Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)

Coupled Multi-Layer Attentions for Co-Extraction of Aspect and Opinion Terms

Wenya Wang,[†‡] Sinno Jialin Pan,[†] Daniel Dahlmeier,[‡] Xiaokui Xiao[†]
[†]Nanyang Technological University, Singapore
[‡]SAP Innovation Center Singapore
[†]{wa0001ya, sinnopan, xkxiao}@ntu.edu.sg, [‡]{d.dahlmeier}@sap.com

Abstract

The task of aspect and opinion terms co-extraction aims to explicitly extract aspect terms describing features of an entity and opinion terms expressing emotions from user-generated texts. To achieve this task, one effective approach is to exploit relations between aspect terms and opinion terms by parsing syntactic structure for each sentence. However, this approach requires expensive effort for parsing and highly depends on the quality of the parsing results. In this paper, we offer a novel deep learning model, named coupled multi-layer attentions. The proposed model provides an end-to-end solution and does not require any parsers or other linguistic resources for preprocessing. Specifically, the proposed model is a multi-layer attention network, where each layer consists of a couple of attentions with tensor operators. One attention is for extracting aspect terms, while the other is for extracting opinion terms. They are learned interactively to dually propagate information between aspect terms and opinion terms. Through multiple layers, the model can further exploit indirect relations between terms for more precise information extraction. Experimental results on three benchmark datasets in SemEval Challenge 2014 and 2015 show that our model achieves state-of-the-art performances compared with several baselines.

Introduction

Aspect and opinion terms co-extraction, which aims at identifying aspect terms and opinion terms from texts, is an important task in fine-grained sentiment analysis (Pang and Lee 2008). An aspect term refers to a word or a phrase (a sequence of words) describing an attribute or feature of an entity, e.g., a product. An opinion term refers to the expression carrying subjective emotions. For example, in the review *"This little place has a cute interior decor and affordable prices"*, *interior decor* and *prices* are aspects, with *cute* and *affordable* as their corresponding opinions.

In the literature, there exist many lines of work for aspect and/or opinion terms extraction which can be categorized as rule-based, feature-engineering-based, or deep-learning-based approaches. For rule-based approaches (Hu and Liu 2004a; 2004b; Qiu et al. 2011), the idea is to manually design some rules based on syntactic or dependency structure of each sentence to expand the extracted aspect and opinion terms iteratively with a seed collection as input. For feature-engineering-based approaches, the idea is to train a classifier with rich, manual-defined features based on linguistic or syntactic information from annotated corpus to predict a label (aspect, opinion, or others) on each token in a sentence (Jin and Ho 2009; Li et al. 2010). These two categories of approaches are labor-intensive for constructing rules or features using linguistic and syntactic information. To reduce the engineering effort, deep-learning-based approaches (Liu, Joty, and Meng 2015; Yin et al. 2016; Wang et al. 2016) are proposed to learn high-level representations for each token, on which a classifier can be trained. Despite some promising results, most deep-learning approaches still require a parser analyzing the syntactic/dependency structure of the sentence to be encoded into the deep models. Therefore, the performances of these approaches rely on the quality of the parsing results.

In practice, the syntactic or dependency structures of many user-generated texts may not be precise with a computational parser, which may degrade the performances of existing deep-learning approaches. Moreover, performing parsing on a long sentence and large dataset can be very time-consuming. Therefore, we propose to use the attention mechanism (Bahdanau, Cho, and Bengio 2014) with tensor operators to replace the role of syntactic/dependency parsers to capture the relations among tokens in a sentence. Specifically, we design a couple of attentions, one for aspects extraction and the other for opinions extraction. They are learned interactively such that label information can be dually propagated among aspect terms and opinion terms by exploiting their relations. Moreover, we use multiple layers of the coupled attentions to extract inconspicuous aspect/opinion terms. Our motivation is similar to (Qiu et al. 2011; Wang et al. 2016) for exploiting aspect-opinion relations. The difference is that our model automatically learns these relations without any parsers or linguistic resources.

In summary, our contributions are two-fold: 1) We propose an end-to-end deep learning model for aspect and opinion terms co-extraction without requiring any syntactic/dependency parsers or linguistic resources to generate additional information as input. 2) We conduct extensive experiments on three benchmark datasets to verify that our model achieves state-of-the-art performance for aspect and opinion terms co-extraction.

Related Work

Aspect and Opinion Terms Extraction

For extracting aspect/opinion terms from texts, Hu and Liu (2004a) proposed to use association rule mining for extracting aspect terms and synonyms/antonyms from WordNet for identifying opinion terms. Qiu et al. (2011) used a dependency parser to augment a seed collection of aspect and opinion terms through double-propagation, similar for (Popescu and Etzioni 2005; Wu et al. 2009). The above methods are unsupervised, but depend on pre-defined rules and linguistic resources. For supervised methods, the task is treated as a sequence labeling problem. Li et al. (2010) and Jin and Ho (2009) implemented CRF and HMM with extensive human-designed features to solve the problem, respectively. Liu et al. (2012; 2013) applied a word alignment model in order to capture relations among opinion words, which requires large amount of training data to obtain desired relations. Topic models were also applied for aspect extraction (Chen, Mukherjee, and Liu 2014; Zhao et al. 2010). Recently, deep learning methods have been proposed for this task. Liu et al. (2015) applied recurrent neural network on top of pre-trained word embeddings for aspect extraction. Yin et al. (2016) proposed an unsupervised embedding method to encode dependency path into a recurrent neural network to learn high-level features for words, which are taken as input features for CRFs for aspect extraction. Wang et al. (2016) proposed a joint model of recursive neural networks and CRFs for aspect and opinion terms co-extraction. The neural network is constructed from the dependency parse tree to capture dual-propagation among aspect and opinion terms. Note that most existing deep models require a syntactic/denpendency parser and auxiliary linguistic features to boost their extraction accuracy. As a comparison, our proposed model does not need any linguistic features, or any pre-constructed syntactic structure as input.

Attention & Memory Network

Attentions (Mnih et al. 2014) and memory networks (Weston, Chopra, and Bordes 2015) have recently been used for various machine learning tasks, including image generation (Gregor et al. 2015), machine translation (Bahdanau, Cho, and Bengio 2014), sentence summarization (Rush, Chopra, and Weston 2015), document sentiment classification (Yang et al. 2016), and question answering (Hermann et al. 2015). The attention mechanism aims to select and attend to relevant parts of the input which could be thought of as a soft-alignment process. A memory network generally consists of multiple layers of attentions, which has shown superior performance in many NLP tasks (Kumar et al. 2016; Sukhbaatar et al. 2015). In this paper, we aim to develop a multi-layer attention network to replace the role of a syntactic/dependency parser to capture the relations among words in a sentence for information extraction.

Problem Statement & Motivation

We denote by s_i a review sentence from the training dataset, which consists of a sequence of tokens $s_i{=}\{w_{i1}, ..., w_{in_i}\}$. The task aims to extract a collection of all the explicit aspect terms $A_i = \{a_{i1}, ..., a_{ij}\}$ and opinion terms $P_i = \{p_{i1}, ..., p_{im}\}$ appearing in s_i. Note that a_{il} or p_{ir} could be a single word or a phrase. The task is modeled as a sequence tagging problem with the BIO encoding scheme. Specifically, we define 5 different classes: BA (beginning of aspect), IA (inside of aspect), BP (beginning of opinion), IP (inside of opinion), and O (others), and let $\mathrm{L} = \{BA, IA, BP, IP, O\}$. Each token $w_{ip} \in s_i$ is classified as $y_{ip} \in \mathrm{L}$. Given a test review $\overline{s}_j = \{\overline{w}_{j1}, ..., \overline{w}_{jn_j}\}$, we aim to obtain a prediction label $\overline{y}_{jq} \in \mathrm{L}$ for each $\overline{w}_{jq}$, where any prediction sequence with BA (BP) at the beginning followed by IA (IP) is extracted as a single aspect (opinion) term.

To fully exploit the syntactic relations among different tokens in a sentence, most existing methods applied a computational parser to analyze the syntactic/dependency structure of each sentence in advance. Figure 1 shows an example dependency structure of a review sentence. In this example, *fish burger* and *tastes* are ground truth aspect terms, accompanied with *best* and *fresh* as their opinions respectively. In (Qiu et al. 2011), several extraction rules are predefined based on the dependency structure. For instance, given *tastes* as an aspect term, *fresh* could be extracted as an opinion term through the direct relation: $A \xrightarrow{xcomp} B$. As another example, given *burger* as an aspect term, *tastes* can be extracted as another aspect term through the indirection relation: $A \xrightarrow{nsubj} C \xleftarrow{acl} B$ because they both have syntactic dependence on the same token *dish*. One major limitation of this rule-based approach is that it is deterministic, and thus may fail to handle uncertainty underlying the data. To address this issue, Wang et al. (2016) proposed to encode the dependency structure into a recursive neural network plugged with a CRF to construct syntactically meaningful and discriminative hidden representations.

Although promising results were shown in (Wang et al. 2016), a dependency parser is still required as a preprocessing step, and some simple feature engineering is also needed to boost its performance. However, there may be many grammar and syntactic errors in user-generated texts, in which case the outputs of a dependency parser may not be precise, and thus degrades the performance. Therefore, in this paper, we offer an end-to-end deep learning model, which models the relations among tokens automatically without any dependency parsing or feature engineering, and achieves state-of-the-art performances for aspect and opinion terms co-extraction.

Coupled Multi-layer Attentions

Our proposed model is named Coupled Multi-layer Attentions (CMLA) which consists of the following features:

- For each sentence, we construct a pair of attentions, one for aspect terms extraction, and the other for opinion terms extraction. Each attention aims to learn a prototype vector for aspect or opinion, a high-level feature vector for each token, and an attention score for each token in the sentence. The feature vector and attention score measure the extent of correlation between each input token and the prototype using a tensor operator, which captures different contexts of a given token when measuring its corre-

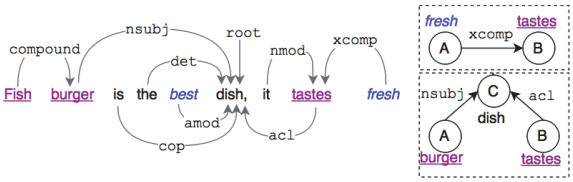


Figure 1: A dependency example for sentiment analysis.

lation to the prototype. Hence, a token with high score indicates a high chance of being an aspect or opinion.

- To capture direct relations between aspect and opinion terms, e.g., the $A \xrightarrow{xcomp} B$ relation shown in Figure 1, the pair of attentions are coupled in learning such that the learning of each attention is affected by the other. This helps to double-propagate information between them.
- To further capture indirect relations among aspect and opinion terms, e.g., the $A \xrightarrow{nsubj} C \xleftarrow{acl} B$ relation shown in Figure 1, we construct a network with multiple layers of coupled attentions.

Attention with Tensor Operator

A basic unit of CMLA is a pair of attentions: aspect attention and opinion attention. In most previous studies, attentions have been used for generating sentence- or document- level representation by computing a weighted sum of the input sequence (Bahdanau, Cho, and Bengio 2014). The weight of each input unit is an attention score obtained from its composition with a prototype vector which guides the model about where to attend. Different from previous approaches, we use attention to identify the possibility of each token being an aspect or opinion term. Figure 2(a) shows an example of a basic attention model for aspect extraction. We denote by $H=\{h_1, ..., h_n\}$ the input sequence of length n, where $h_i \in \mathbb{R}^d$ is the feature representation for the i-th token w_i.[1]

In the aspect attention, we first generate a prototype vector u^a for aspects which can be viewed as a general feature representation for aspect terms. This aspect prototype will guide the model to attend to the most relevant tokens.[2] Given u^a and H, the model scans the input sequence and computes an attention vector r_i^a and an attention score e_i^a for the i-th token. To obtain r_i^a, we first compute a composition vector $\beta_i^a \in \mathbb{R}^K$ that encodes the extent of correlations between h_i and prototype vector u^a through a tensor operator f^a:

$$\beta_i^a = f^a(h_i, u^a) = \tanh(h_i^\top G^a u^a), \quad (1)$$

where $G^a \in \mathbb{R}^{K \times d \times d}$ is a 3-dimensional tensor. Motivated by (Socher et al. 2013), a tensor operator could be viewed as multiple bilinear terms that could model more complicated compositions between 2 units. As shown in the bottom of Figure 2(a), G^a could be decomposed into K slices, where each slice $G_k^a \in \mathbb{R}^{d \times d}$ is a bilinear term that interacts with 2 vectors and captures one type of composition, e.g., a specific syntactic relation. Hence $h_i^\top G^a u^a \in \mathbb{R}^K$ inherits K different kinds of compositions between h_i and u^a that indicates complicated correlations between each input token and the aspect prototype. By adding a non-linear transformation $\tanh(\cdot)$, β_i^a encodes more abstract and high-level correlation features. Then r_i^a is obtained from β_i^a via a GRU network:

$$r_i^a = (1 - z_i^a) \odot r_{i-1}^a + z_i^a \odot \widetilde{r}_i^a, \quad (2)$$

where $g_i^a = \sigma(W_g^a r_{i-1}^a + U_g^a \beta_i^a)$, $z_i^a = \sigma(W_z^a r_{i-1}^a + U_z^a \beta_i^a)$, and $\widetilde{r}_i^a = \tanh(W_r^a (g_i^a \odot r_{i-1}^a) + U_r^a \beta_i^a)$. Here, g_i^a and z_i^a are reset and update gates respectively that control the information flow from the previous timestamp. W_g^a, U_g^a, W_z^a and U_z^a are weight matrices to be learned for transforming r_{i-1}^a and β_i^a to gate units. By applying GRU on β_i^a, the attention vector $r_i^a \in \mathbb{R}^K$ becomes context-dependent with the ability to inherit past information. For example, as shown in Figure 2(a), if *Fish* has high correlations with aspect prototype, its next token *burger* also has high chance of being active, because r_2^a inherits information from r_1^a. Indeed, many aspect terms consist of multiple tokens, and exploiting context information helps their predictions. For simplicity, we use $r_i^a = \text{GRU}(f^a(h_i, u^a), \theta^a)$ to denote (2), where $\theta^a = \{W_g^a, U_g^a, W_z^a, U_z^a, W_r^a, U_r^a\}$.

An attention score e_i^a for token w_i is then computed as

$$e_i^a = {v^a}^\top r_i^a. \quad (3)$$

Since r_i^a is a correlation feature vector, $v^a \in \mathbb{R}^K$ can be deemed as a weight vector that weighs each feature accordingly. Hence, e_i^a becomes a scalar score, where a higher score indicates higher correlation with the prototype, and higher chance of being attended. For example, as shown in Figure 2(a), u^a helps the model to attend to *Fish* and *burger* which indicates their high chance of being aspect terms. Note that the output attention vector r_i^a is also used as the final feature representation for w_i. Thus, a prediction on each token can be generated by $l_i^a = \text{softmax}(C^a r_i^a)$, where $C^a \in \mathbb{R}^{c \times K}$ is a classification matrix for converting final feature vectors to labels, and c is the number of classes.[3]

The procedure for opinion attention is similar. In the subsequent sections, we use a superscript p to denote the opinion attention. In the final prediction, each token only belongs to 1 of the 5 classes in L mentioned previously. After l_i^a and l_i^p are obtained for each token, we pick the largest value from each vector. If both of them correspond to O, then the final prediction is O. If only one of them is O, we pick the other one as final prediction. When neither of them are O, the two values are compared and the largest one is chosen.

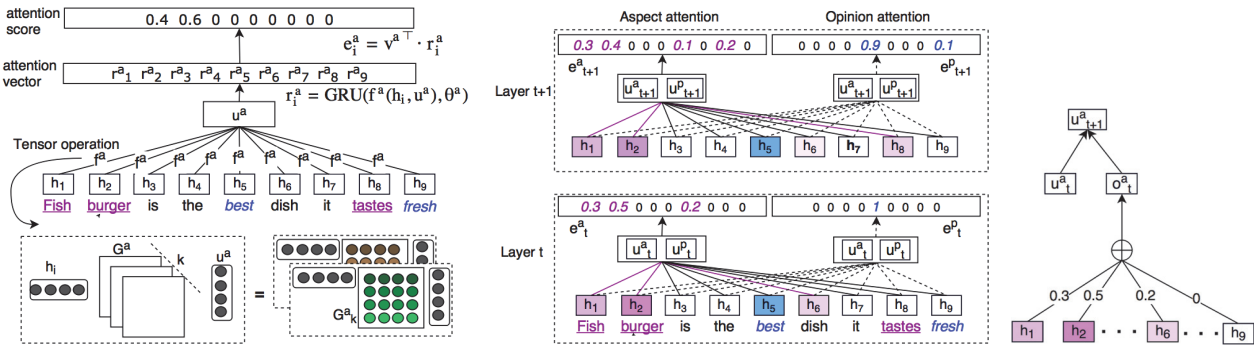
Coupled Attentions for Dual Propagation

As discussed in previous sections, a crucial issue for co-extraction of aspect and opinion terms is how to fully exploit the relations between aspect terms and opinion terms

[1] For initialization of h_i, we first pre-train a word embedding $x_i \in \mathbb{R}^D$ (Mikolov et al. 2013) for w_i, and then apply Gated Recurrent Unit (GRU) (Cho et al. 2014) to obtain h_i by encoding context information.

[2] We randomly initialize u^a from a uniform distribution: $u^a \sim U[-0.2, 0.2] \in \mathbb{R}^d$, which is then trained and updated iteratively.

[3] Here, $c=3$. Classes in the aspect attention are BA, IA and O, while classes in the opinion attention are BP, IP and O.



(a) A single-layer attention model with tensor. (b) Multi-layer Coupled attentions. (c) Attention prototype.

Figure 2: Illustration of the proposed model.

such that the information can be propagated to each other to assist final predictions. However, independent learning of the aspect or opinion attention fails to utilize their relations. Therefore, we propose to couple the learning of the two attentions such that information of each attention can be dually propagated to the other. Specifically, as shown in Figure 2(b), solid lines and dashed lines denote aspect attention and opinion attention, respectively. The two attentions share the same feature vector h_i for each input token w_i. Different from a single attention, the prototype to be fed into each attention module becomes a pair of vectors $\{u^a, u^p\}$, and the tensor operator in (1) becomes a pair of tensors $\{G^m, D^m\}$:

$$f^m(h_i, u^a, u^p) = \tanh([h_i^\top G^m u^m : h_i^\top D^m u^{\overline{m}}]), \quad (4)$$

where $[:]$ denotes concatenation of vectors, and $m \in \{a, p\}$ is the index of the two attentions, $\overline{m} = a$ if $m = p$, and $\overline{m} = p$ if $m = a$. The new tensor $D^m \in \mathbb{R}^{K \times d \times d}$ is used to model the correlations of h_i with the prototype $u^{\overline{m}}$ from the conjugate attention, which captures the dual-propagation between aspect terms and opinion terms. For example, if h_8 for *tastes* is already attended through the aspect attention and incorporated in u^a, it will help to attend *fresh* for opinion attention due to its strong correlation with *tastes*. This indicates *fresh* as a possible opinion term. Similar to (2), the outputs r_i^m and e_i^m are obtained through

$$r_i^m = \text{GRU}(f^m(h_i, u^a, u^p), \theta^m), \text{ and } e_i^m = {v^m}^\top r_i^m. \quad (5)$$

Multi-Layer Coupled Attentions

A couple of attentions is only able to capture the direct relations between aspect terms and opinion terms, but not the indirect relations among them, such as the $A \xrightarrow{nsubj} C \xleftarrow{acl} B$ relation shown in Figure 1. To address this issue, we propose a network with multi-layer coupled attentions. Specifically, we present an example consisting of two layers in Figure 2(b), where each layer consists of coupled attentions as illustrated in the previous section. For each layer $t+1$ as shown in Figure 2(c), the prototype vectors u_{t+1}^m, where $m \in \{a, p\}$, are updated based on the prototype vectors in the previous layer u_t^m to incorporate more feasible representations for aspect or opinion terms through

$$u_{t+1}^m = \tanh(V^m u_t^m) + o_t^m, \quad (6)$$

where $V^m \in \mathbb{R}^{d \times d}$ is a recurrent transformation matrix to be learned, and o_t^m is an accumulated vector computed via

$$o_t^m = \sum_{i=1}^{n} \alpha_{ti}^m h_i, \text{ and } \alpha_{ti}^m = \exp(e_{ti}^m) / \sum_{j}^{n} \exp(e_{tj}^m), \quad (7)$$

where α_{ti}^m is a normalized attention score for e_{ti}^m. Intuitively, o_t^m is dominated by the input feature vectors $\{h_i\}$'s with higher attention scores. Therefore, o_t^m will approach to the attended feature vectors of aspect or opinion tokens. As a result, u_{t+1}^m will capture more accurate feature representation about aspect or opinion terms, which in return is used to guide the model about where to attend in the next layer.

We use Figure 2(b) to illustrate how the multi-layer coupled attentions model can capture indirect relations, e.g., the $A \xrightarrow{nsubj} C \xleftarrow{acl} B$ relation. Suppose at layer t, u_t^a incorporates h_1 and h_2 for *Fish* and *burger*, u_t^p incorporates h_5 for *best*. For the aspect attention, $\{u_t^a, u_t^p\}$ interact with each h_i to obtain the score e_{ti}^a. We see that *dish* is attended because h_6 is highly correlated with both h_2 and h_5. As a result, u_{t+1}^a will be updated, and incorporate h_6, which in turn assists focusing attention on *tastes* in the next layer, because of the strong correlation between h_6 and h_8. In this case, the aspect term *tastes* is extracted indirectly through two layers of the coupled attentions. This shows that the multi-layer attention network is able to progressively attend the aspect or opinion words that are non-obvious and have indirect relations.

Similar to the single-layer coupled attention model, the proposed network first accumulates high-level representations r_{ti}^m in (5) for each token i at each layer t to generate the prediction vectors $l_i^m = \text{softmax}(C^m \sum_{t=1}^{T} r_{ti}^m)$, and then outputs a final prediction for each token.

Experiments

Datasets & Experimental Setup

We evaluate and compare our proposed model on three benchmark datasets, as described in Table 1. They are taken from SemEval Challenge 2014 task 4 subtask 1 (Pontiki et al. 2014) and SemEval Challenge 2015 task 12 subtask 1 (Pontiki et al. 2015). Note that the original datasets in the challenges only contain labels for aspect terms. For S1 and

Dataset	Description	Training	Test	Total
S1	SemEval-14 Restaurant	3,041	800	3,841
S2	SemEval-14 Laptop	3,045	800	3,845
S3	SemEval-15 Restaurant	1,315	685	2,000

Table 1: Dataset description with number of sentences

S2, we use the labels on opinion terms provided by (Wang et al. 2016), and manually label all the opinion terms for S3.

The pre-trained word embeddings are obtained using the word2vec tool[4] on two different corpora, as the three datasets belong to two domains: restaurant and laptop. Following the setup in (Wang et al. 2016), for restaurant domain, we apply word2vec on Yelp Challenge dataset[5] consisting of 2.2M restaurant reviews with 54K vocabulary size. For laptop domain, we use the corpus from electronic domain in Amazon reviews (McAuley et al. 2015), which contains 1M reviews with 590K vocabulary size. The dimensions of word embeddings are 200 for restaurant domain and 150 for laptop domain in our experiments.

For the input feature vectors to the attention network, we convert the pre-trained word embeddings to hidden representations through GRU implemented with the Theano library.[6] The size of the hidden units for each layer is 50 for all three datasets. We use a 2-layer attention network for experiments. For each layer, the first dimension K of tensors is set to be 20 for S1 and S3 (15 for S2).We use a fixed learning rate for all experiments: 0.07 for S1, S3, and 0.1 for S2. To avoid overfitting, the network is regularized with dropout. We follow the idea of (Zaremba, Sutskever, and Vinyals 2014) which shows that partial dropout (only apply dropout to non-recurrent parameters) is better than applying dropout to all parameters for RNN. The dropout rate is set to be 0.5 for non-recurrent parameters of GRU. Note that all the above parameters are chosen through cross-validation.

Experimental Results

We compare CMLA with the following baseline models:

- DLIREC, IHS_RD, EliXa: the top performing systems for S1, S2 in SemEval Challenge 2014, and S3 in SemEval Challenge 2015, respectively.
- LSTM: an LSTM network built on top of word embeddings proposed by (Liu, Joty, and Meng 2015). The settings are the same as (Wang et al. 2016).
- WDEmb: the model proposed by (Yin et al. 2016) using word and dependency path embeddings combined with linear context embedding features, dependency context embedding features as CRF input.[7]
- RNCRF: the joint model with CRF and recursive neural network proposed by (Wang et al. 2016), which has been shown to outperform CRFs with hand-crafted features.

[4]https://radimrehurek.com/gensim/models/word2vec.html

[5]http://www.yelp.com/dataset_challenge

[6]http://deeplearning.net/software/theano/

[7]We report the original result from (Yin et al. 2016) as the source code is not available.

	S1		S2		S3	
Model	AS	OP	AS	OP	AS	OP
DLIREC	84.01	-	73.78	-	-	-
IHS_RD	79.62	-	74.55	-	-	-
EliXa	-	-	-	-	70.04	-
LSTM	81.15	80.22	72.73	74.98	64.30	66.43
WDEmb	84.31	-	74.68	-	69.12	-
WDEmb*	84.97	-	75.16	-	69.73	-
RNCRF	84.05	80.93	76.83	76.76	67.06	66.90
RNCRF*	84.93	**84.11**	**78.42**	79.44	67.74	67.62
CMLA	**85.29**	83.18	77.80	**80.17**	**70.73**	**73.68**

Table 2: Comparison results in terms of F_1 scores. AS (OS) refers to aspect (opinion) terms extraction.

- WDEmb*, RNCRF*: the corresponding models with additional human-engineered linguistic features.

The comparison results in terms of F_1 scores are shown in Table 2. We report results for both aspect terms extraction (AS) and opinion terms extraction (OP) for all the three datasets. To make fair comparisons, we use the same corpus as in LSTM, RNCRF, RNCRF* for training word embeddings, and same training set with both aspect and opinion labels. Among deep-learning-based models, the models that combine neural network with CRF (i.e., WDEmb and RNCRF) perform better than LSTM because of the incorporation of dependency structure. It is clear that CMLA achieves the state-of-the-art results for most of the time without any pre-extracted linguistic/syntactic information. Specifically, CMLA outperforms WDEmb by 0.98%, 3.12% and 1.61%, and RNCRF by 1.24%, 0.97% and 3.67% for aspect extraction on S1, S2 and S3, respectively. Even compared with the deep models with additional hand-crafted features, i.e., WDEmb* and RNCRF*, CMLA still gets 0.32%, 2.64% and 1.00% improvement over WDEmb* for aspect extraction on S1, S2 and S3, and 0.36% and 2.99% increase over RNCRF* for aspect extraction on S1 and S3, respectively. Moreover, the improvements over RNCRF and RNCRF* are all significant ($p < 0.01$), except for the aspects extraction on S1 and S2 over RNCRF*. Note that besides linguistic features, WDEmb* and RNCRF* also require dependency parsers to perform the task. Therefore, CMLA is more effective and simpler to implement.

To show the effect of the number of layers, we present experimental results varying the number of layers in Table 4. The best results are obtained with 2 layers. With only one layer, the results for aspect extraction are 0.39%, 0.52% and 1.46% inferior than the best scores on S1, S2 and S3, respectively, but they are still comparable with other baselines shown in Table 2. Similar observations can be found for the results with 3 layers. This shows that CMLA with 2 layers is enough to exploit most of the relations among input tokens.

We also conducted experiments to explicitly show the advantage of coupling the learning of aspect and opinion attentions. The second part in Table 4 specifies different setups of the model. ASL refers to the multi-layer network with only aspect attention and is trained with aspect labels only. We can see that even without opinion labels, the network still proves comparable and even superior than deep models

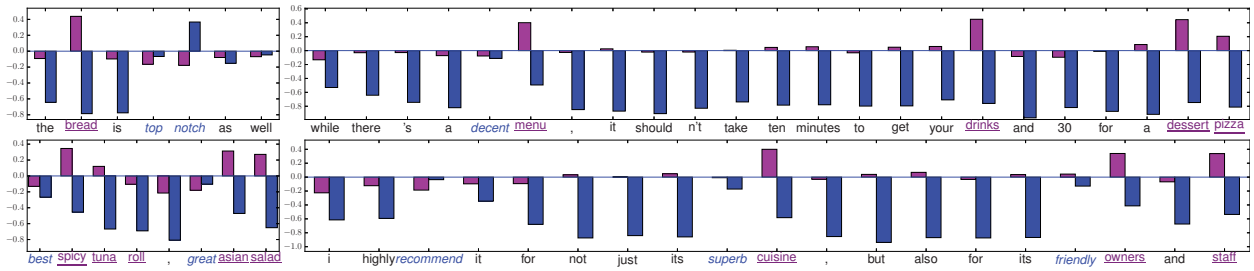


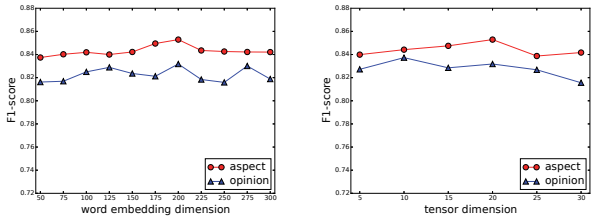
Figure 3: Visualization of attention weights for different tokens within a sequence.

Prediction with CMLA	**Prediction with RNCRF**
also *stunning* "colors" and *speedy*	also *stunning colors* and speedy
Only 2 "usb ports" ... seems kind of *limited*	Only 2 "usb ports" ... seems kind of limited
strong "build" though which really adds to its "durability"	*strong* "build" though which really adds to its durability
Save room for "deserts" - they're to *die for*	Save room for "deserts" - they're to die for
You must try "Odessa stew" or "Rabbit stew"; "salads" - all *good*	You must try "Odessa stew or Rabbit stew"; salads - all good

Table 3: Prediction comparison between CMLA and RNCRF

		S1		S2		S3	
		AS	OP	AS	OP	AS	OP
Layer	1	84.90	81.85	77.28	78.12	69.27	69.56
	2	**85.29**	**83.18**	**77.80**	**80.17**	**70.73**	**73.68**
	3	84.41	82.38	77.24	79.29	69.78	71.95
Setup	ASL	84.38	-	76.45	-	69.53	-
	ASL+OPL	84.14	82.10	77.05	79.66	69.49	72.73
	CMLA	**85.29**	**83.18**	**77.80**	**80.17**	**70.73**	**73.68**

Table 4: Comparisons under varying layers and setups.



(a) On word embedding. (b) On tensor interaction.

Figure 4: Sensitivity studies for data S1.

without linguistic features for aspect terms extraction shown in Table 2. This shows that multi-layer attentions with tensors is advantageous for exploiting interactions. ASL+OPL in Table 4 trains the aspect attention and opinion attention independently using (1) where each attention predicts one of the three labels. The results of ASL+OPL in terms of aspect extraction are similar to ASL, which shows that the additional opinion labels have little effect on aspect extraction if they are not interactively trained. By coupling the aspect and opinion attentions, CMLA achieves the best performance.

As a core component, an attention computes a score for each token to indicate its correlation with the corresponding prototype. We visualize the actual attention scores for the tokens of 4 sentences in Figure 3. The y-axis represents the scores before normalization which can be positive or negative, but only the magnitude matters. Higher scores mean larger correlations with the aspect/opinion prototype. As the aspect and opinion attention have different sets of parameters, the scores can correspond to different ranges of the values. Tokens in purple (blue) are the ground-truth aspect (opinion) terms. Obviously, purple tokens correspond to large scores for aspect extraction (purple bars with large values), and blue tokens correspond to large scores for opinion extraction (blue bars with large values). All the other non-relevant terms have lower scores. This shows that our model is able to extract terms of interest.

As mentioned previously, CMLA is able to extract target terms without any dependency parser, and hence does not depend on the quality of the parsing results. To show that, we pick a few example reviews from the test datasets as presented in Table 3. The left and right column show the prediction results from the proposed model and RNCRF (Wang et al. 2016), respectively, where predicted opinions are made *italic*, and aspects are "quoted". Obviously, the listed reviews are not formal enough to be parsed correctly. Hence, RNCRF fails to extract some of the targets, unlike CMLA which identifies all possible target terms.

To show the robustness of CMLA, we provide two sensitivity studies on word embedding dimensions and the number of different interactions within a 3-dimensional tensor on S1 in Figure 4. From the plot, we can see that the performances for both aspect and opinion terms extraction are relatively stable when varying word embedding dimensions, with the highest scores achieved at 200. For the number of tensor interactions, the model attains the best performance at 20 for aspect extraction and 10 for opinion extraction.

Conclusion

We present a novel end-to-end network with coupled multi-layer attentions, CMLA, for aspect-opinion co-extraction, which does not require any parsers or linguistic resources. Different from traditional attention network, we propose coupled attentions to exploit the correlations among input tokens, especially between aspect and opinion terms, through tensor operators. Moreover, the multi-layer structure helps to extract non-obvious targets with indirect relations. Experimental results on 3 benchmark datasets verify the effectiveness of CMLA.

Acknowledgements

This research is partially funded by the Economic Development Board and the National Research Foundation of Singapore. Sinno J. Pan thanks the support from the NTU Singapore Nanyang Assistant Professorship (NAP) grant M4081532.020.

References

Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate. In *CoRR abs/1409.0473*.

Chen, Z.; Mukherjee, A.; and Liu, B. 2014. Aspect extraction with automated prior knowledge learning. In *ACL*, 347–358.

Cho, K.; van Merrienboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *EMNLP*, 1724–1734.

Gregor, K.; Danihelka, I.; Graves, A.; Rezende, D. J.; and Wierstra, D. 2015. DRAW: A recurrent neural network for image generation. In *ICML*, 1462–1471.

Hermann, K. M.; Kočiský, T.; Grefenstette, E.; Espeholt, L.; Kay, W.; Suleyman, M.; and Blunsom, P. 2015. Teaching machines to read and comprehend. In *NIPS*.

Hu, M., and Liu, B. 2004a. Mining and summarizing customer reviews. In *KDD*, 168–177.

Hu, M., and Liu, B. 2004b. Mining opinion features in customer reviews. In *AAAI*, 755–760.

Jin, W., and Ho, H. H. 2009. A novel lexicalized hmm-based learning framework for web opinion mining. In *ICML*, 465–472.

Kumar, A.; Irsoy, O.; Ondruska, P.; Iyyer, M.; James Bradbury, I. G.; Zhong, V.; Paulus, R.; and Socher, R. 2016. Ask me anything: Dynamic memory networks for natural language processing. In *ICML*.

Li, F.; Han, C.; Huang, M.; Zhu, X.; Xia, Y.-J.; Zhang, S.; and Yu, H. 2010. Structure-aware review mining and summarization. In *COLING*, 653–661.

Liu, K.; Xu, L.; Liu, Y.; and Zhao, J. 2013. Opinion target extraction using partially-supervised word alignment model. In *IJCAI*, 2134–2140.

Liu, P.; Joty, S.; and Meng, H. 2015. Fine-grained opinion mining with recurrent neural networks and word embeddings. In *EMNLP*, 1433–1443.

Liu, K.; Xu, L.; and Zhao, J. 2012. Opinion target extraction using word-based translation model. In *EMNLP-CoNLL*, 1346–1356.

McAuley, J.; Targett, C.; Shi, Q.; and van den Hengel, A. 2015. Image-based recommendations on styles and substitutes. In *SIGIR*, 43–52.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*, 3111–3119.

Mnih, V.; Heess, N.; Graves, A.; and Kavukcuoglu, K. 2014. Recurrent models of visual attention. In *NIPS*. 2204–2212.

Pang, B., and Lee, L. 2008. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval* 2(1-2).

Pontiki, M.; Galanis, D.; Pavlopoulos, J.; Papageorgiou, H.; Androutsopoulos, I.; and Manandhar, S. 2014. Semeval-2014 task 4: Aspect based sentiment analysis. In *SemEval*, 27–35.

Pontiki, M.; Galanis, D.; Papageorgiou, H.; Manandhar, S.; and Androutsopoulos, I. 2015. Semeval-2015 task 12: Aspect based sentiment analysis. In *SemEval*, 486–495.

Popescu, A.-M., and Etzioni, O. 2005. Extracting product features and opinions from reviews. In *EMNLP*, 339–346.

Qiu, G.; Liu, B.; Bu, J.; and Chen, C. 2011. Opinion word expansion and target extraction through double propagation. *Comput. Linguist.* 37(1):9–27.

Rush, A. M.; Chopra, S.; and Weston, J. 2015. A neural attention model for abstractive sentence summarization. In *EMNLP*, 379–389.

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

Sukhbaatar, S.; Szlam, A.; Weston, J.; and Fergus, R. 2015. End-to-end memory networks. In *NIPS*, 2440–2448.

Wang, W.; Pan, S. J.; Dahlmeier, D.; and Xiao, X. 2016. Recursive neural conditional random fields for aspect-based sentiment analysis. In *EMNLP*.

Weston, J.; Chopra, S.; and Bordes, A. 2015. Memory networks. In *ICLR*.

Wu, Y.; Zhang, Q.; Huang, X.; and Wu, L. 2009. Phrase dependency parsing for opinion mining. In *EMNLP*, 1533–1541.

Yang, Z.; Yang, D.; Dyer, C.; He, X.; Smola, A.; and Hovy, E. 2016. Hierarchical attention networks for document classification. In *NAACL*, 1480–1489.

Yin, Y.; Wei, F.; Dong, L.; Xu, K.; Zhang, M.; and Zhou, M. 2016. Unsupervised word and dependency path embeddings for aspect term extraction. In *IJCAI*.

Zaremba, W.; Sutskever, I.; and Vinyals, O. 2014. Recurrent neural network regularization. *CoRR* abs/1409.2329.

Zhao, W. X.; Jiang, J.; Yan, H.; and Li, X. 2010. Jointly modeling aspects and opinions with a maxent-lda hybrid. In *EMNLP*, 56–65.