

## Singapore Management University Institutional Knowledge at Singapore Management University

---

Research Collection School Of Information Systems

School of Information Systems

---

5-2017

# A neural network model for semi-supervised review aspect identification

Ying DING

Singapore Management University, [ying.ding.2011@phdis.smu.edu.sg](mailto:ying.ding.2011@phdis.smu.edu.sg)

Changlong YU


Singapore Management University, [clyu@smu.edu.sg](mailto:clyu@smu.edu.sg)

Jing JIANG

Singapore Management University, [jingjiang@smu.edu.sg](mailto:jingjiang@smu.edu.sg)

**DOI:** [https://doi.org/10.1007/978-3-319-57529-2\\_52](https://doi.org/10.1007/978-3-319-57529-2_52)

Follow this and additional works at: [https://ink.library.smu.edu.sg/sis\\_research](https://ink.library.smu.edu.sg/sis_research)

 Part of the [Databases and Information Systems Commons](#), and the [Numerical Analysis and Scientific Computing Commons](#)

---

### Citation

DING, Ying; YU, Changlong; and JIANG, Jing. A neural network model for semi-supervised review aspect identification. (2017). *Advances in knowledge discovery and data mining: 21st Pacific-Asia Conference, PAKDD 2017, Jeju, South Korea, May 23-26, Proceedings*. 10235, 668-680. Research Collection School Of Information Systems.

**Available at:** [https://ink.library.smu.edu.sg/sis\\_research/3724](https://ink.library.smu.edu.sg/sis_research/3724)

This Conference Proceeding Article is brought to you for free and open access by the School of Information Systems at Institutional Knowledge at Singapore Management University. It has been accepted for inclusion in Research Collection School Of Information Systems by an authorized administrator of Institutional Knowledge at Singapore Management University. For more information, please email [libIR@smu.edu.sg](mailto:libIR@smu.edu.sg).

# A Neural Network Model for Semi-supervised Review Aspect Identification

Ying Ding<sup>(✉)</sup>, Changlong Yu, and Jing Jiang

School of Information Systems, Singapore Management University,  
Singapore, Singapore  
{ying.ding.2011,jingjiang}@smu.edu.sg, changlong.yu1@gmail.com

**Abstract.** Aspect identification is an important problem in opinion mining. It is usually solved in an unsupervised manner, and topic models have been widely used for the task. In this work, we propose a neural network model to identify aspects from reviews by learning their distributional vectors. A key difference of our neural network model from topic models is that we do not use multinomial word distributions but instead embedding vectors to generate words. Furthermore, to leverage review sentences labeled with aspect words, a sequence labeler based on Recurrent Neural Networks (RNNs) is incorporated into our neural network. The resulting model can therefore learn better aspect representations. Experimental results on two datasets from different domains show that our proposed model can outperform a few baselines in terms of aspect quality, perplexity and sentence clustering results.

## 1 Introduction

Sentiment analysis of online customer reviews has been well studied for over a decade. One of the key tasks in mining customer reviews is aspect identification [15]. Here aspects refer to features, components and other criteria on which a product or service may be evaluated by online users. Since the seminal work in [10], aspect identification has been recognized as a central problem in mining and summarizing customer reviews. Given a collection of reviews from the same domain (e.g., reviews of restaurants), aspect identification aims to discover a set of aspects, each associated with a set of aspect terms (or a distribution over such terms). For example, from restaurant reviews, we may expect to discover an aspect on service, with aspect terms such as “waiter” and “serve,” and another aspect on food, with aspect terms such as “pizza” and “burger.” The aspect identification task is useful for downstream tasks such as aspect-based review summarization [32] and product comparison [17].

Aspect identification is generally treated as an unsupervised task and a commonly adopted solution is based on topic models such as LDA (Latent Dirichlet Allocation) [1]. Here each aspect is modeled as a topic, which is essentially a multinomial distribution over words, and reviews are modeled as mixtures of these topics. A number of special topic models have been proposed for aspect identification [9, 21, 31].

With recent advances in neural networks and representation learning for natural language processing, embedding words in a low-dimensional hidden space to capture their distributional behaviors has shown to be effective for a number of data mining tasks [7, 28, 30]. In this paper, we explore how neural network models can be used to address the review aspect identification problem and whether they can outperform standard topic models. Our work is motivated by two observations: (1) Compared with the traditional multinomial word distribution based language models, neural language models constructed in a continuous space may better handle low-frequency words in reviews and address the data sparsity problem. (2) Sometimes review sentences with aspect terms annotated are available. For example, the Aspect Based Sentiment Analysis task in SemEval-2014 provides such annotated data. It has been shown that neural network models can achieve strong results on the supervised aspect term extraction task [16, 29]. We would like to explore how these trained neural network models can be used to help the aspect identification task.

In this work, we propose a neural network model for review aspect identification. Different from existing topic model based approaches to aspect identification, our model is based on continuous space language models, and it uses a small amount of labeled review sentences to train an RNN model for semi-supervised learning. Using reviews from two different domains, we show that our model improves the quality of the identified aspects compared with some baseline models, and both components of our proposed model contribute to the improved performance.

## 2 Related Work

Unsupervised topic models are one of the most popular techniques used for aspect identification. They have the advantages of requiring no supervision and being easy to extend. A model that jointly considers aspect words and sentiment words was proposed in [14]. Simple prior information based on sentiment lexicons is used in this work. Zhao et al. [31] developed a more advanced model by using a Maximum Entropy classifier to separate words belonging to different types. To further improve the performance of unsupervised topic model, some distant supervision based on domain knowledge or prior information has been incorporated [4–6]. With both users' ratings and reviews available from online review websites, aspect identification based on topic models is jointly studied with many other tasks such as rating prediction [24] and item recommendation [19, 27]. While these studies have advanced aspect identification effectively, they do not take advantage of new emerging techniques like neural networks and word embeddings.

Neural networks and word embeddings have been proven to be effective in various data mining tasks, especially supervised learning problems. They have been applied to information retrieval [23], opinion mining [8], recommender systems [11], online advertising [8] and many other various tasks. In recent years, neural networks for unsupervised learning have also been invented. Autoencoder

is one representative model among them [12,25]. However, these models lack interpretability. So neural network based topic models are proposed to overcome this shortcoming [2,13,22]. However, no one has combined supervised neural networks and unsupervised neural networks for aspect identification, which is what we study in this paper.

### 3 Method

In this section, we present our neural network model for aspect identification.

#### 3.1 Problem Formulation

The setup of our aspect identification task is as follows. We assume that we have a set of unlabeled reviews  $\mathcal{R}$  from the same domain, e.g., a set of restaurant reviews. In addition, we have a set of review sentences  $\mathcal{S}$  from the same domain annotated with aspect terms, as shown in Table 1. Our goal is to discover  $K$  aspects from  $\mathcal{R}$  and  $\mathcal{S}$ , where each aspect is associated with some parameter  $\mathbf{v}_k$  and from  $\mathbf{v}_k$  we can understand the meaning of the  $k^{\text{th}}$  aspect. In traditional topic model-based approaches to aspect discovery, each  $\mathbf{v}_k$  would be a distribution over the words in the vocabulary, and the words with the highest probabilities in  $\mathbf{v}_k$  would well represent the aspect. In our work, we do not constrain  $\mathbf{v}_k$  to be a probability distribution, as we will explain below.

**Table 1.** Examples of annotated sentences. Aspect words are highlighted and enclosed with brackets.

---

From the <b>[appetizers]</b> we ate, the <b>[dim sum]</b> and other variety of <b>[food]</b> , it was impossible to criticize.
The <b>[design]</b> and <b>[atmosphere]</b> are just so good.

---

#### 3.2 Model Overview

The general idea behind our model is as follows. We aim to re-construct the reviews in  $\mathcal{R}$  from a set of parameters capturing various properties of the reviews. To re-construct a review, we treat the review as a bag of sentences and generate the sentences one by one in a probabilistic way. Each sentence will probabilistically be assigned an aspect, and then be treated as a bag of words sharing the same aspect.

Different from standard topic models, however, we also model the context of each word using a recurrent neural network (RNN) and the context will be used to influence the probability of generating the word. Specifically, the probability of generating a word comes from a combination of a number of vectors representing different aspect models and a background model. This kind of a

mixture model is inspired by [31]. However, our model has notably the following differences from [31]: (1) Unlike [31], which is an extension of LDA, we do not use multinomial distributions to model topics (i.e., aspects in this case). Instead, we use a neural networks with continuous vectors to derive the probabilities of generating different words. This treatment is similar to a number of recent work on neural topic models [2, 22]. (2) Unlike [31], which uses a Maximum Entropy model to incorporate the context of word into its probabilistic modeling, we use an RNN to incorporate the context, which presumably is more effective given the recent success of using RNN models for sequence modeling problems.

### 3.3 Review Generation Process

**Modeling Aspects.** We assume that there are  $K$  underlying aspects. Similar to [31], which assumes that each aspect has two word distributions, namely an aspect word distribution and an opinion word distribution, we assume that each aspect  $k$  has two embedding vectors associated with it:  $\mathbf{v}_k \in \mathbb{R}^d$  and  $\mathbf{c}_k \in \mathbb{R}^d$ . Here  $\mathbf{v}_k$  is meant to capture words that directly describe the aspect, such as “pizza” and “cake” for the aspect on food or “waiter” and “waitress” for the aspect on service.  $\mathbf{c}_k$  is meant to capture other words closely associated with the aspect but are not considered opinion target terms (as those highlighted terms in Table 1). These may include “delicious” and “tasty” for the aspect on food or “friendly” for the aspect on service. Note however that neither  $\mathbf{v}_k$  nor  $\mathbf{c}_k$  is a distribution over the words in the vocabulary, and we will explain later how they are used to generate words.

**Modeling Background Words.** We assume that there is a background distribution over words, which we denote with  $\theta^b$ . This distribution represents how reviews may contain words not related to any aspect.

**Modeling Documents.** Similar to [31], we assume that each review has a multinomial distribution over the  $K$  aspects. Let us use  $\beta_r$  to represent this distribution for the  $r^{\text{th}}$  review. We also assume that there is a document-independent probability  $\lambda$  that controls how likely a word is associated with an aspect or with the background model  $\theta^b$ .

**Modeling Word Context.** We use  $w_{r,s,n}$  to represent the  $n^{\text{th}}$  word in the  $s^{\text{th}}$  sentence in the  $r^{\text{th}}$  review. Here  $1 \leq w_{r,s,n} \leq V$  is an index in the vocabulary and  $V$  is the vocabulary size. We assume that this word has a vector  $\mathbf{h}_{r,s,n}$  that encodes its context using an RNN model we will describe later. With this vector  $\mathbf{h}_{r,s,n}$  and the RNN model, there is a probability  $\pi_{r,s,n}$  associated with word  $w_{r,s,n}$  to indicate how likely this word is an opinion target term rather than an opinion term, i.e., how likely  $w_{r,s,n}$  is going to be generated from some  $\mathbf{v}_k$  or from some  $\mathbf{c}_k$ .

**Review Generation.** With the various embedding vectors and probabilities defined above, we now describe the re-construction loss function which we try to minimize in order to learn the parameters. We use the negative log likelihood of generating the words inside all the reviews in  $\mathcal{R}$  as our objective function. The overall objective function is as follows:

$$\begin{aligned}
 -\log p(\mathcal{R}) &= -\sum_{r=1}^{|\mathcal{R}|} \log p(\mathbf{w}_r) = -\sum_{r=1}^{|\mathcal{R}|} \sum_{s=1}^{M_r} \log \sum_{k=1}^K \beta_{r,k} p(\mathbf{w}_{r,s}|k), \\
 p(\mathbf{w}_{r,s}|k) &= \prod_{n=1}^{N_{r,s}} p(w_{r,s,n}|k) \\
 &= \prod_{n=1}^{N_{r,s}} \left[ (1-\lambda)\theta_{w_{r,s,n}}^b + \lambda \left( \pi_{r,s,n} \phi_{k,w_{r,s,n}} + (1-\pi_{r,s,n}) \psi_{k,w_{r,s,n}} \right) \right],
 \end{aligned}$$

where  $M_r$  is the number of sentences in the  $r^{\text{th}}$  review,  $N_{r,s}$  is the number of words in the  $s^{\text{th}}$  sentence in the  $r^{\text{th}}$  review,  $\mathbf{w}_r$  represents all the words in the  $r^{\text{th}}$  review,  $\mathbf{w}_{r,s}$  represents all the words in the  $s^{\text{th}}$  sentence in the  $r^{\text{th}}$  review, and  $\phi_k$  and  $\psi_k$  are two distributions corresponding to aspect terms and opinion terms, which we will explain below.

Basically the loss function above shows that to generate a review  $r$ , for each sentence in the review we pick an aspect  $k$  according to the distribution  $\beta_r$ . Then for each word in this sentence, we generate it either from the background model  $\theta_b$  or one of the two models  $\phi_k$  and  $\psi_k$ .

So far the model above is very similar to [31]. However,  $\phi_k$  and  $\psi_k$  are modeled differently from [31]. Instead of treating these as multinomial distributions and directly learning the probabilities, we assume that they are derived from the embedding vectors  $\mathbf{v}_k$  and  $\mathbf{c}_k$  as follows:

$$\begin{aligned}
 \phi_k &= \text{softmax}(\mathbf{v}_k \cdot \mathbf{W}_A), \\
 \psi_k &= \text{softmax}(\mathbf{c}_k \cdot \mathbf{W}_C).
 \end{aligned}$$

$\mathbf{W}_A \in \mathbb{R}^{d \times V}$  and  $\mathbf{W}_C \in \mathbb{R}^{d \times V}$  are two matrices to model the semantic representations of words, which are initialized with pre-trained Google word2vec.<sup>1</sup> Each column in them is used to encode one word type.

### 3.4 RNN to Incorporate Context

We now explain how we obtain  $\pi_{r,s,n}$  for each word  $w_{r,s,n}$  by making use of the annotated review sentences. Our method is again inspired by the MaxEnt-LDA model [31], in which a Maximum Entropy model was trained on some labeled data to help separate aspect words, opinion words and background words.

<sup>1</sup> <https://code.google.com/archive/p/word2vec/>.

---

**Algorithm 1.** Gibbs-EM algorithm for learning

---

```
1: for  $i \leftarrow 1, \text{maxEpoch}$  do  $\triangleright$   $\text{maxEpoch}$  is the maximum number of epochs.
2:   E-step:
3:   for  $r \leftarrow 1, |\mathcal{R}|$  do
4:     for  $s \leftarrow 1, M_r$  do
5:       Sample an aspect  $t_{r,s}^i$  according to Formula 1.
6:     end for
7:   end for
8:   M-step:
9:   Keep  $\mathbf{T}_i$  fixed. Compute the gradient  $\frac{\partial \mathcal{L}_i}{\partial \Theta}$  by back-propagation.
10:  Use the gradient to update all parameters  $\Theta$ .
11: end for
```

---

The same idea applies to our problem, but here we use a Recurrent Neural Network (RNN) model, which represents the state of the art for aspect term extraction [16].

The motivation of making use of the labeled review sentences is that there are some patterns we can learn to locate aspect terms. For example, nouns following adjectives which are sentiment words, such as the word “service” in the phrase “excellent service,” are more likely to be aspect terms. We can try to learn such patterns from the labeled review sentences, even though the labels only indicate which words are aspect terms but do not group them into aspects.

Because usually there is only a small amount of such labeled review sentences, to address the data sparsity problem, here we again make use of dense vector representations to train a classifier. Specifically, we use Recurrent Neural Network (RNN) models. Let us assume that  $(\mathbf{l}_1, \mathbf{l}_2, \dots, \mathbf{l}_n)$  is the sequence of words in a labeled sentence, where each  $\mathbf{l}_i \in \mathbb{R}^d$  is a dense word embedding vector. Let  $(y_1, y_2, \dots, y_n)$  represent the corresponding labels marking the positions of the aspect terms. We can build an RNN model from the sequence  $(\mathbf{l}_1, \mathbf{l}_2, \dots, \mathbf{l}_n)$  as follows:

$$\mathbf{h}_i = f(\mathbf{U}\mathbf{h}_{i-1} + \mathbf{V}\mathbf{l}_i + \mathbf{e}),$$

where  $f(\cdot)$  is a non-linear activation function,  $\mathbf{U} \in \mathbb{R}^{d_o \times d_o}$ ,  $\mathbf{V} \in \mathbb{R}^{d_o \times d}$  and  $\mathbf{e} \in \mathbb{R}^{d_o}$  are parameters to be learned,  $d_o$  is the output dimension and  $\mathbf{h}_i$  is the hidden state at position  $i$ . We can then use  $\mathbf{h}_i$  to predict the label  $y_i$  through a softmax layer. While there exist some other RNN structures like LSTM(Long Short Term Memory), Bidirectional-RNN, Bidirectional-LSTM and so on, RNN has simpler structure and competitive performance [16]. So we only use RNN to predict  $\pi_{r,s,n}$  in this work.

To train this model, we maximize the probabilities of the observed labels in the training dataset  $\mathcal{S}$ . Given a new sentence, we can use the trained RNN model to obtain the hidden states  $\mathbf{h}$ , and for each word in the sentence, we can use its corresponding hidden state to obtain a probability  $\pi_{r,s,n}$  for the word to be an aspect term.

### 3.5 Connections with Topic Models

With certain configurations, our model is closely connected with traditional topic models. However, our model learns aspect vectors and uses a linear transformation followed by the softmax function to model topic-word dependencies. Compared with multinomial distributions, which are typically used in topic models, our model can incorporate more information, like semantic meanings of words and topics. In recent years, neural network based topic models have been invented to incorporate pre-trained word embeddings [2, 13, 22]. Compared with these models, our model is a more general framework. Each component of it can be replaced with other suitable options. So it is easier to extend and adapt to different tasks. Besides this, we use RNN to separate aspect words from context words, which can potentially help us learn better topics. This has not been used in existing neural topic models.

### 3.6 Learning

To learn our model, we need to find the optimal values of  $\mathbf{v}_k$ ,  $\mathbf{c}_k$ ,  $\boldsymbol{\theta}^b$ ,  $\boldsymbol{\beta}_r$ ,  $\mathbf{W}_A$ ,  $\mathbf{W}_c$  and  $\lambda$  that can minimize the objective function  $-\log p(\mathcal{R})$ .

Back-propagations cannot be directly used to learn our neural network as there are some constraints placed on  $\mathbf{h}_d$ . To deal with this, one alternative is variational-EM algorithm. However, it is not an exact estimation algorithm as it tries to optimize the lower bound of the objective function. Instead of using variational inference to approximate posterior distributions at the E-step, we adopt Gibbs sampling to sample an aspect for the  $s^{\text{th}}$  sentence in the  $r^{\text{th}}$  review according to

$$p(t_{r,s} = k) = \frac{\beta_{r,k} p(\mathbf{w}_{r,s} | k)}{\sum_{k'} \beta_{r,k'} p(\mathbf{w}_{r,s} | k')}. \quad (1)$$

Then, in the M-step, we apply back-propagation to update all parameters in our neural network with the sampled aspect for sentence fixed. The objective function for the M-step in the  $i$ th epoch is

$$\mathcal{L}_i = -\log p(\mathcal{R} | \mathbf{T}_i) = -\sum_{r=1}^{|\mathcal{R}|} \sum_{s=1}^{M_r} \log p(\mathbf{w}_{r,s} | t_{r,s}^i), \quad (2)$$

where  $\mathbf{T}_i$  is the sampled aspects of all sentences in epoch  $i$  and  $t_{r,s}^i$  is the sampled aspect in epoch  $i$  for the  $s^{\text{th}}$  sentence in the  $r^{\text{th}}$  review. An overview of the learning process can be found in Algorithm 1, where  $\Theta$  represents all parameters to be learned:  $\Theta = \{\mathbf{v}_k, \mathbf{c}_k, \boldsymbol{\theta}^b, \boldsymbol{\beta}_r, \mathbf{W}_A, \mathbf{W}_C, \lambda\}$ ,  $k \in \{1, 2, \dots, K\}$ ,  $r \in \{1, 2, \dots, |\mathcal{R}|\}$ .

## 4 Experiments

In this section, we evaluate our proposed model from different angles. Through the evaluation we mainly want to test if our neural network model using aspect



and context vectors to generate words work better than traditional topic models based on multinomial unigram word distributions for aspect identification. In addition, we also look at the generative ability and the effectiveness of clustering sentences using our model.

We consider the following different models for comparison.

- **LDA:** Latent Dirichlet Allocation. This is a classical topic modeling technique proposed in [1].
- **JST:** Joint Sentiment/Topic Model. It is an extension of LDA that models both sentiments and topics [14].
- **ME-LDA:** LDA with Maximum Entropy classifier [31]. This models uses both traditional topic models based on multinomial unigram word distributions and Maximum Entropy models for supervision.
- **RNN-LDA:** LDA with RNN.

We replace the maximum entropy classifier in ME-LDA with the trained RNN model to estimate the probability of each word being an aspect word or not. By comparing with this model, we can evaluate the effect of using aspect and context vectors together with softmax to generate words.

- **ME-NA:** Neural network for aspect identification with Maximum Entropy. This is a variation of our model. We replace LDA in ME-LDA with our neural network model.

By comparing with this model, we can evaluate the usefulness of using RNN instead of standard linear classifiers for the supervision.

- **RNN-NA:** Neural network for aspect identification with RNN. This is our complete model as presented in Sect. 3, where we use both unlabeled and labeled data for aspect identification. We do not fine tune  $\mathbf{W}_A$  and  $\mathbf{W}_C$ , i.e., the word embeddings are not updated during training.
- **RNN-NA-t:** This is also our complete model RNN-NA. However, we initialize  $\mathbf{W}_A$  and  $\mathbf{W}_C$  with word embeddings and fine-tune them during training.

To compare the models above, we first conduct three experiments to evaluate the quality of identified aspects. Then we do a quantitative evaluation based on perplexity to check the model’s ability to predict words in unseen reviews. We also do another quantitative evaluation using sentence clustering to evaluate each model’s effectiveness in grouping review sentences into different aspects.

## 4.1 Data

We use two datasets for our experiments. The first one contains restaurant reviews from the Yelp academic dataset.<sup>2</sup> As the original dataset contains millions of reviews from different businesses, we only keep the restaurant reviews and randomly sample 20,000 from them. The other dataset is a laptop dataset crawled from Amazon, used by [26].<sup>3</sup> For the set of labeled training sentences, we use the sentences tagged with aspect terms from SemEval competitions.

---

<sup>2</sup> [https://www.yelp.com.sg/dataset\\_challenge](https://www.yelp.com.sg/dataset_challenge).

<sup>3</sup> <http://www.cs.virginia.edu/~hw5x/dataset.html>.

For the restaurant domain, the training sentences are from SemEval 2014 and 2015, and for the laptop domain, the training sentences are from SemEval 2015.

To pre-process the review data, we remove stop words and words with no pre-trained embeddings. Sentences with less than 3 words are also removed. After preprocessing, the Yelp dataset contains 17948 reviews, with each document containing 9.1 sentences on average and each sentence containing 5.8 words on average. In the Laptop dataset, there are 31,363 documents, where each document has 8.8 sentences on average and each sentence has 7.6 words on average.

## 4.2 Aspect Quality

**Word Intrusion.** To evaluate the quality of aspects identified by our models, we conduct the word intrusion experiment [3]. For each discovered aspect, we extract 5 most probable words. We also extract another intrusion word that has a high probability in some other aspect but low probability in the current aspect. There words are then mixed and presented to the annotators to pick out the intrusion word. We ask four graduate students for the annotation. Fleiss’ Kappa, which is a standard way to measure agreement among more than two annotators, shows that the inter-annotator agreement is 0.353 for the Yelp dataset and 0.487 for the Laptop dataset. These two scores indicate fair agreement and moderate agreement respectively. Model Precision ( $MP$ ) is used as the evaluation metric, which is defined as

$$MP = \frac{1}{N} \sum_{a=1}^N \frac{M_a}{T}.$$

Here,  $N$  is the number of annotators,  $T$  is the number of aspects,  $M_a$  is the number of intrusion words that are correctly identified by annotator  $a$ .

The performances of all models with aspect number set to be 10 and 20 are shown in Table 2. We can see that RNN-NA-t performs the best most of the time, which demonstrates that our model is effective in mining aspects with high quality. RNN-NA can only outperform RNN-NA-t in one case. It proves that fine-tuning word embeddings in our model is important.

**Coherence.** Besides human evaluation, we also evaluated our models with topic coherence, which is a metric measuring aspect quality based on co-occurrence of words [20]. It is defined as

**Table 2.** Model precision ( $MP$ ) of word intrusion by various models.

Dataset	#Aspect	JST	LDA	ME-LDA	RNN-LDA	ME-NA	RNN-NA	RNN-NA-t
Yelp	10	0.63	0.50	0.65	0.45	0.50	0.53	<b>0.65</b>
	20	0.44	0.45	0.51	0.40	0.50	<b>0.63</b>	0.55
Laptop	10	0.40	0.33	0.50	0.70	0.70	0.58	<b>0.73</b>
	20	0.64	0.44	0.59	0.74	0.65	0.55	<b>0.75</b>

**Table 3.** Topic coherence.

Dataset	#Aspect	JST	LDA	ME-LDA	RNN-LDA	RNN-NA	ME-NA	RNN-NA-t
Yelp	10	-3.589	-2.854	-4.421	-4.110	-0.757	-0.639	<b>-0.363</b>
	20	-3.579	-2.833	-4.319	-4.129	-0.698	-0.628	<b>-0.443</b>
Laptop	10	-3.218	-3.424	-5.476	-5.591	-1.090	-1.077	<b>-0.866</b>
	20	-3.236	-3.459	-5.514	-5.698	-1.186	-1.111	<b>-0.787</b>

$$C(t, V^{(t)}) = \frac{2}{M(M+1)} \sum_{m=2}^M \sum_{l=1}^{m-1} \log \frac{D(v_m^{(t)}, v_l^{(t)}) + 1}{D(v_l^{(t)})},$$

where  $V^{(t)}$  contains the  $M$  most probable words in topic  $t$ .  $v_m^{(t)}$  and  $v_l^{(t)}$  are the  $m$ th and  $l$ th words in  $V^{(t)}$ .  $D(v_l^{(t)})$  is the number of documents containing word  $v_l^{(t)}$  and  $D(v_m^{(t)}, v_l^{(t)})$  is the number of documents containing both  $v_m^{(t)}$  and  $v_l^{(t)}$ .

Table 3 displays the averaged topic coherence of different models. All models based on our proposed neural network can get better performance than others. Meanwhile, RNN-NA-t consistently gets the best performance. It proves that aspects discovered by our models are more coherent than those discovered by the competitors.

**Qualitative Evaluation.** To qualitatively study the quality of aspects identified by our proposed model, we show 4 sample aspects of the laptop dataset identified by RNN-NA-t and ME-LDA in Table 4. The top 10 most probable words of each aspect are displayed. Words that are closely related to the aspect are emphasized in bold font. From the tables we can see that aspects learned by RNN-NA-t look more coherent and more words are closely related to the topic. The qualitative evaluation shows the advantage of our neural network for aspect identification in discovering meaningful and coherent aspects.

**Table 4.** Sampled learned aspects from the Laptop dataset.

RNN-NA-t				ME-LDA			
Network	Display	OS	Support	Network	Display	OS	Support
<b>Wifi</b>	<b>Screen</b>	<b>Windows</b>	<b>Support</b>	Windows	<b>Screen</b>	<b>Windows</b>	<b>Warranty</b>
<b>Wireless</b>	<b>Display</b>	<b>OS</b>	<b>Service</b>	Screen	Keyboard	<b>System</b>	<b>Service</b>
<b>Connection</b>	<b>Resolution</b>	<b>System</b>	<b>Customer</b>	Support	Windows	<b>OS</b>	<b>Customer</b>
<b>Internet</b>	Keyboard	<b>Operating</b>	<b>Warranty</b>	<b>Wireless</b>	Battery	Screen	<b>Support</b>
Windows	<b>Color</b>	Software	<b>Tech</b>	<b>Wifi</b>	Quality	<b>Operating</b>	Drive
Driver	<b>Size</b>	<b>XP</b>	<b>Shipping</b>	<b>Connection</b>	<b>Display</b>	Software	Screen
<b>Card</b>	Quality	<b>Vista</b>	Samsung	System	Sound	Use	Hard
<b>Network</b>	<b>Colors</b>	Use	Screen	<b>Internet</b>	Price	Keyboard	Windows
Drivers	<b>Brightness</b>	Works	Battery	Battery	Touch	Drive	Battery
Support	<b>Retina</b>	Hardware	System	Keyboard	Drive	Battery	<b>Shipping</b>

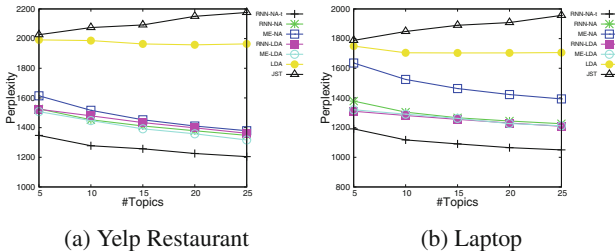


Fig. 1. Perplexities over different numbers of aspects for different models.

### 4.3 Perplexity

We evaluate all models’ generative abilities using perplexity, which is a commonly used metric to evaluate the quality of language models and topic models. The definition of perplexity is as follows:

$$\text{perplexity} = \exp\left(-\frac{1}{N} \sum_{s \in \mathcal{T}} P(s)\right), \quad (3)$$

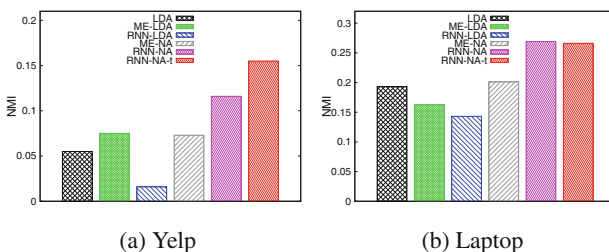
where  $\mathcal{T}$  is our held-out test dataset,  $N$  is the total number of sentences in it and  $P(s)$  is the probability of generating sentence  $s$ . In our experiment, we leave 20% of our dataset for testing and train the models based on the remaining 80% dataset. Perplexities over different numbers of aspects are shown in Fig. 1.

We can see that our complete model with fine tuning of word embeddings is performing the best over various numbers of aspects on both datasets. Meanwhile, using RNN models to help separate aspect words from the rest performs better than using Maximum Entropy based models most of the time. Both findings verify that using neural networks in our model can improve generalization capabilities.

### Sentence Clustering

To show how topical embeddings learned by different models benefit downstream tasks, we compare the different models in terms of sentence clustering. We manually labeled 100 sentences from the Yelp dataset and 100 sentences from the Laptop dataset. Normalized mutual information [18], which is a popular metric in text clustering, is used to measure performances in our experiment. As topics discovered by JST are sentiment oriented, we do not include it in this evaluation.

The results are shown in Fig. 2. We can see that our proposed neural network models outperform all other competitors. As all sentences are from the same domain, it is uneasy to effectively discover clear aspects and cluster sentences by using co-occurrence statistics. So traditional topic models perform poorly. By learning topic embeddings, our models can improve a lot. Figure 2 also shows that using RNN to help separate out aspect words is much more effective than Maximum Entropy classifier.



**Fig. 2.** Normalized mutual information.

## 5 Conclusions

We explored aspect identification from reviews by proposing a novel neural network model. Our model is able to associate aspects and words using distributional vectors. An RNN model trained on labeled sentences is embedded into our model, which helped the model learn cleaner and more discriminative topics. Experiments on two datasets from different domains show that our model is effective in discovering meaningful aspects, predicting words and benefiting downstream applications such as sentence clustering. In the future, we will explore more complex neural network layers to model aspects and documents, and to jointly train the RNN with the neural network model for aspect identification.

**Acknowledgement.** This research is supported by the National Research Foundation, Prime Ministers Office, Singapore under its International Research Centres in Singapore Funding Initiative.

## References

1. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. *JMLR* **3**, 993–1022 (2003)
2. Cao, Z., Li, S., Liu, Y., Li, W., Ji, H.: A novel neural topic model and its supervised extension. In: *AAAI*, pp. 2210–2216 (2015)
3. Chang, J., Boyd-Graber, J.L., Gerrish, S., Wang, C., Blei, D.M.: Reading tea leaves: how humans interpret topic models. In: *NIPS*, pp. 288–296 (2009)
4. Chen, Z., Mukherjee, A., Liu, B.: Aspect extraction with automated prior knowledge learning. In: *ACL*, pp. 347–358 (2014)
5. Chen, Z., Mukherjee, A., Liu, B., Hsu, M., Castellanos, M., Ghosh, R.: Discovering coherent topics using general knowledge. In: *CIKM*, pp. 209–218 (2013)
6. Chen, Z., Mukherjee, A., Liu, B., Hsu, M., Castellanos, M., Ghosh, R.: Exploiting domain knowledge in aspect extraction. In: *EMNLP*, pp. 1655–1667 (2013)
7. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.P.: Natural language processing (almost) from scratch. *JMLR* **12**, 2493–2537 (2011)
8. Du, H., Xu, X., Cheng, X., Wu, D., Liu, Y., Yu, Z.: Aspect-specific sentimental word embedding for sentiment analysis of online reviews. In: *WWW*, pp. 29–30 (2016)

9. Fei, G., Chen, Z., Liu, B.: Review topic discovery with phrases using the Pólya urn model. In: COLING, pp. 667–676 (2014)
10. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: KDD, pp. 168–177 (2004)
11. Kim, Y.: Convolutional neural networks for sentence classification. In: EMNLP, pp. 1746–1751 (2014)
12. Li, J., Luong, M., Jurafsky, D.: A hierarchical neural autoencoder for paragraphs and documents. In: ACL, pp. 1106–1115 (2015)
13. Li, S., Zhu, J., Miao, C.: A generative word embedding model and its low rank positive semidefinite solution. In: ACL (2016)
14. Lin, C., He, Y.: Joint sentiment/topic model for sentiment analysis. In: CIKM, pp. 375–384 (2009)
15. Liu, B.: Sentiment analysis and opinion mining. Synth. Lect. Hum. Lang. Technol. **5**(1), 1–167 (2012)
16. Liu, P., Joty, S., Meng, H.: Fine-grained opinion mining with recurrent neural networks and word embeddings. In: EMNLP, pp. 1433–1443 (2015)
17. Lu, Y., Zhai, C., Sundaresan, N.: Rated aspect summarization of short comments. In: WWW, pp. 131–140 (2009)
18. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, New York (2008)
19. McAuley, J., Leskovec, J.: Hidden factors and hidden topics: understanding rating dimensions with review text. In: RecSys, pp. 165–172 (2013)
20. Mimno, D., Wallach, H.M., Talley, E., Leenders, M., McCallum, A.: Optimizing semantic coherence in topic models. In: EMNLP, pp. 262–272 (2011)
21. Mukherjee, A., Liu, B.: Aspect extraction through semi-supervised modeling. In: ACL, pp. 339–348 (2012)
22. Nguyen, D.Q., Billingsley, R., Du, L., Johnson, M.: Improving topic models with latent feature word representations. TACL **3**, 299–313 (2015)
23. Severyn, A., Moschitti, A.: Learning to rank short text pairs with convolutional deep neural networks. In: SIGIR, pp. 373–382 (2015)
24. Wang, H., Ester, M.: A sentiment-aligned topic model for product aspect rating prediction. In: EMNLP, pp. 1192–1202 (2014)
25. Wang, H., Wang, N., Yeung, D.Y.: Collaborative deep learning for recommender systems. In: KDD, pp. 1235–1244 (2015)
26. Wang, H., Lu, Y., Zhai, C.: Latent aspect rating analysis without aspect keyword supervision. In: KDD, pp. 618–626 (2011)
27. Wu, Y., Ester, M.: FLAME: a probabilistic model combining aspect based opinion mining and collaborative filtering. In: WSDM, pp. 199–208 (2015)
28. Zhai, S., Chang, K., Zhang, R., Zhang, Z.M.: Deepintent: learning attentions for online advertising with recurrent neural networks. In: KDD, pp. 1295–1304 (2016)
29. Zhang, M., Zhang, Y., Vo, D.: Neural networks for open domain targeted sentiment. In: EMNLP, pp. 612–621 (2015)
30. Zhang, Q., Gong, Y., Wu, J., Huang, H., Huang, X.: Retweet prediction with attention-based deep neural network. In: CIKM, pp. 75–84 (2016)
31. Zhao, W.X., Jiang, J., Yan, H., Li, X.: Jointly modeling aspects and opinions with a MaxEnt-LDA hybrid. In: EMNLP, pp. 56–65 (2010)
32. Zhu, J., Wang, H., Zhu, M., Tsou, B.K., Ma, M.: Aspect-based opinion polling from customer reviews. IEEE Trans. Affect. Comput. **2**(1), 37–49 (2011)