w_j is an aspect word.

Finally, we expand those candidate words w_j with $\hat{y}_j \geq \tau$ into the aspect lexicon $\mathcal{V}_A^{(t+1)}$, where $\tau = 0.8$ is a threshold. We also add candidate words w_j with embedding vectors close to any known aspect words $w_{j'} \in \mathcal{V}_A^{(t)}$ into the expanded lexicon, as long as $\text{sim}(f(w_j), f(w_{j'})) \geq \beta$ and $\hat{y}_j \geq 0.5$, where β is another threshold.

EXAMPLE 2. *Figure 3 shows an example. With the given pattern features, we can first extract the candidate aspect words "food" and "luck" as they fit into the pattern features' aspect alias for at least once. We then build feature vectors for all the candidates and known aspect/sentiment words. We treat known aspect words ("room", "price", ...) as positive samples and sentiment words ("great", ...) as negative samples to train a random forest classifier. Finally, we can obtain the prediction results $\hat{y}$'s for candidate aspect words.*

3.3 Iterative expansion. With a given set of pattern features P, we can perform aspect and sentiment lexicon expansion. Moreover, with the newly expanded aspect words, more candidate sentiment words can be extracted and values of pattern features for candidate sentiment words can be updated. This may trigger more sentiment words to be expanded and vice versa. Therefore, we iteratively perform aspect and sentiment word expansion until convergence.

4 Aspect-Based Sentiment Classification

In this section, we leverage the aspect and sentiment lexicons to perform aspect-based sentiment classification on documents.

The general idea is to utilize the embedding vector of sentiment words. First, we try to derive a "positive vector" and a "negative vector" in the semantic space modeled by word embedding based on the seed sentiment words and the corpus. Then, we can assign sentiment polarity to words in the sentiment lexicon $\mathcal{V}_S$ based on how close they are to the positive or the negative vector. Finally, polarity of sentiment mentions in each document can be aggregated to obtain sentiment classification results.

4.1 A simple baseline. We start by introducing a relatively straightforward baseline.

Polarity assignment in sentiment lexicon. We introduce a naïve way to assign polarity labels to words in sentiment lexicon. Notice that for each word in the seed sentiment lexicon $w_j \in \mathcal{V}_S^{(0)}$, a polarity label $r_0(w_j) \in \{0, 1\}$ is given. We use $\mathcal{V}_{S+}^{(0)}$ and $\mathcal{V}_{S-}^{(0)}$ to represent subset of seed sentiment words with positive and negative polarity labels respectively.

We can derive a "positive vector" $\mathbf{v}_+$ and a "negative vector" $\mathbf{v}_-$ by taking the average embedding vectors of words with positive and negative labels respectively:

$$\mathbf{v}_+ = \frac{1}{|\mathcal{V}_{S+}^{(0)}|} \sum_{w_j \in \mathcal{V}_{S+}^{(0)}} f(w_j), \quad \mathbf{v}_- = \frac{1}{|\mathcal{V}_{S-}^{(0)}|} \sum_{w_j \in \mathcal{V}_{S-}^{(0)}} f(w_j)$$

where $f(w_j)$ represents the embedding vector of word w_j, as mentioned in Section 2.

Intuitively, words with embedding vector closer to $\mathbf{v}_+$ are more likely to convey positive sentiment, while words with embedding vector closer to $\mathbf{v}_-$ would be more likely negative. Therefore, we can assign sentiment polarity scores to all the other words in the sentiment lexicon. Suppose for a word $w_j \in \mathcal{V}_S \setminus \mathcal{V}_S^{(0)}$, we can assign a sentiment rate $r(w_j) \in [0, 1]$:

$$r(w_j) = \varphi\big(\text{sim}(\mathbf{v}_+, f(w_j)) - \text{sim}(\mathbf{v}_-, f(w_j))\big)$$

where $\varphi(\cdot)$ is the standard logistic function. Meanwhile, if a word w_j is in seed sentiment words, it will still keep its ground-truth label $r(w_j) = r_0(w_j)$.

Based on the polarity scores of sentiment lexicon, we can further obtain the sentiment classification label of each document.

Aspect-based sentiment classification. For each document d_i, we extract all of the mentions of words in aspect lexicon, denoted as $\mathbf{a}_i = \{a_k \in d_i | a_k \in \mathcal{V}_A\}$. We also extract all of the sentiment mentions and map them into the closest aspect mention within the same sentence. Therefore, for the k-th aspect mention a_k, there is a set of sentiment mentions $\mathbf{o}_{ik}$. Aspect mentions with $|\mathbf{o}_{ik}| = 0$ are not included in practices.

We can aggregate the sentiment polarity scores for each aspect mentions $a_k \in \mathbf{a}_i$. We calculate y_k as the sentiment polarity score of the k-th aspect mention a_k in document d_i by averaging the sentiment polarity scores for all the words in $\mathbf{o}_{ik}$:

$$y_k = \mathbb{I}\Big[\frac{1}{|\mathbf{o}_{ik}|} \sum_{o_j \in \mathbf{o}_{ik}} r(o_j) > 0.5\Big]$$

where $\mathbb{I}(\cdot)$ is an indicator function.

Thereby we can obtain final output $\mathbf{a}_i$ and $\mathbf{y}_i = \{y_k\}_{k=1}^{|\mathbf{a}_i|}$ for document d_i.

4.2 Rectification of polarity assignment. The positive and negative vectors $\mathbf{v}_+$, $\mathbf{v}_-$ utilized above are derived from a very small labeled lexicon. Hence, they may not be sufficiently accurate to reflect the actual distribution of positive and negative sentiment words in the embedding space.

We utilize a novel model to rectify their directions. We propose a graphical model to characterize the generation of aspect and sentiment words in review data, where the means of the two hidden sentiment word distributions corresponds to the "real" positive and negative vectors. By inferring the model, we can obtain more accurate positive and negative vectors.

Instead of using a mixture of multinomial distributions to generate words, we model the generation of each word's embedding vector directions by a mixture

of von-Mises Fisher distributions. The von Mises-Fisher (vMF) distribution is a widely adopted distribution in directional statistics to model unit vectors in a spherical space and shows stronger power [1, 28] than Gaussian in modeling embedding vectors in different applications. Its formalized definition can be found in the supplementary file.

Model. Our model assumes T hidden aspect vMF distributions and 2 hidden sentiment vMF distributions. For each document d_i, an aspect multinomial distribution θ_i^A and T sentiment multinomial distributions $\{\theta_{i,t}^S\}_{t=1}^T$ will be generated respectively. Each aspect mention $a_k \in \mathbf{a}_i$ will be assigned with a label z_k generated from θ_d^A, while each associated sentiment mention $o_j \in \mathbf{o}_{ik}$ will be generated from θ_{i,z_k}^S. Then, the unit vector of each word will be generated from corresponding vMF distributions as indicated by their labels.

To summarize:

$$\begin{aligned}
\theta_i^A &\sim \text{Dirichlet}(\cdot|\boldsymbol{\alpha}_A), && d_i \in \mathcal{D}\\
\theta_{i,t}^S &\sim \text{Dirichlet}(\cdot|\boldsymbol{\alpha}_S), && d_i \in \mathcal{D}, t \in [T]\\
z_k &\sim \text{Categorical}(\cdot|\theta_i^A), && a_k \in \mathbf{a}_i\\
y_j &\sim \text{Categorical}(\cdot|\theta_{i,z_k}^S), && o_j \in \mathbf{o}_{ik}\\
\mathbf{x}_{a_k} &\sim \text{vMF}(\cdot|\boldsymbol{\mu}_{z_k}^A, \kappa_{z_k}^A), && a_k \in \mathbf{a}_i\\
\mathbf{x}_{o_j} &\sim \text{vMF}(\cdot|\boldsymbol{\mu}_{y_j}^S, \kappa_{y_j}^S), && o_j \in \mathbf{o}_{ik}
\end{aligned}$$

We infer the model by Gibbs sampling. We specify the prior for sentiment vMF distribution's mean vector $\boldsymbol{\mu}_y^S$ as vMF centered as the mean direction of seed sentiment words with label y as a guidance. The technical details are omitted due to limited space, but can be found in the supplementary file.

By estimating the mean direction $\hat{\boldsymbol{\mu}}_y^S$ for the sentiment vMF distributions, we can derive the rectified positive and negative vectors $\mathbf{v}_+ = \hat{\boldsymbol{\mu}}_1^S$ and $\mathbf{v}_- = \hat{\boldsymbol{\mu}}_0^S$:

$$\mathbf{v}_+ = \frac{C_S\boldsymbol{\mu}_S + \kappa_y^S\mathbf{x}_{S,1}}{\left\|C_S\boldsymbol{\mu}_S + \kappa_1^S\mathbf{x}_{S,1}\right\|},\ \mathbf{v}_- = \frac{C_S\boldsymbol{\mu}_S + \kappa_0^S\mathbf{x}_{S,1}}{\left\|C_S\boldsymbol{\mu}_S + \kappa_0^S\mathbf{x}_{S,0}\right\|}$$

where $\mathbf{x}_{S,1}$ and $\mathbf{x}_{S,0}$ are the sum of unit embedding vector of sentiment mentions with inferred hidden variable as 1 and 0 respectively.

We can use these rectified vectors to perform polarity assignment for sentiment lexicon and aspect-based sentiment classification as the baseline.

5 Experiments

In this section, we verify the effectiveness of our proposed methods on real world review data sets.

5.1 Data set. We introduce the data sets used in our experiments.

Table 1: Performance comparison of aspect lexicon expansion (%).

Data set	Method	P	R	F_1
Hotel	HU	60.95	43.90	51.04
	DP	51.56	99.89	68.01
	PF	80.27	72.99	76.46
Restaurant	HU	85.80	35.39	50.11
	DP	57.54	99.37	72.88
	PF	87.88	68.25	76.83

Table 2: Performance comparison of sentiment lexicon expansion (%).

Data set	Method	P	R	F_1
Hotel	HU	48.43	95.37	64.24
	DP	70.75	84.30	76.93
	PF	84.83	75.94	80.14
Restaurant	HU	38.37	97.33	55.04
	DP	59.36	93.27	72.55
	PF	84.71	70.58	77.00

Hotel. We utilize a hotel review data set from [25, 26]. In the hotel data, reviewers can provide 1 to 5 star ratings on several aspects such as room, service *etc.*We utilize reviews from 181 hotels, as hotels with less than 50 reviews are removed, which results in 17,865 reviews.

Restaurant. We create a randomly sampled subset of 10,000 public Yelp[1] restaurant reviews. For the purpose of evaluation, we also include 6,060 labeled sentences from restaurant reviews [4], where each sentence is labeled with a set of aspect mentions along with their corresponding sentiment orientation. Sentences without aspect words or with neutral sentiment are removed in our evaluation.

For both data sets, we preprocess with phrase mining [10] and then train a 200 dimension word embedding by word2vec [17]. Details are described in the supplementary file.

5.2 Lexicon expansion. We first evaluate the task of lexicon expansion.

Methods evaluated. We compare the following lexicon expansion methods.

- *Frequency-based method (HU).* A lexicon expansion method based on word frequencies and their PoS tags, proposed by Hu *et al.* [5]. We use an NLP pipeline[2] to parse sentences.
- *Double propagation (DP).* A double propagation method proposed by Qiu *et al.* [21] that relies on hand-crafted rules based on dependency information. We use the same NLP pipeline as above to obtain the syntactic information.

[1] https://www.yelp.com/dataset/challenge
[2] https://spacy.io/

Table 3: Performance comparison of aspect-based sentiment classification (%)

Data set	Method	P	R	F_1
Hotel	DP	37.17	8.45	13.77
	ASUM	24.82	74.41	37.23
	HU+EMB+R	16.05	57.43	25.08
	DP+EMB+R	45.58	33.72	38.76
	PF+EMB	37.44	50.02	42.83
	PF+EMB+R	44.60	52.04	48.03
Restaurant	DP	57.45	5.31	9.72
	HU+EMB+R	58.33	22.03	31.98
	DP+EMB+R	74.47	22.95	35.09
	PF+EMB	78.89	22.30	34.76
	PF+EMB+R	74.41	26.69	39.29

* *Pattern features (PF).* Our proposed method based on pattern feature mining.

Evaluation metrics. We use a pooling strategy to generate the ground-truth. We obtain the aspect and sentiment lexicons generated by all the evaluated methods, and label the union of all the lexicons. We only label words with frequency no less than 50 due to the large lexicon size. We evaluate the performance by weighted versions of precision (P), recall (R) and F_1-score (F_1), where each word is weighted by its frequency in the corpus. Similar measures are adopted in [12].

Experiment setup. We use 10 seed aspect words and 10 seed sentiment words for each data set. Notice that the size of seed in our experiments is substantially smaller than the seed sets in previous studies such as [21], where more than 1,000 seed sentiment words are used. We set β to 0.7 for both data sets.

Results. We present the results in Table 1 and Table 2. It can be observed that our method outperforms baselines in terms of F_1 score on both tasks and both data sets. While DP generally has higher recall, its rules are developed for product reviews and are likely to generate a lot of false positives. In comparison, our method does not rely on domain specific rules. We achieve the highest precision in both data sets and both tasks by our relative prudent expansion method, while keeping a decent recall.

5.3 Sentiment classification. We also evaluate the performance of aspect-based sentiment classification.

Methods evaluated. The following methods are compared in our experiments.

- *Double propagation (DP).* The method proposed in [21], which includes both lexicon expansion and polarity assignment.
- *Aspect and sentiment unification model (ASUM).* The method proposed by [7].
- *HU/DP+EMB+R.* Feeding the lexicon constructed by a previously mentioned baseline into our sentiment classification method.
- *PF+EMB.* Our proposed baseline method merely using embedding vector and seed sentiment words.

* *PF+EMB+R.* Our proposed method with rectified polarity assignment of sentiment lexicon.

Evaluation metrics. For Hotel data set, we label documents with 1 or 2 star rating as negative sentiment, while 4 or 5 star rating as positive sentiment for each ground-truth aspect. Since user rating ground-truth is only available for more "general" aspects, we carefully pick a set of frequent aspect words A_k for each ground-truth aspect k. In evaluation, we take the average of the output sentiment polarity labels of aspect mentions corresponding to the same ground-truth aspect as the aggregated output.

If a document does not have user rating on ground-truth aspect k or does not contain words from A_k, then it is not evaluated on ground-truth aspect k. Moreover, documents with 3-star rating on k are also not evaluated on ground-truth aspect k.

We evaluate the performance by precision (P), recall (R) and F_1-score (F_1). Notice that reviews with positive sentiment are overwhelmingly more than reviews with negative sentiment in our data set, which makes the prediction a relatively trivial task. Hence we treat negative sentiment as "positive" label while calculating the evaluation measures.

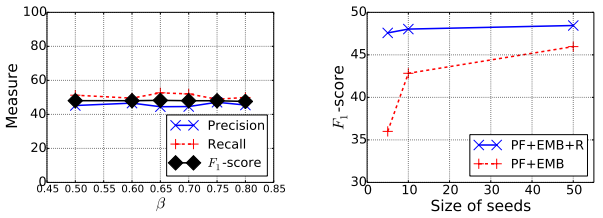
Results. The results are shown in Table 3. It can be observed that our method performs the best in terms of F_1-score. On both data sets, our method constantly achieves around +5% improvement over the best performed baselines.

The results confirm that our lexicons are better than lexicons constructed by other baselines. Our method achieves +5-10% improvement in terms of F_1 on both data sets comparing to the same method with other baseline lexicons.

Another observation is that our rectification step substantially improves the performance. On both data sets, it achieves around +4% of improvement. This is because it combines the embedding signals with the information from the corpus.

Notice that our evaluation setting is much more challenging due to the minimal supervision and imbalance distribution of sentiment labels. Thus the performance is generally lower than typical sentiment analysis.

Parameter analysis. We first study the sensitivity of threshold parameter β in lexicon expansion. We measure the performance of aspect-based sentiment classification on Hotel data set based on lexicon constructed



(a) β (b) Number of seeds

Figure 4: Performance w.r.t several parameters.

Table 4: Case study. For each sentence, we show the identified aspect mentions, sentiment mentions and their sentiment polarity assignment for each method.

Method	Aspect	(Sentiment, Polarity)
Sentence	*The bathroom is large.*	
PF+EMB+R	bathroom	(large, +)
PF+EMB	bathroom	(large, −)
Sentence	*Quiet room.*	
PF+EMB+R	room	(quiet, +)
PF+EMB	room	(quiet, −)
Sentence	*We found our bedding sooooo awful!*	
PF+EMB+R	bedding	(awful, −)
DP+EMB+R	bedding	(sooooo, +), (awful,−)
Sentence	*Reception staff were not friendly and occasion quite rude*	
PF+EMB+R	staff	(not friendly, −), (rude, −)
DP+EMB+R	occasion	(not friendly, −), (rude, −)

with β set to different values between 0.5 to 0.8. As Figure 4(a) suggests, the performance remains stable. The difference of F_1-score is within 1%.

We also study how the performance change w.r.t. the size of seed set. As Figure 4(b) shows, with only 5 seeds for each lexicon, the performance of our method is higher than our baseline with 50 seeds. This shows the power of our rectification method in the sentiment classification stage.

Case study. Table 4 shows a case study to provide an in-depth analysis of how our method outperforms other method. We majorly compare our PF+EMB+R method with our proposed baseline PF+EMB, as well as a variation with lexicon built by DP.

The first two sentences in Table 4 show how our rectification method improves the performance. In the first sentence, our method with sentiment rectification can correctly assign a positive polarity score to "large", while the baseline without rectification mistakenly mark it as negative. Similarly, the baseline recognizes "quiet" as a negative sentiment word, while the rectified version correctly identifies it as positive.

The other two sentences shows how the lower precision of lexicon affects the overall sentiment classification performance. In the sentence "*We found our bedding sooooo awful!*", the lexicon constructed by DP mistakenly take "sooooo" as a sentiment word, while it actually should be an intensifier with informal spelling. In the last sentence, the misspelled "occasion" is identified as an aspect mention by DP lexicon. It steals the sentiment mentions "friendly" and "rude" from the actual aspect mention "staff" in this sentence as we assign sentiment mentions to the closest aspect mention. However, our lexicon correctly output the only aspect mention "staff" in this sentence.

6 Related Work

In this section, we will review unsupervised and weakly supervised effort in aspect-based sentiment analysis.

Aspect and sentiment lexicon expansion. Aspect and sentiment words (*a.k.a.* opinion words) play an important role in aspect-based sentiment analysis. The strategy to start with seed sets of words and expand the lexicon from a corpus is also adopted previously.

Hu *et al.* [5] use a frequency based method to identify frequent nouns as aspect words. Then they extract adjacent adjectives to aspect words as sentiment words. However, they rely on PoS tagging information. Another study using frequency based method is [20], but they rely on more external resource such as the web statistics and WordNet.

Qiu *et al.* [21] propose to expand by syntactic rules. They first parse each sentence in the corpus to obtain PoS tags and dependency structures. Then they expand the aspect and sentiment lexicons by a set of user selected syntactic rules. There are two drawbacks of their method. The first is they heavily rely on the correctness of the dependency parser. However, the effectiveness of dependency parser on cannot be guaranteed on a new domain. Another drawback is that they need user specified rules. Although there are some follow-up studies to improve this algorithm, they still suffer from these drawbacks [11, 12].

Other studies merely focus on either aspect lexicon [27] or sentiment lexicon [13] expansion. They either utilize additional language resources or require human effort to produce rules.

Polarity assignment in sentiment lexicon. A number of studies propose to automatically assign polarity to sentiment words from seeds.

A typical method is proposed in [5], where the sentiment polarity is propagated on the network based on synonym/antonym relations. Similar idea is also adopted in [18, 8, 13, 20]. However, such methods heavily rely on the external resource, which is not always available or accurate.

Another common intuition is to utilize the different levels of syntactic signals from the corpus. For example, [21] specifies several rules based on dependency relations to assign polarity.

Aspect-based sentiment analysis. There are several other studies on aspect-based sentiment analysis.

Wang *et al.* [25, 26] have a series of work on utilizing generative model to predict rating on each aspect. Similarly, Titov and McDonald [24] propose a multi-aspect sentiment model to jointly model the aspect and sentiment rating of users. They both need to utilize the rating information from data. Our model does not require the rating information and thus can be applied to more data sets.

Mei *et al.* [16] propose a joint topic model for the dynamics of topics as well as the general sentiment, but focus more on summarizing the entire corpus instead of classifying aspect-level sentiments for each document.

Jo and Oh [7] propose a generative topic model. They use a mixture of joint aspect-sentiment topics to model the generation of each sentence with a seed set of sentiment words.

7 Conclusion

We study to perform aspect-based sentiment analysis with minimal user guidance. We start with a lexicon expansion step and then develop a generative model to improve sentiment classification based on word embedding. This facilitates building a sentiment analysis tool for domains with limited resource. In principle, this work is language agnostic and can be seamlessly extended to other language, which would be an interesting direction for future work.

References

[1] K. Batmanghelich, A. Saeedi, K. Narasimhan, and S. Gershman. Nonparametric spherical topic modeling with word embeddings. In *ACL*, 2016.

[2] K. Bauman, B. Liu, and A. Tuzhilin. Aspect based recommendations: Recommending items with the most valuable aspects based on user reviews. In *KDD*, 2017.

[3] L. Chen, G. Chen, and F. Wang. Recommender systems based on user reviews: the state of the art. *User Modeling and User-Adapted Interaction*, 2015.

[4] R. He, W. S. Lee, H. T. Ng, and D. Dahlmeier. Exploiting document knowledge for aspect-level sentiment classification. In *ACL*, 2018.

[5] M. Hu and B. Liu. Mining and summarizing customer reviews. In *KDD*, 2004.

[6] W. Jin, H. H. Ho, and R. K. Srihari. Opinionminer: a novel machine learning system for web opinion mining and extraction. In *KDD*, 2009.

[7] Y. Jo and A. H. Oh. Aspect and sentiment unification model for online review analysis. In *WSDM*, 2011.

[8] J. Kamps, M. Marx, R. J. Mokken, and M. d. Rijke. Using wordnet to measure semantic orientations of adjectives. In *LREC*, 2004.

[9] F. Li, C. Han, M. Huang, X. Zhu, Y.-J. Xia, S. Zhang, and H. Yu. Structure-aware review mining and summarization. In *COLING*, 2010.

[10] J. Liu, J. Shang, C. Wang, X. Ren, and J. Han. Mining quality phrases from massive text corpora. In *SIGMOD*, 2015.

[11] Q. Liu, Z. Gao, B. Liu, and Y. Zhang. Automated rule selection for aspect extraction in opinion mining. In *IJCAI*, 2015.

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

[13] Y. Lu, M. Castellanos, U. Dayal, and C. Zhai. Automatic construction of a context-aware sentiment lexicon: an optimization approach. In *WWW*, 2011.

[14] D. Marcheggiani, O. Täckström, A. Esuli, and F. Sebastiani. Hierarchical multi-label conditional random fields for aspect-oriented opinion mining. In *ECIR*, 2014.

[15] M. L. McHugh. The chi-square test of independence. *Biochemia medica: Biochemia medica*, 2013.

[16] Q. Mei, X. Ling, M. Wondra, H. Su, and C. Zhai. Topic sentiment mixture: modeling facets and opinions in weblogs. In *WWW*, 2007.

[17] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013.

[18] S. Moghaddam and M. Ester. Opinion digger: an unsupervised opinion miner from unstructured product reviews. In *CIKM*, 2010.

[19] T. H. Nguyen, K. Shirai, and J. Velcin. Sentiment analysis on social media for stock movement prediction. *Expert Systems with Applications*, 2015.

[20] A.-M. Popescu and O. Etzioni. Extracting product features and opinions from reviews. In *Natural language processing and text mining*. 2007.

[21] G. Qiu, B. Liu, J. Bu, and C. Chen. Opinion word expansion and target extraction through double propagation. *Computational linguistics*, 2011.

[22] K. Schouten and F. Frasincar. Survey on aspect-level sentiment analysis. *TKDE*, 2016.

[23] J. Si, A. Mukherjee, B. Liu, Q. Li, H. Li, and X. Deng. Exploiting topic based twitter sentiment for stock prediction. In *ACL*, 2013.

[24] I. Titov and R. McDonald. A joint model of text and aspect ratings for sentiment summarization. *ACL-08: HLT*.

[25] H. Wang, Y. Lu, and C. Zhai. Latent aspect rating analysis on review text data: a rating regression approach. In *KDD*, 2010.

[26] H. Wang, Y. Lu, and C. Zhai. Latent aspect rating analysis without aspect keyword supervision. In *KDD*, 2011.

[27] L. Zhang, B. Liu, S. H. Lim, and E. O'Brien-Strain. Extracting and ranking product features in opinion documents. In *COLING*, 2010.

[28] H. Zhuang, C. Wang, F. Tao, L. Kaplan, and J. Han. Identifying semantically deviating outlier documents. In *EMNLP*, 2017.

[29] C. Zirn, M. Niepert, H. Stuckenschmidt, and M. Strube. Fine-grained sentiment analysis with structural features. In *IJCNLP*, 2011.