Aspect-level Sentiment Analysis using AS-Capsules

Yequan Wang[1,*] Aixin Sun[2] Minlie Huang[1] Xiaoyan Zhu[1]
[1]Institute for Artificial Intelligence, State Key Lab of Intelligent Technology and Systems
[1]Beijing National Research Center for Information Science and Technology
[1]Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
[2]School of Computer Science and Engineering, Nanyang Technological University, Singapore
tshwangyequan@gmail.com;axsun@ntu.edu.sg;aihuang@tsinghua.edu.cn;zxy-dcs@tsinghua.edu.cn

ABSTRACT

Aspect-level sentiment analysis aims to provide complete and detailed view of sentiment analysis from different aspects. Existing solutions usually adopt a two-staged approach: first detecting aspect category in a document, then categorizing the polarity of opinion expressions for detected aspect(s). Inevitably, such methods lead to error accumulation. Moreover, aspect detection and aspect-level sentiment classification are highly correlated with each other. The key issue here is how to perform aspect detection and aspect-level sentiment classification jointly, and effectively. In this paper, we propose the *aspect-level sentiment capsules* model (**AS-Capsules**), which is capable of performing aspect detection and sentiment classification simultaneously, in a joint manner. AS-Capsules utilizes the correlation between aspect and sentiment through shared components including capsule embedding, shared encoders, and shared attentions. AS-Capsules is also capable of communicating with different capsules through a shared Recurrent Neural Network (RNN). More importantly, AS-Capsules model does not require any linguistic knowledge as additional input. Instead, through the attention mechanism, this model is able to attend aspect related words and sentiment words corresponding to different aspect(s). Experiments show that the AS-Capsules model achieves state-of-the-art performances on a benchmark dataset for aspect-level sentiment analysis.

ACM Reference Format:
Yequan Wang[1,*] Aixin Sun[2] Minlie Huang[1] Xiaoyan Zhu[1]. 2019. Aspect-level Sentiment Analysis using AS-Capsules. In *Proceedings of the 2019 World Wide Web Conference (WWW '19), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3308558.3313750

1 INTRODUCTION

Sentiment analysis aims to analyze people's sentiments, opinions, evaluations, attitudes, and emotions from human languages [17, 24]. Current researches focus on document-level (*e.g.,* document, paragraph, and sentence), or in-depth aspect-level analysis. As a fine-grained task, aspect-level sentiment analysis provides complete and detailed view of sentiments from different aspects. In general, this task requires aspect detection and aspect-level sentiment classification. Most existing solutions first detect aspect category in a sentence, and then categorize the polarity of opinion expressions with respect to the detected aspect(s). In other words, the two subtasks are tackled separately. As a result, errors made in aspect detection would affect aspect-level sentiment classification. On the other hand, the two subtasks, *i.e.,* aspect detection and aspect-level sentiment classification, are highly correlated with each other.

In this paper, we propose the *aspect-level sentiment capsules* (**AS-Capsules**) model. This model utilizes the correlation between aspects and corresponding sentiments. Hence, we jointly perform the two subtasks: aspect detection and aspect-level sentiment classification.

The Research Problem. In aspect-level sentiment analysis, we have a predefined set of aspects $\mathcal{A} = \{a_1, a_2, \cdots, a_M\}$, and a predefined set of sentiment polarities $\mathcal{P} = \{o_1, o_2, \cdots, o_P\}$. Given a piece of text (*e.g.,* a sentence or a paragraph), denoted by $S = [w_1, w_2, \ldots, w_N]$, the task is to predict the aspect(s) and the corresponding sentiment(s), *i.e.,* aspect-sentiment pairs $\{\langle a_i, o_i\rangle\}$, expressed in the text.

Consider the task is to be conducted on restaurant reviews. The set of aspects $\mathcal{A}$ can be {*food, service, price, ambience, anecdote*}, and the set of sentiment polarities $\mathcal{P}$ may include {*positive, neutral, negative*}. Each restaurant review is considered a piece of text S. Given an example review "Staffs are not that friendly, but the taste covers all.", the expected output will be aspect-sentiment pairs $\{\langle\text{food, positive}\rangle, \langle\text{service, negative}\rangle\}$. Accordingly, such pairs are available for sample inputs as training data, where a supervised sentiment detection algorithm could learn from.

Recently, sentiment analysis has attracted wide attention. Aspect detection has been studied as a subtask of aspect-level sentiment analysis. The goal of aspect detection is to identify the aspect categories in sentence instead of extracting aspect terms. The aspect category set is predefined in advance. Most existing researches focus on classical machine learning models using classifiers with rich features, or deep learning models. LR and SVM are among the popular and effective classifiers. Unigram, Bigram and Lexicon features are the most important features for aspect detection. Many neural network models have been proposed for various tasks including sentiment analysis, such as Recursive Auto Encoder (RAE)[36, 37], Recurrent Neural Network (RNN) [19, 38], Convolutional Neural Network (CNN) [13, 14, 16], and more. The common solutions typically train deep learning models by using the basic models, and fine-tuning the word representations pre-trained by word2vec [20] or glove [25]. Attention-based LSTM with

[*]This work was done when Yequan was a visiting Ph.D student at School of Computer Science and Engineering, Nanyang Technological University, Singapore.

WWW '19, May 13–17, 2019, San Francisco, CA, USA

ACM ISBN 978-1-4503-6674-8/19/05.
https://doi.org/10.1145/3308558.3313750

aspect embedding (ATAE-LSTM) [44] is shown effective to enforce the neural model to attend the related part of a sentence, with response to a specific aspect. Some variants of RNN and attention are proposed to improve the performance of aspect-level sentiment classification. However, ATAE-LSTM and its variants need aspect category as input, which limits the application. As we mentioned before, how to take advantage of the relationship between aspect detection and sentiment classification is the key.

In spite of the great success of neural network models, linguistic knowledge [30] is often required to achieve the best performance for sentiment analysis. However, linguistic knowledge is domain specific and costly to obtain. For example, the words to express positive and negative opinions in restaurant reviews will be very different from the words used in movie reviews. Further, many neural network models cannot provide explanations on how and why the predictions are made. Very recently, RNN-Capsule [45] demonstrates state-of-the-art accuracy on sentence-level sentiment classification. It does not require linguistic knowledge and is capable of outputting meaningful words with sentiment tendency. Inspired by RNN-Capsule, we propose AS-Capsules model for aspect-level sentiment analysis.

The AS-Capsules Model. The concept of "capsule" was proposed by Hinton *et al.* in 2011 [8]. A capsule is a group of neurons that "perform some quite complicated internal computations on their inputs and then encapsulate the results of these computations into a small vector of highly informative outputs" [8]. Following this high-level concept, each capsule in RNN-Capsule was designed to predict one sentiment polarity (*e.g.,* positive, negative, and neutral) [45]. In this work, we follow the same high-level concept of capsule as in RNN-Capsule, to design the AS-Capsules model for aspect-level sentiment analysis.[1]

Specifically, each individual capsule in the AS-Capsules model contains *an attribute*, *a state*, *a capsule embedding*, and *four modules*. The four modules are 'aspect representation module', 'aspect probability module', 'sentiment representation module', and 'sentiment distribution module'. For each predefined aspect, we build a capsule whose attribute is the same as the aspect category (*e.g., food* or *price*). Given a piece of text, we represent the given text by the low-level hidden vectors encoded by an encoder RNN. All capsules take the low-level hidden representations as their input, and each capsule outputs: (i) the aspect probability computed by its aspect probability module, and (ii) the sentiment distribution computed by its sentiment distribution module. All capsules utilize a shared RNN to communicate with each other to prevent capsules from attending conflict parts. The hidden representation of shared RNN is high-level because its input is the output of low-level encoder RNN. All attentions in capsule rely on the capsule embedding, which is capable of enforcing the model to focus on the correlated parts with respect to aspect. Aspect representation module and sentiment representation module share one representation generated by a component known as shared attention.

Compared with most existing neural network models for sentiment analysis, the AS-Capsules model does not heavily rely on the quality of input instance representation. The RNN layer to encode the given text input can be realized through the widely used LSTM models, GRU models or their variants. The model does not require any linguistic knowledge. Instead, each capsule is capable of outputting two kinds of attended words, one kind of words to reflect its assigned aspect category, and the other to reflect the sentiment tendency. Both sets of words are learned through the attention mechanisms. Experiments show that the words attended by each capsule well reflect the capsule's aspect category and sentiment tendency. We observe that the attended words cover a wide range of words from high frequency words to low frequency words. As low frequency words are not usually covered in sentiment lexicon, the domain-dependent aspect and sentiment words could be extremely useful sense making from the feedbacks to services or products. To summarize, the main contributions of this work are as follows:

- We propose AS-Capsules model to simultaneously perform aspect detection and aspect-level sentiment classification. A capsule is easy to build with input representations encoded by RNN. Each capsule contains an attribute, a state, a capsule embedding, and four modules known as aspect representation module, aspect probability module, sentiment representation module, and sentiment distribution module.
- The proposed AS-Capsules model does not require any linguistic knowledge to achieve state-of-the-art performance. Instead, the model is able to attend both sentiment words and aspect words reflecting the aspect knowledge of the dataset.
- We conduct experiments on a benchmark dataset (SemEval 2014 Task 4 dataset) to compare our model with strong baselines. Results show that our model is competitive and robust. We further show that our model trained on the SemEval dataset could be directly applied on Yelp reviews and output meaningful results.

2 RELATED WORK

Early approaches for sentiment analysis are mostly based on feature engineering and manually defined rules [35]. Recently, neural networks become the mainstream for sentiment analysis. Most current studies focus on improving the quality of vector representation of input instance using different models *e.g.,* RNN, RAE, CNN. We briefly review the related works on aspect detection, and aspect-level sentiment classification.

Aspect Detection. Aspect detection aims at identifying aspects about which users express their sentiments. A popular approach for aspect detection features a frequency-based method [10], where single nouns and compound nouns are considered possible aspects. Hence, only explicit aspects are detected. The authors in [6] then employ association rules mining to find implicit aspects. Instead of focusing on frequencies, syntax-based methods have also been used to detect aspects by means of syntactical relations [48, 49]. In general, this kind of models operates in an unsupervised manner. In [29], the authors propose a hybrid model where pointwise mutual information is used to find possible aspects, which are then fed into a Naive Bayes classifier to output a set of explicit aspects. There are also works like [11] to formulate aspect detection as a labeling problem, and a linear chain Conditional Random Field (CRF) is utilized.

[1] Our AS-Capsules model is different from the idea of Capsule Network (CapsNet) [34].

Recently, relevant aspects are identified by employing word embedding techniques [32]. The method utilizes semantic and syntactic relationships in word embedding vectors in order to improve the extraction of multiple words aspects and distinguish conflict aspects. The effectiveness of word embeddings is investigated for aspect-level sentiment analysis in [1], in which both semantic and sentiment information are encoded. A semi-supervised word embedding algorithm is proposed in [50] to obtain continuous word embeddings on a large set of reviews. Then the word representations could be used to generate deeper and hybrid features to predict the aspect category.

To improve the performance of aspect detection, additional convolutional neural network features are extended in [42] besides extracting lots of features including lexicon, syntactic and cluster. Nevertheless expensive human effort and CNN operations limit its application.

Aspect-level Sentiment Analysis. Aspect-level sentiment classification deals with fine-grained classification with respect to specific aspect(s). Traditional approaches are to design a set of features manually. There are lexicon-based features built for sentiment analysis [22] with the abundance of sentiment lexicons [12, 26, 31]. Many studies focus on building sentiment classifiers with bag-of-words, sentiment lexicons, and other features, using SVM [23] or other classifiers. However, the results highly depend on the quality of features, and feature engineering is expensive.

Recently, many neural network models have been developed to tackle sentiment analysis. Transferring knowledge from existing public datasets [7] or pre-annotated information [39] improves the performance of aspect-term level sentiment classification. There are methods using memory network or linguistic knowledge to improve the performance of aspect-level sentiment classification. An innovative model named CEA is proposed in [46] using context memory, entity memory, and aspect memory.

Attention mechanism has shown to be effective in many applications including machine translation [2, 43], sentiment analysis [40, 44], summarization [33], and more. Given a sentence and the corresponding aspect(s), Attention-based LSTM with Aspect Embedding (ATAE-LSTM) [44] is able to predict the sentiment polarity at the aspect level. ATAE-LSTM is the first model to predict different sentiment tendencies with respect to different aspects in the same text. Aspect embedding and aspect based attention mechanism are designed in ATAE-LSTM to utilize aspect information effectively. The attention mechanism is well-designed to attend the different parts of a sentence when considering different aspects. To improve the performance of attention mechanism, [41] proposes a method named Aspect Fusion LSTM for incorporating aspect information for learning attentions. Motivated by similar reason, [18] proposes content attention with two enhancing attention mechanism. Multi-Head attention [43] jointly attends to information from different representation subspaces at different positions. In the proposed AS-Capsules model, we design attentions for every representation module in each capsule, which benefit from both low-level representation and high-level representation through different levels of encoder. In addition, a shared attention is able to generating shared representation as a part of both aspect and sentiment representations.

3 ASPECT-LEVEL SENTIMENT CAPSULES

The aspect-level sentiment capsules (AS-Capsules) model has its root in RNN-Capsule. Next we briefly describe RNN-Capsule, then detail the design of AS-Capsules model and its optimization method.

3.1 Preliminary: RNN-Capsule

Recurrent Neural Network. As the name suggests, RNN-Capsule is based on RNN. A recurrent neural network (RNN) is able to exhibit dynamic temporal behavior for a time sequence through connections between units. A unit can be realized by an LSTM model, a GRU model, or their variants. RNNs can be bi-directional, by using a finite sequence to predict or label each element in the sequence based on the element's past and future contexts. This is achieved by concatenating the outputs of two RNNs, one processes the sequence from left to right, and the other from right to left.

Briefly speaking, in an RNN realized by LSTM, the hidden states h_t and memory cell c_t in LSTM is a function of the previous h_{t-1} and c_{t-1}, and input vector x_t, or formally as follows:

$$c_t, h_t = \mathrm{LSTM}(c_{t-1}, h_{t-1}, x_t) \tag{1}$$

The hidden state h_t denotes the representation of position t while encoding the preceding contexts of the position. More details about LSTM are given in [9].

RNN-Capsule. RNN-Capsule is designed to predict the sentiment category (*e.g.,* positive, negative, and neutral) of a given piece of text. The input text is encoded by RNN and the hidden vector representations are input to all capsules. One capsule is built for one sentiment category and each capsule contains an attribute, a state, and three modules. The attribute of a capsule reflects its dedicated sentiment category (*e.g.,* positive). The three modules are: (i) representation module for building capsule representation using attention mechanism, (ii) probability module for predicting the capsule's state probability based on its representation, and (iii) reconstruction module for rebuilding the representation of the input instance. A capsule's state is 'active' if the output of its probability module is the largest among all capsules, and 'inactive' otherwise.

There are two learning objectives in RNN-Capsule network. The first is to maximize the state probability of the capsule corresponding to the groundtruth sentiment, and to minimize the state probabilities of other capsule(s). The second is to minimize the distance between the input representation and the reconstruction representation of the capsule corresponding to the ground truth, and to maximize such distances for other capsule(s).

RNN-Capsule is not designed for aspect-level sentiment analysis and each capsule in RNN-Capsule corresponds to one sentiment category. The sentiment category predicted by RNN-Capsule therefore does not reflect the sentiment on any particular aspect. Aspect-level sentiment classification relies on aspect category heavily, so well-designed shared components for the two subtasks will benefit a lot. The high-level shared RNN between capsules is important, it allows capsules to cooperate to prevent capsules from attending conflict parts.

3.2 The AS-Capsules Model

The architecture of the proposed AS-Capsules model is depicted in Figure 1. The number of the capsules M equals to the number

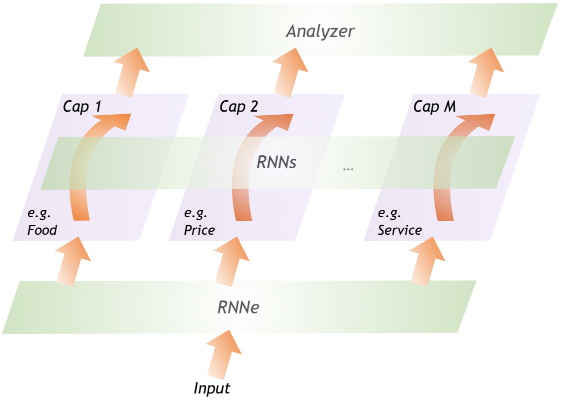

Figure 1: Architecture of AS-Capsules Model. Number of capsules equals the number of aspect categories. The hidden vectors H_1 are encoded by RNN$_e$, which encodes the input text. All capsules take H_1 as input and each capsule outputs its aspect probability p and sentiment distribution $\mathcal{P}$.

of predefined aspect categories. For example, we need 5 capsules to model a set of 5 categories {*Food*, *Price*, *Service*, *Ambience*, *Anecdote/miscellaneous* }, and a capsule is built for each aspect category. As in RNN-Capsule [45], we use an encoder RNN named RNN$_e$ to encode the input text.

Given a piece of text (*e.g.,* a sentence, a paragraph, a document), RNN$_e$ encodes the given instance and outputs the hidden representations H_1. Briefly speaking, in the encoder RNN, the hidden matrix H_1 are function of RNN$_e$ and input word representations W, or formally:

$$H_1 = \text{RNN}_e(W) \tag{2}$$

The word representations $W = [w_1, w_2, \ldots, w_N]$ are obtained from glove, and N is the length of input text. The base unit of RNN is an LSTM, GRU, or their variants, *e.g.,* bi-directional LSTM.

All the capsules take the same hidden vectors H_1 as their input, and they share an RNN named RNN$_s$. Obviously, different aspect ought to attend different parts of input text, which is not well-designed in RNN-Capsule [45]. RNN$_s$ allows capsules to communicate with each other, which is capable of preventing capsules from attending conflict parts. The high-level hidden representations are the hidden vectors of RNN$_s$. High-level here means that the input of RNN$_s$ is the hidden representation of RNN$_e$. There are special designed attentions in capsule, and we will explain them in Section 3.3. Each capsule outputs an aspect probability and a sentiment distribution, through its aspect probability module and sentiment distribution module, respectively. Analyzer refers to the training objective strategy, which is capable of utilizing the correlation between aspect detection and aspect-level sentiment classification to improve the performance.

3.3 Structure of A Single Capsule

The structure of a single capsule is shown in Figure 2. A capsule contains *an attribute*, *a state*, *a capsule embedding*, and *four modules* (aspect representation module, aspect probability module, sentiment representation module, and sentiment distribution module).

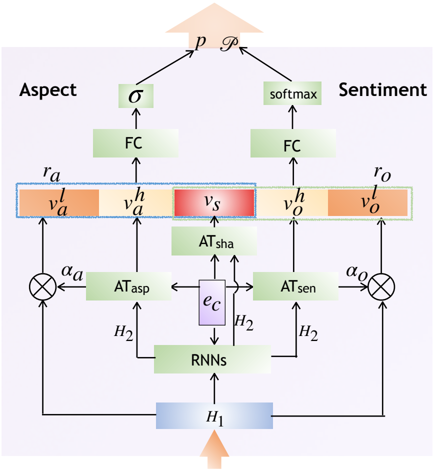

Figure 2: The architecture of a single capsule. The input to a capsule is the hidden vectors H_1 from RNN$_e$. e_c is the capsule embedding. The output is aspect probability p and sentiment distribution $\mathcal{P}$ of this capsule.

- The attribute of a capsule reflects its dedicated aspect category, which is pre-assigned when we build the capsule. Depending on the number of aspect categories in a given problem, the same number of capsules are built to reflect each aspect category.
- The state of a capsule, *i.e.,* 'active' or 'inactive', is determined by aspect probability. To learn the model, a capsule's state is active if the current aspect appears in input text. Then AS-Capsules maximizes the aspect probability of the current active capsule. In testing, a capsule's state will be active if its aspect probability p is above a predefined threshold *e.g.,* 0.5. Note that, a given piece of text may contain opinions on multiple aspects, then multiple capsules will be active.
- Capsule embedding e_c is a vector representation of the current capsule learned during training as in [44]. Recall that each capsule is assigned to learn one particle embedding and different aspects often demonstrate different word features for expressing the aspect and sentiment tendency. For example, "the pizza is delicious" demonstrates *food* aspect with positive sentiment. Here, the words "pizza" and "delicious" will not be applicable for other aspects like *service* or *price*. Therefore, for each capsule, we learn its capsule embedding.
- Aspect representation module learns the aspect representation r_a including three parts $[v_a^l, v_a^h, v_s]$, as shown in Figure 2. Given the hidden representations H_1 as input, we compute the high-level representation H_2 through RNN$_s$. The first part v_a^h of r_a is the weighted representation using the aspect-based attention AT$_{\text{asp}}$ with capsule embedding e_c and H_2 as input. Utilizing the attention scores α_a of AT$_{\text{asp}}$ and H_2, we can get the second representation v_a^l for aspect. The shared representation v_s, which is the output of the shared attention AT$_{\text{sha}}$, is the last part of r_a. AT$_{\text{sha}}$ is shared with sentiment representation module. The aspect probability module then predicts the capsule's aspect probability p based on r_a. Similarly, sentiment representation module computes sentiment representation r_o

of this capsule through similar method. Based on r_o, the sentiment distribution module generates sentiment distribution $\mathcal{P}$ on the predefined sentiment categories *e.g.,* positive, negative, neutral.

The essence of the AS-Capsules model is the four modules briefed above. Next, we detail the four modules.

Aspect Representation Module. Given the hidden vectors H_1 encoded by RNN_e, we are able to compute high-level hidden representations H_2 using RNN_s, which is shared by all capsules. Then, we use capsule embedding e_c and two attention mechanisms known as aspect-based attention AT_{asp} and shared attention AT_{sha}, to construct aspect representation r_a inside each capsule. The aspect-based attention attends the indicative words based on the aspect detection task. The shared attention benefits both aspect detection and sentiment classification. Our formulation is inspired by [2, 4, 44, 47]. Specifically, given H_1 as the low-level representations of the input text, we get a high-level hidden representations through

$$H_2 = \text{RNN}_s(H_1, e_c), \tag{3}$$

where, RNN_s is the shared encoder. The input of RNN_s is the concatenation of H_1 and e_c to enforce the RNN to attend the aspect correlated parts, inspired by AE-LSTM [44]. After getting H_2, we compute the high-level aspect representation v_a^h and attention weights α_a through aspect-based attention AT_{asp}, or formally:

$$v_a^h, \alpha_a = \text{AT}_{\text{asp}}(H_2, e_c) \tag{4}$$

Aspect-based attention, sentiment-based attention and shared attention have the same structure, so here we only detail the attention mechanism in AT_{asp}:

$$M_a = \tanh(W_h H_2 + W_c e_c \otimes N) \tag{5}$$

$$\alpha_a = \text{softmax}(w_a^T M_a) \tag{6}$$

$$v_a^h = H_2 \alpha_a^T \tag{7}$$

Here, $e_c \otimes N$ is the operator that repeatedly concatenates e_c for N times. W_h, W_c and w_a are the parameters of the current capsule for the aspect-based attention layer. The attention importance score α_a is obtained by multiplying the representations with the weight matrix, and then normalizing to a probability distribution over the words. Lastly, the high-level aspect representation vector v_a^h is a weighted summation over all the positions using the attention importance scores as weights.

The shared attention AT_{sha} is computed in a similar manner as the aspect-based attention.

$$v_s = \text{AT}_{\text{sha}}(H_2, e_c) \tag{8}$$

To obtain the original, or low-level aspect representation v_a^l from the input of capsule, we use the attention weights from aspect-based attention to weight the low-level input H_1.

$$v_a^l = H_1 \alpha_a^T \tag{9}$$

Lastly, we get the aspect representation by concatenating v_a^h, v_a^l and v_s,

$$r_a = [v_a^h, v_a^l, v_s], \tag{10}$$

where square brackets refer to the concatenation of vectors.

Aspect Probability Module. The aspect probability p is computed by a sigmoid function after getting the aspect representation r_a.

$$p = \sigma(W_p r_a + b_p), \tag{11}$$

where W_p and b_p are the parameters for the aspect probability module of the current capsule.

Sentiment Representation Module. The sentiment representation module is computed in a similar manner as the aspect representation module. The sentiment representation r_o contains three parts, known as high-level sentiment representation v_o^h, low-level sentiment representation v_o^l and shared representation v_s. The high-level sentiment representation v_o^h is computed in sentiment-based attention:

$$v_o^h, \alpha_o = \text{AT}_{\text{sen}}(H_2, e_c) \tag{12}$$

After getting the attention importance scores α_o in sentiment representation module, we utilize the low-level information through

$$v_o^l = H_1 \alpha_o^T \tag{13}$$

In the two formulas above, AT_{sen} is the sentiment-based attention. e_c is the capsule embedding. H_1 and H_2 are the hidden representations of RNN_e and RNN_s. The attention importance score for each position for sentiment is α_o. The sentiment representation vector r_o is obtained by concatenating v_o^h, v_o^l and v_s.

Note that, both the aspect representation and the sentiment representation obtained from attention layer are high-level encodings of the entire input text. The attention mechanism designed in the model is for improving the model's capability and robustness.

Sentiment Distribution Module. The sentiment distribution $\mathcal{P}$ is computed by a softmax function after getting the sentiment representation vector r_o.

$$\mathcal{P} = \text{softmax}(W_p' r_o + b_p'), \tag{14}$$

where W_p' and b_p' are the parameters for the sentiment distribution module of the current capsule.

The above four modules complement each other in the AS-Capsules model. From a macro perspective *i.e.,* the full dataset, the words attended by different capsules match the capsules' attribute. From micro perspective *i.e.,* a piece of input text, the state 'active' or 'inactive' of a capsule is determined by its aspect probability p. The sentiment distribution is with respect to the aspect category of the current capsule.

There are two different ways to utilize the correlation between aspect and sentiment. One way is to draw support from the capsule structure. In our first attempt, we design the capsule structure with two independent attentions based on their own embeddings named aspect embedding and sentiment embedding. However, the two independent embeddings have the same effect, and cannot benefit from the correlation between aspect and sentiment effectively. Motivated by shared embedding [44] and multi-head attention [43], in our current design, aspect representation module and sentiment representation module share one embedding known as capsule embedding. Specially, we design aspect-based attention, sentiment-based attention and shared attention. Aspect-based attention and sentiment-based attention are shared by low-level and high-level RNN encoders to get hierarchical representations of input text. Shared attention is used to generate shared representation for both aspect detection and sentiment classification. The other way to utilize the correlation considers the co-occurrence of aspect and sentiment pair. We use the idea of mask. That is, we only keep the cross entropy of aspect(s) that appear in the input text and ignore the irrelevant aspect(s) in the learning objective.

3.4 Training Objective

The training of the proposed AS-Capsules model has two objectives. One is to maximize the aspect probability of active capsule(s) matching the ground truth and minimize aspect probability of the inactive capsule(s). The other is to minimize the cross-entropy of sentiment distribution of active capsules.

Aspect Probability Objective. A given text may express sentiments on multiple aspects. Hence, we have both positive sample(s) (*i.e.,* the active capsule(s)) and negative sample(s) (*i.e.,* the inactive capsule(s)). Recall that our objective is to maximize the aspect probability of active capsules and to minimize the aspect probability of inactive capsules. The classification objective J can be formulated by cross entropy loss:

$$J(\theta) = \sum \frac{1}{M} \sum_{i=1}^{M} \text{cross-entropy}(y_a^i, [p_i, 1 - p_i]), \qquad (15)$$

where p_i is the aspect probability of the capsule i. For a given training instance, $y_a^i = 1$ for an active capsule (*i.e.,* the corresponding aspect occurs in the given text), and $y_a^i = 0$ for an inactive capsule. M is the number of aspect categories, where is 5.

Aspect-level Sentiment Classification Objective. The other objective is to ensure the accuracy of aspect-level sentiment classification of the active capsule(s). Similarly, the unregularized objective U can be formulated as cross entropy loss:

$$U(\theta) = \sum \frac{1}{\sum_{\forall y_a^i=1} 1} \sum_{\forall y_a^i=1} \text{cross-entropy}(y_o^i, \mathcal{P}_i) \qquad (16)$$

We only utilize the cross-entropy loss of the 'active' capsule(s). y_o^i is the groundtruth sentiment of the 'active' capsule i. $\mathcal{P}_i$ is the predicted sentiment distribution of the capsule i.

Considering both objectives, our final objective function L is obtained by adding J and U:

$$L(\theta) = J(\theta) + U(\theta) \qquad (17)$$

4 EXPERIMENT

We now evaluate the proposed AS-Capsules for aspect detection and aspect-level sentiment classification, against baselines.

4.1 Dataset and Model Implementation Details

Dataset. We conduct experiments on SemEval 2014 Task 4 dataset [27].[2] The dataset consists of customer reviews for laptops and restaurants, but only restaurant reviews are annotated with aspect-specific polarity. Hence we conduct experiments on restaurant reviews.

The restaurant reviews consist of about 3K English sentences from [5]. We randomly cut out $\frac{1}{8}$ as validation dataset, and the rest is the training dataset. Additional restaurant reviews, not in the original dataset of [5], are used as test data. The set of aspects $\mathcal{A}$ is {*food, service, price, ambience, anecdote*}. There are four sentiment categories: *positive*, *neutral*, *negative* and *conflict*. Each sentence is assigned one or more aspects together with a polarity label for each aspect, *e.g.,* "Staffs are not that friendly, but the taste covers all." would be assigned the aspect-sentiment pairs $\{\langle\text{food, positive}\rangle, \langle\text{service, negative}\rangle\}$. In our experiments, we use the first three sentiment categories, as in most other studies.

Table 1: The statistics of restaurant reviews on SemEval 2014 Task 4 dataset. The sentiment category 'conflict' is not used.

Aspect	Positive		Negative		Neural	
	Train	Test	Train	Test	Train	Test
Food	867	302	209	69	90	31
Price	179	51	115	28	10	1
Service	324	101	218	63	20	3
Ambience	263	76	98	21	23	8
Anecdote	546	127	199	41	357	51
Total	2179	657	839	222	500	94

Table 1 presents the statistics of the SemEval 2014 Task 4 dataset. The two aspects *food* and *anecdote* have the largest number of instances. However, it is hard to think about words that are indicative for aspect *anecdote*.

Implementation Details. In our experiments, all word vectors are initialized by Glove [25].[3] The pre-trained word embeddings and capsule embedding have dimensions of 300 and 256, respectively. The dimension of hidden vectors encoded by RNN is 256 (hence 512 if RNN is bidirectional). We use a single layer bidirectional LSTM in AS-Capsules. The model is trained with a batch size of 16 examples, and there is a checkpoint every 8 mini-batch due to the small size of dataset. Dropout is 0.5 for word embedding and linear layer in aspect probability module and sentiment distribution module.

We implement our model on Pytorch (version 0.4).[4] Model parameters are randomly initialized. Adam [15] is utilized to optimize our model, and we use $1e - 3$ as learning rate for model parameters except word vectors, and $1e - 4$ for word vectors.

4.2 Evaluation on Three Subtasks

SemEval 2014 task 4 dataset is widely used for aspect-level sentiment analysis. Because of the detailed annotation, multiple evaluations can be conducted. We report our experiments on three subtasks: aspect detection, sentiment classification on given aspects, and aspect-level sentiment analysis.

4.2.1 Subtask 1: Aspect Detection. Given a piece of text, the task of aspect detection is to predict the existence of predefined aspects, which is a typical multi-label classification task. We compare AS-Capsules with state-of-the-art baselines designed for aspect detection. Specially, we compare detailed F_1 of all categories with effective joint deep learning baselines including Bi-LSTM, AE-LSTM, AT-LSTM, and RNN-Capsule.

[2] Refer to http://alt.qcri.org/semeval2014/task4/ for more details.

[3] http://nlp.stanford.edu/projects/glove/

[4] https://github.com/pytorch

Table 2: Micro F_1 of methods on SemEval 2014 Task 4 dataset. Best results are in bold face and second best underlined.

Model	micro-F_1	Model	micro-F_1
KNN	63.9	HLBL	69.7
LR	66.0	C&W	72.5
SVM	80.8	word2vec	83.3
SemEval-Average	73.8	Hybrid-WRL-300	88.6
NRC-Lexicon	84.1	Hybrid-WRL-Best	**90.1**
NRC	88.6	AS-Capsules	89.6

KNN is the baseline provided by SemEval official [28]. First, the Dice coefficient is used to calculate the similarity to find k most similar sentences in training dataset for given test sentence. Then, m most frequent aspect categories of the k retrieved sentences will be assigned to the test sentence. m is the number of most frequent aspect categories per sentence among the k sentences. Logistic Regression (LR) and Support Vector machine (SVM) are used as classifiers using unigram and bigram features. SemEval-Average is the average result of all the systems in SemEval 2014. NRC, the best system in SemEval 2014, adopts SVM as the classifier with some well-designed features including n-grams, stemmed n-grams, character n-grams, non-contiguous n-grams, word cluster n-grams and lexicons. NRC-Lexicon is the result without the lexicon feature.

Some word representation methods are compared with our AS-Capsules. C&W [3] and word2vec [20] are the powerful and accepted generally methods for word representation learning. HLBL [21] is a hierarchical model, which performs well using a carefully constructed hierarchy over words. Hybrid-WRL [50] is a word representation learning method using hybrid features including shared-features and aspect-specific features.

Table 2 lists the micro F_1 of aspect detection on restaurants reviews. Our proposed AS-Capsules model achieves the second best. Notice that, there are two kinds of Hybrid-WRL methods. The difference between them is the size of word representation. Hybrid-WRL-300 uses 300 as word representation size, 600 for Hybrid-WRL-Best. Our AS-Capsules adopts 300 as the dimension of word representation, improves 1 percentage than Hybrid-WRL at the same size of word representation. Obviously, there are more complexity and parameters with bigger size of word representation. For classical machine learning methods, SVM performs better than KNN and LR. NRC achieves the best SemEval result with the textual features and lexicons features using SVM as classifier. Without Lexicons features, NRC-Lexicon is 4 percentage lower than NRC. Word representation methods perform better than KNN, LR and SVM, however they perform worse than NRC with well-designed features.

Although classical machine learning methods have shown their effectiveness, it is better to compare with neural network methods. Bidirectional-LSTM (Bi-LSTM) is a variant of LSTM which is introduced in Section 3.1. It is capable of utilizing the content information through the bidirectional structure. Aspect Embedding LSTM (AE-LSTM) is proposed in [44], where aspect embeddings are concatenated with word vectors. AE-LSTM considers aspect information so it is expected to perform better than Bi-LSTM. Different parts of input sentence have different importance, so attention is a powerful way to address this problem. Attention based LSTM (AT-LSTM) [44] further uses attention mechanism, which performs better than other baselines. RNN-Capsule uses each capsule to detect one category, sentiment category in its original paper [45], and aspect in this experiment.

Table 3: The average F_1 and F_1 of different aspects for Subtask 1: aspect detection.

Model	Average	Food	Price	Service	Ambience	Anecdote
Bi-LSTM	83.0	92.6	79.2	88.8	74.8	79.6
AE-LSTM	84.5	**93.8**	83.3	88.8	78.2	78.5
AT-LSTM	84.5	91.6	83.0	85.6	83.0	79.2
RNN-Capsule	85.5	**93.8**	85.4	89.4	80.6	78.3
AS-Capsules	**87.2**	93.4	**85.9**	**91.0**	**83.3**	**82.4**

Table 4: Accuracy and F_1 of sentiment categories for Subtask 2: sentiment classification on given aspects.

Model	Accuracy	F_1-Positive	F_1-Neutral	F_1-Negative
Bi-LSTM	82.1	89.2	49.7	73.2
AE-LSTM	82.8	89.8	54.2	72.6
AT-LSTM	83.0	89.8	47.3	75.8
ATAE-LSTM	84.3	90.1	**61.9**	77.5
AS-Capsules	**85.0**	**91.2**	50.7	**78.7**

Reported in Table 3, our proposed AS-Capsules is the best performing method, followed by RNN-Capsule. Specifically, AS-Capsules achieves the best average F_1, best results on four aspects except aspect *food.* The gap between AS-Capsules and the best model in aspect *food* is very small. AE-LSTM and AT-LSTM deliver similar results and Bi-LSTM performs the poorest.

4.2.2 Subtask 2: Sentiment Classification on Given Aspects. Given a piece of text and also the annotated aspect, the task is to predict the sentiment expressed in the text on the given aspect. For this subtask, we compare AS-Capsules with four baselines: Bi-LSTM, AE-LSTM, AT-LSTM, and Attention-based LSTM with Aspect Embedding (ATAE-LSTM) [44]. ATAE-LSTM unitizes aspect information to attend the important words in sentence with respect to a specific aspect. Note that RNN-Capsule cannot be applied to this subtask because RNN-Capsule is self-attentive, and it cannot take aspect as an additional input.

From the results reported in Table 4, we observe that AS-Capsules achieves the best accuracy of 85.0. It outperforms all baselines with respect to the F_1 results on the *positive* and *negative* sentiment categories. Due to less neutral data in dataset, AS-Capsules performs not good enough. Among the baseline methods, ATAE-LSTM outperforms the rest. AE-LSTM and AT-LSTM perform better than Bi-LSTM. Among three sentiment categories, *positive* is much easier to predict and *neutral* is the most difficult category due to the smallest size of data.

4.2.3 Subtask 3: Aspect-level Sentiment Analysis. Given a piece of text, subtask 3 requires a method to detect $\langle aspect, sentiment\rangle$ pair(s) from the input text. A detected pair is considered correct if

Table 5: Accuracy and F_1 of different sentiment categories for Subtask 3: aspect-level sentiment analysis.

Model	Accuracy	F_1-Positive	F_1-Neutral	F_1-Negative
Bi-LSTM	62.3	80.2	43.6	56.6
AE-LSTM	64.7	82.4	50.3	55.7
AT-LSTM	65.6	82.4	46.6	57.7
AS-Capsules	**68.1**	**83.3**	**53.6**	**61.6**

both components in the pair are correctly identified. We compare AS-Capsules with Bi-LSTM, AE-LSTM and AT-LSTM. Again, RNN-Capsule is not designed to classify sentiment with given aspect(s) and ATAE-LSTM needs aspect as additional input; hence these two models are not applicable to this task.

Reported in Table 5, AS-Capsules model delivers the best results on accuracy. It also outperforms all baselines with respects to the F_1 results on the three sentiment categories. AE-LSTM and AT-LSTM perform similarly and both outperform Bi-LSTM. Similar to earlier observations, sentiment *positive* is relatively easier to detect and *neutral* is the hardest category.

5 EXPLAINABILITY ANALYSIS

We have shown that AS-Capsules model outperforms all baseline models for aspect-level sentiment analysis. Because of the attention mechanism, AS-Capsules model is able to attend meaningful words in the aspect representation module and sentiment representation module. Specifically, each word is assigned an attention weight in aspect representation module and sentiment representation module by multiplying *aspect probability* with *aspect attention weight* and *sentiment attention weight*, respectively.

As a case study, we show the words attended by AS-Capsules during test. That is, after all test instances are evaluated, we obtain two lists of attended words from each capsule with their attention weights for aspect and sentiment respectively. Due to page limit, we can only display a small number of words with some ranking criteria. A straightforward ranking is by the *averaged attention weight* of a word. By this ranking, most top-ranked words are of low frequency. That is, some words have significant attention weight (or strong aspect or sentiment tendencies) but do not appear very often. Another way of ranking is by the product of *averaged attention weight* and *the logarithm of word frequency.*

Words Attended for Aspect Detection. Table 7a lists the top-ranked 20 words by the product of average attention weight and logarithm of word frequency, for the five aspects. Table 7b lists the top-ranked 20 words by average attention weight.

From the two tables, we observe that almost attended words are self-explanatory for the assigned aspect category. For instance, the attended words by capsule *food* are mostly food categories or ingredients. Capsule *service* attends words for personnel involved and their attitude. Without the need of any linguistic knowledge, the AS-Capsules model is able to identify words that reflect the aspect categories. This provides an easy way to build domain specific lexicons on domain specific data.

Table 6: Word attended by sentiment representation module in Capsule *food*. Significant words and low frequency words are ranked by *average attention weight* × log*(word frequency)* and *average attention weight*, respectively.

No.	Significant words	Freq	Low frequency words	Freq
1	delicious	22	divine	1
2	great	67	satisfying	2
3	tasty	8	favourites	1
4	good	55	scrumptious	2
5	best	24	terrific	1
6	fresh	23	yummy	5
7	yummy	5	greatest	1
8	excellent	23	frosty	1
9	amazing	7	delicious	22
10	outstanding	5	fave	1
11	fantastic	5	luscious	1
12	wonderful	8	tasty	8
13	mouth	5	unexpected	1
14	superb	3	winner	1
15	delectable	3	refreshing	2
16	recommend	8	recomend	1
17	satisfying	2	flavor	2
18	scrumptious	2	highlight	1
19	perfect	6	lemons	1
20	sweet	5	delicate	1

Words Attended for Aspect-level Sentiment Classification. We show the sentiment words attended by AS-Capsules for capsule *food* with *positive* sentiment tendency. Other aspects are omitted due to limited space. Similarly, Table 6 lists the top-ranked 20 significant words and low frequency words ranked by *average attention weights* × *logarithm of word frequency* and *average attention weight*, respectively. These words are consistent with sentiment lexicons identified in related studies [12, 26, 31]. All of those words reflect the positive sentiment tendency significantly. The attention weights of all significant words are above 0.35. More importantly, the words are used commonly. For low frequency words, the attention weights are over 0.77. Even though the words are used not so often, they belong to small but beautiful word group. 'delicious' exists in both columns due to a very positive sentiment tendency and common use. The *neutral* sentiment attends lots of punctuation marks and meaningless words *e.g.*,, 'and', 'a' and 'is', so they are not shown. *Negative* sentiment attends significant words including 'terrible', 'worst' and 'disappointed' whose attention weights are over 0.1.

5.1 Applying AS-Capsules on Yelp Reviews

To the best of our knowledge, the SemEval 2014 Task 4 dataset is the only dataset that comes with aspect-level sentiment annotations. This limits the evaluation of our model on aspect-level sentiment analysis. On the other hand, many restaurant reviews are available from other domains without aspect-level manual annotation. The Yelp dataset[5] is an example. As most reviews on Yelp are for restaurants, we can directly apply the trained AS-Capsules to conduct a qualitative evaluation. Because there are no aspect-level annotations on Yelp dataset, we are unable to report quantitative measures

[5] https://www.yelp.com/dataset/challenge

Table 7: Words attended by aspect representation module in all capsules on SemEval dataset.

(a) Top ranked words by average attention weight × log(word frequency).

No.	Food	Freq	Price	Freq	Service	Freq	Ambience	Freq	Anecdote	Freq
1	food	150	prices	18	service	76	atmosphere	21	!	23
2	sushi	22	cheap	10	staff	28	decor	5	meal	3
3	pizza	14	price	11	waiter	10	ambiance	5	.	172
4	meal	12	priced	7	waiters	7	space	4	restaurant	20
5	menu	23	value	5	bartender	6	music	5	sushi	3
6	desserts	8	inexpensive	3	attentive	8	cozy	6	food	7
7	portions	7	expensive	4	waitress	4	ambience	3	menu	6
8	pasta	5	bill	4	owner	5	interior	2	experience	11
9	wine	14	overpriced	2	servers	3	room	4	place	28
10	sauce	11	money	3	hostess	3	intimate	4	italian	4
11	shrimp	7	affordable	2	friendly	22	quiet	3	in	43
12	soup	9	pay	3	courteous	3	clean	2	the	98
13	dessert	5	over	3	greeted	3	chic	2	dining	4
14	dishes	9	reasonable	5	politely	2	crowded	2	at	26
15	cheese	10	the	81	accomodating	2	scene	2	money	3
16	seafood	6	for	23	rude	9	downstairs	2	pizza	2
17	chicken	17	your	4	prompt	5	,	78	for	36
18	cuisine	4	reasonably	4	bartenders	2	place	21	night	7
19	crab	5	worth	4	manager	4	laid-back	4	here	17
20	crust	4	at	14	asked	3	like	8	this	53

(b) Words ranked by average attention weight.

No.	Food	Freq	Price	Freq	Service	Freq	Ambience	Freq	Anecdote	Freq
1	truffles	1	overpriced	2	service	76	claustrophobic	1	thai	1
2	hotdogs	1	pricey	1	politely	2	chairs	1	cost	1
3	meat	1	inexpensive	3	accomodating	2	bathroom	1	meal	3
4	meats	1	cheap	10	lady	1	patio	1	agave	1
5	wines	2	deal	1	servers	3	decor	5	service	1
6	sangria	2	prices	18	hostess	3	chill	1	desserts	1
7	pancakes	1	price	11	solicitous	1	decoration	1	delicacy	1
8	mojitos	1	priced	7	brusquely	1	landscaping	1	sum	1
9	codfish	1	value	5	bartenders	2	pretentious	1	pizza	2
10	crepes	1	wallet	1	courteous	3	singer	1	tacos	1
11	pizzas	1	expensive	4	port	1	setting	1	sushi	3
12	breads	1	cost	1	waitress	4	lighting	1	martinis	1
13	guacamole	1	bill	4	greeted	3	atmosphere	21	delivary	1
14	tequila	1	spend	1	polite	2	space	4	crepes	1
15	cookies	1	14	1	staff	28	bumping	1	steak	1
16	pudding	1	cheaper	1	awful	1	rooftop	1	astoria	1
17	marscapone	1	investment	1	waiters	7	uncomfortably	1	taste	1
18	meatball	1	50	1	gracious	1	air	1	brunch	2
19	bbq	2	affordable	2	bartender	6	ambiance	5	diamond	1
20	chili	2	6.25	1	owner	5	interior	2	stock	1

like accuracy or F_1. However, it is interesting to observe whether the AS-Capsules model learned on SemEval 2014 Task 4 dataset can be used to identify meaningful words on Yelp dataset, to reflects its aspects and sentiment categories.

In this case study, we take the first 1,000 reviews from Yelp dataset. As Yelp reviews are relatively long and each review has several paragraphs so we split the reviews by paragraphs and consider each paragraph a test input to AS-Capsules model. As a result, we have 3,776 test instances from Yelp dataset.

Words Attended for Aspect Detection on Yelp. Table 8 lists the top ranked words identified by AS-Capsules for the five aspects. We provide two rankings following our earlier ranking criteria. Observe that regardless word frequencies, most of them well reflect the corresponding aspect category. For example, 'beers', 'sauce' and 'pizza' are identified as important words for *food*. Interestingly, most low frequency words are meaningful to the related aspects. For example, 'porridge', 'terrine' and 'fillings' are attended by capsule *food*. More interestingly, some numbers *e.g.,* 45, 50 and 20, are

Table 8: Words attended by aspect representation module on Yelp dataset.

No.	Ranked by average attention weight × log(word frequency)					Ranked by average attention weight				
	Food	Price	Service	Ambience	Anecdote	Food	Price	Service	Ambience	Anecdote
1	food	prices	service	atmosphere	!	chilaquiles	pricy	managers	jazz	gelato
2	beers	cheap	staff	decor	craving	porridge	affordable	station	scene	pies
3	sauce	price	friendly	patio	again	terrine	overpriced	staffs	smoke	greeted
4	beer	overpriced	polite	cramped	breakfast	fillings	overcharged	unattentive	pretension	catering
5	menu	priced	server	ambiance	sushi	oatmeal	prices	waitstaff	cramped	panini
6	foods	expensive	waitress	music	eaten	miso	inexpensive	politely	noisy	intriguing
7	pizza	pricey	employees	vibe	pasta	sauces	cheap	greeting	venues	eww
8	meal	deal	courteous	crowded	burger	milkshakes	priced	apologetic	cozy	ahhh...
9	cheese	cost	greeted	bathroom	buying	wintermelon	expensive	answering	rowdy	cakes
10	meat	buy	attentive	rooms	dinner	concepts	pricey	polite	atmosphere	sandwiches
11	chicken	cheaper	bartender	seating	eat	beers	price	courteous	decor	burgers
12	sushi	dollar	servers	cozy	grocery	unagi	cheaper	marketing	nondescript	treated
13	beef	pay	manager	walls	menu	shakes	deal	service	crowded	soeur
14	dessert	affordable	waiter	relaxing	sandwiches	cheeses	45	greet	pretentious	tidy
15	burger	bucks	desk	floor	service	foods	costs	handled	pop	salty
16	sauces	million	patient	noisy	dining	appetite	cost	staff	interior	pasta
17	pork	pricing	waitstaff	space	lunch	concoctions	600	servers	jukebox	donut
18	ingredients	buck	apologetic	loud	heaven	maya	pricing	manager	claustrophobic	digress
19	burgers	charged	answering	air	brunch	tartare	50	employees	ornate	nooo...
20	lemonade	inexpensive	prompt	interior	steakhouse	pig	25	greeted	mellow	bagel

Table 9: Word attended by sentiment representation module in Capsule *food* on Yelp dataset.

No.	Significant words	Freq	Low frequency words	Freq
1	delicious	108	dynamic	1
2	tasty	51	delicious	108
3	yummy	31	yummy	31
4	excellent	28	adventurous	3
5	fresh	51	friendliest	1
6	fantastic	37	satisfying	6
7	great	203	superb	2
8	good	390	tasty	51
9	amazing	71	meaty	1
10	flavorful	13	palates	1
11	flavor	79	heavenly	6
12	satisfying	6	scrumptious	4
13	flavour	13	chai	1
14	incredible	14	efficient	1
15	heavenly	6	loves	3
16	awesome	34	delectable	1
17	enjoy	44	excellent	28
18	flavors	20	palate	3
19	enjoying	9	flavorful	13
20	goodness	9	bite	6

attended in capsule *price*. The phenomenon reflects that our AS-Capsules model performs well for low frequency words.

Words Attended for Aspect-level Sentiment Classification on Yelp. We now present the words attended by sentiment representation module. Table 9 lists the top ranked significant words and low frequency words identified by capsule *food* for *positive* sentiments. Most of the words attended by *neutral* are not very meaningful, so they are not shown. Most of the attended words fit their sentiment tendency in sentiment lexicons. Specially, we observe that most of them are able to reflect *food* aspect. As we mentioned before, 'delicious' is a very common word for describing food, it is also shown in low frequency words because we rank the attended words by their attention weight.

6 CONCLUSION

In this paper, we study aspect-level sentiment analysis and propose aspect-level sentiment capsules (AS-Capsules) model. The key idea of AS-Capsules model is to use capsule structure to focus on each aspect category. Each capsule outputs its aspect probability and sentiment distribution on the targeted aspect. The objective of learning is to maximize the aspect probability of the capsule(s) matching the groundtruth and to minimize its (their) sentiment cross entropy loss. Through shared components including capsule embedding, shared encoders and shared attentions, our model utilizes the correlation between aspects and corresponding sentiments effectively. Experiments show that the proposed AS-Capsules model achieves state-of-the-art performance without the need of linguistic knowledge. We show that the capsules are able to identify words best reflect the aspect category and sentiment tendency. We also show that the model can be directly applied to restaurant reviews on Yelp, demonstrating its effectiveness and robustness.

ACKNOWLEDGMENTS

This work was partially supported by Singapore Ministry of Education Research Fund MOE2014-T2-2-066, the National Key R&D Program of China (Grant No. 2018YFC0830200), the National Science Foundation of China (Grant No. 61876096/61332007), and China Scholarship Council.

REFERENCES

[1] Abdulaziz Alghunaim, Mitra Mohtarami, Scott Cyphers, and Jim Glass. 2015. A Vector Space Approach for Aspect Based Sentiment Analysis. In *Proc. NAACL HLT*. 116–122.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural Machine Translation by Jointly Learning to Align and Translate. *CoRR* abs/1409.0473 (2014).

[3] Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: deep neural networks with multitask learning. In *Proc. ICML*. 160–167.

[4] Bjarke Felbo, Alan Mislove, Anders Søgaard, Iyad Rahwan, and Sune Lehmann. 2017. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. In *Proc. EMNLP*. 1615–1625.

[5] Gayatree Ganu, Noemie Elhadad, and Amélie Marian. 2009. Beyond the Stars: Improving Rating Predictions using Review Text Content. In *Proc. WebDB*.

[6] Zhen Hai, Kuiyu Chang, and Jung-jae Kim. 2011. Implicit Feature Identification via Co-occurrence Association Rule Mining. In *Proc. Computational Linguistics and Intelligent Text Processing - International Conference, CICLing, Part I*. 393–404.

[7] Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2018. Exploiting Document Knowledge for Aspect-level Sentiment Classification. In *Proc. ACL, Volume 2: Short Papers*. 579–585.

[8] Geoffrey E. Hinton, Alex Krizhevsky, and Sida D. Wang. 2011. Transforming Auto-Encoders. In *Proc. ICANN, Part I*. 44–51.

[9] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9, 8 (1997), 1735–1780.

[10] Minqing Hu and Bing Liu. 2004. Mining Opinion Features in Customer Reviews. In *Proc. National Conference on Artificial Intelligence, Conference on Innovative Applications of Artificial Intelligence*. 755–760.

[11] Niklas Jakob and Iryna Gurevych. 2010. Extracting Opinion Targets in a Single and Cross-Domain Setting with Conditional Random Fields. In *Proc. EMNLP*. 1035–1045.

[12] Nobuhiro Kaji and Masaru Kitsuregawa. 2007. Building Lexicon for Sentiment Analysis from Massive Collection of HTML Documents. In *Proc. EMNLP-CoNLL*. 1075–1083.

[13] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A Convolutional Neural Network for Modelling Sentences. In *Proc. ACL, Volume 1: Long Papers*. 655–665.

[14] Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *Proc. EMNLP*. 1746–1751.

[15] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *CoRR* abs/1412.6980 (2014).

[16] Tao Lei, Regina Barzilay, and Tommi S. Jaakkola. 2015. Molding CNNs for text: non-linear, non-consecutive convolutions. In *Proc. EMNLP*. 1565–1575.

[17] Bing Liu. 2012. *Sentiment Analysis and Opinion Mining*. Morgan & Claypool Publishers.

[18] Qiao Liu, Haibin Zhang, Yifu Zeng, Ziqi Huang, and Zufeng Wu. 2018. Content Attention Model for Aspect Based Sentiment Analysis. In *Proc. WWW*. 1023–1032.

[19] Tomáš Mikolov. 2012. Statistical language models based on neural networks. *Presentation at Google, Mountain View, 2nd April* (2012).

[20] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Proc. NIPS*. 3111–3119.

[21] Andriy Mnih and Geoffrey E. Hinton. 2008. A Scalable Hierarchical Distributed Language Model. In *Proc. NIPS*. 1081–1088.

[22] Saif Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. 2013. NRC-Canada: Building the State-of-the-Art in Sentiment Analysis of Tweets. In *Proc. Workshop on Semantic Evaluation, SemEval@NAACL-HLT*. 321–327.

[23] Tony Mullen and Nigel Collier. 2004. Sentiment Analysis using Support Vector Machines with Diverse Information Sources. In *Proc. EMNLP*. 412–418.

[24] Bo Pang and Lillian Lee. 2007. Opinion Mining and Sentiment Analysis. *Foundations and Trends in Information Retrieval* 2, 1-2 (2007), 1–135.

[25] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global Vectors for Word Representation. In *Proc. EMNLP*. 1532–1543.

[26] Verónica Pérez-Rosas, Carmen Banea, and Rada Mihalcea. 2012. Learning Sentiment Lexicons in Spanish. In *Proc. LREC*. 3077–3081.

[27] Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. SemEval-2014 Task 4: Aspect Based Sentiment Analysis. In *Proc. Workshop on Semantic Evaluation, SemEval@COLING*. 27–35.

[28] Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. SemEval-2014 Task 4: Aspect Based Sentiment Analysis. In *Proc. SemEval@COLING*. 27–35.

[29] Ana-Maria Popescu and Oren Etzioni. 2005. Extracting Product Features and Opinions from Reviews. In *Proc. HLT/EMNLP*. 339–346.

[30] Qiao Qian, Minlie Huang, JinHao Lei, and Xiaoyan Zhu. 2017. Linguistically Regularized LSTMs for Sentiment Classification. In *Proc. ACL*, Vol. 1. 1679–1689.

[31] Delip Rao and Deepak Ravichandran. 2009. Semi-Supervised Polarity Lexicon Induction. In *Proc. EACL*. 675–682.

[32] Seyyed Aref Razavi and Masoud Asadpour. 2017. Word embedding-based approach to aspect detection for aspect-based summarization of persian customer reviews. In *Proc. IML*. 33:1–33:10.

[33] Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A Neural Attention Model for Abstractive Sentence Summarization. In *Proc. EMNLP*. 379–389.

[34] Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. 2017. Dynamic Routing Between Capsules. In *Proc. NIPS*. 3859–3869.

[35] Kim Schouten and Flavius Frasincar. 2016. Survey on Aspect-Level Sentiment Analysis. *IEEE Trans. Knowl. Data Eng.* 28, 3 (2016), 813–830.

[36] Richard Socher, Jeffrey Pennington, Eric H. Huang, Andrew Y. Ng, and Christopher D. Manning. 2011. Semi-Supervised Recursive Autoencoders for Predicting Sentiment Distributions. In *Proc. EMNLP*. 151–161.

[37] Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proc. EMNLP*, Vol. 1631. Citeseer, 1642.

[38] Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks. In *Proc. ACL, Volume 1: Long Papers*. 1556–1566.

[39] Duyu Tang, Bing Qin, Xiaocheng Feng, and Ting Liu. 2015. Target-Dependent Sentiment Classification with Long Short Term Memory. *CoRR* abs/1512.01100 (2015).

[40] Duyu Tang, Bing Qin, and Ting Liu. 2015. Document Modeling with Gated Recurrent Neural Network for Sentiment Classification. In *Proc. EMNLP*. 1422–1432.

[41] Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2018. Learning to Attend via Word-Aspect Associative Fusion for Aspect-Based Sentiment Analysis. In *Proc. AAAI*. 5956–5963.

[42] Zhiqiang Toh and Jian Su. 2016. NLANGP at SemEval-2016 Task 5: Improving Aspect Based Sentiment Analysis using Neural Network Features. In *Proc. NAACL HLT*. 282–288.

[43] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Proc. NIPS*. 6000–6010.

[44] Yequan Wang, Minlie Huang, Li Zhao, and Xiaoyan Zhu. 2016. Attention-based LSTM for Aspect-level Sentiment Classification. In *Proc. EMNLP*. 606–615.

[45] Yequan Wang, Aixin Sun, Jialong Han, Ying Liu, and Xiaoyan Zhu. 2018. Sentiment Analysis by Capsules. In *Proc. WWW*. 1165–1174.

[46] Jun Yang, Runqi Yang, Chongjun Wang, and Junyuan Xie. 2018. Multi-Entity Aspect-Based Sentiment Analysis With Context, Entity and Aspect Memory. In *Proc. AAAI*. 6029–6036.

[47] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alexander J. Smola, and Eduard H. Hovy. 2016. Hierarchical Attention Networks for Document Classification. In *Proc. NAACL HLT*. 1480–1489.

[48] Lei Zhang, Bing Liu, Suk Hwan Lim, and Eamonn O'Brien-Strain. 2010. Extracting and Ranking Product Features in Opinion Documents. In *Proc. COLING, Posters Volume*. 1462–1470.

[49] Yanyan Zhao, Bing Qin, Shen Hu, and Ting Liu. 2010. Generalizing Syntactic Structures for Product Attribute Candidate Extraction. In *Proc. HLT NAACL*. 377–380.

[50] Xinjie Zhou, Xiaojun Wan, and Jianguo Xiao. 2015. Representation Learning for Aspect Category Detection in Online Reviews. In *Proc. AAAI*. 417–424.