Neural Process Lett (2017) 46:581–594
DOI 10.1007/s11063-017-9605-7

Cascaded Convolutional Neural Networks for Aspect-Based Opinion Summary

Xiaodong Gu[1] · **Yiwei Gu**[1] · **Haibing Wu**[1]

Published online: 1 March 2017

Abstract This paper studies aspect-based opinion summary (AOS) of reviews on particular products. In practice, an AOS system needs to address two core subtasks, aspect extraction and sentiment classification. Most existing approaches to aspect extraction, using linguistic analysis or topic modeling, are general across different products but not precise enough or suitable for particular products. Instead we take a less general but more precise scheme, which directly maps each review sentence into pre-defined aspects. To tackle aspect mapping and sentiment classification, we propose a convolutional neural network (CNN) based method, cascaded CNN (C-CNN). C-CNN contains two levels of convolutional networks. Multiple CNNs at level 1 deal with aspect mapping task. If a review sentence belongs to pre-defined aspect categories, a single CNN at level 2 determines its sentiment polarity. Experimental results show that C-CNN with pre-trained word embedding outperform cascaded SVM with feature engineering. We also build a system called OpiSum with C-CNN. The demo of OpiSum can be found at http://114.215.167.42.

Keywords Aspect-based opinion summary · Sentiment classification · Convolutional neural networks · Data mining

1 Introduction

Analysis of data is a process of modeling data with the goal of discovering useful information to support decision-making. Various data analysis techniques are widely used in many research fields, such as analysis of time series data in order to extract meaningful statistics and other characteristics [1–4], analysis of text to extract and classify information from textural sources. This paper focuses on text analysis especially on analysis of online reviews. User generated reviews on products are expanding rapidly with the emergence and advance-

✉ Xiaodong Gu
xdgu@fudan.edu.cn

[1] Department of Electronic Engineering, Fudan University, Shanghai 200433, China

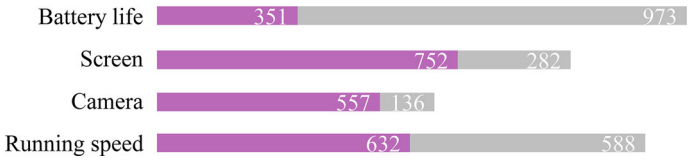

Fig. 1 An example aspect-based summary of smartphone reviews

ment of e-commerce. These reviews are valuable to business organizations for improving their products and to individual consumers for making informed decisions. Unfortunately, reading through all the product reviews is hard, especially for popular products with large volume of review texts. It is therefore essential to provide coherent and concise summaries of user generated reviews. This has bred a new line of data mining research on aspect-based opinion summary (AOS) [5]. Given a set of product reviews, an AOS system extracts aspects discussed in the reviews and predicts reviewers' sentiments toward these aspects. Figure 1 presents an example summary of smartphone reviews. The smartphone aspects, such as battery life and screen, with the hyperlinks and numbers of positive and negative opinions, are illustrated in a structured way. The goal of this paper is to generate such aspect-based opinion summary.

Standard AOS typically involves two component subtasks, aspect extraction and sentiment classification. Aspect extraction finds related aspects and extracts all textual mentions associated with each aspect. Sentiment classification task classifies sentiment over each aspect using the associated textual mentions.

Existing researches on aspect extraction move along two quite different lines. The first extracts aspect expressions using linguistic patterns or supervised sequence labeling (see Sect. 2). This scheme is very limited for only identifying explicit aspects and failing to handle implicit aspects. Besides, it needs additional efforts to group synonymous aspect expressions into the same category. The second is based on topic modeling (see Sect. 2). Topic modeling is fully unsupervised, saving the labeling of training data. It handles implicit aspects well, and simultaneously extracts and groups aspects. It is, however, not suitable for summarizing reviews on particular products in many respects. The unsupervised nature makes it more general across different products, but less precise for particular products compared to supervised learning methods. The learned topics of topic modeling are implicit and often do not correlate well with human judgments, making it not applicable if users care about some particular product aspects. Topic modeling categorizes aspects, but its unsupervised nature makes the grouping not controllable or adaptable. Categorizing aspects is subjective because for different applications the user may need different categorizations. For example, in smartphone reviews, front camera and back camera can be regarded as two separate aspects, but can also be one general aspect, camera.

For some vertical e-commerce websites that focus on particular products, users already know what aspects a product has. Ontologies negates the need for identifying aspects automatically. Herein the most pressing challenge is to extract all relevant text mentions for each aspect. Therefore, this paper takes a line different from prior work on aspect extraction: directly mapping each review sentence into pre-defined aspect categories. That is, we formulate aspect extraction as sentence-level aspect mapping (or classification) problem. This scheme extracts relevant text mentions for pre-defined aspects and enjoys a lot of advantages. It handles both explicit and implicit aspects, and simultaneously extracts and categorizes different aspect expressions into the same aspect category. It also enables users to design different aspect categories for different application purposes.

Besides aspect extraction, sentiment classification is also necessary to enable real applications. This paper presents an aspect-based summary system which addresses both tasks. Most previous work on AOS deals with a single task, either aspect extraction or sentiment classification, using traditional machine learning. Motivated by the recent success of deep Convolutional Neural Network (CNN), we propose a CNN-based approach to jointly tackle aspect mapping and sentiment classification problems. The method is a two-level Cascaded CNN (C-CNN). At level 1, multiple convolutional networks map the input sentences into pre-defined aspects. If a review sentence belongs to pre-defined aspect categories, a single convolutional network at level 2 predicts the sentiment polarities of the input sentences. As no such benchmark corpus involving a collection of sentences labeled with pre-defined aspects and sentiments can be found, we create two datasets detailed in Sect. 4 especially for such aspect-based opinion summarization tasks. Empirical results show that C-CNN with pre-trained word embedding representation outperform cascaded SVM with feature engineering. With the proposed C-CNN, we also build a system called OpiSum, which can generate aspect-based summary like Fig. 1.

2 Related Work

AOS has attracted a lot of attentions with the advent of online user generated reviews [6–11]. Deep learning and representation learning, initially enjoying great success in computer vision, have also achieved some success in Natural Language Processing (NLP).

An AOS system needs to address two core tasks, aspect extraction and sentiment classification. One line of work on aspect extraction detects aspect expressions using linguistic patterns (e.g. part-of-speech and dependency relations) [5,12–15] or supervised sequence labeling such as CRFs [10,16–19]. This scheme is very limited in many respects. It only extracts explicit aspect expressions, and cannot deal with implicit aspects well. For example, in the sentence "this phone runs smoothly and fast, but its battery life is very poor", battery life is explicitly mentioned, while running speed is implicitly mentioned and thus cannot directly discovered using linguistic patterns or sequence labeling. This scheme is also limited for not grouping aspect expressions into aspect categories. For example, screen, display and retina refer to the same aspect for iPhone. After extracting all aspect expressions, additional efforts are required to categorize domain synonyms into the same aspect.

Another line of related work applies variants of standard topic modeling such as LDA [9–11,20–26]. Topic modeling deals with implicit aspects to some degree, and simultaneously extracts and groups aspects. However, it often learns incoherent topics since its objective functions do not always correlate well with human judgments. Compared with supervised methods, unsupervised topic modeling is more general across different products, but less precise for particular products. In addition, mapping from topics to aspects is not explicit, making it not a good choice if users care about opinions on some particular aspects. Topic modeling categories aspects based on co-occurrence counts. However, categorizing aspects is subjective because for different applications the user may need different categorizations. For example, in smartphone reviews, front camera and back camera can be treated as two different aspects, but can also be only one aspect. The unsupervised nature of topic modeling makes the grouping not controllable or adaptable.

An AOS system also involves sentiment classification. This task aims to classify an opinionated review as expressing positive or negative sentiment over an aspect. Compared to aspect extraction, sentiment classification was studied earlier and more extensively. Most prior work used traditional machine learning with complicated feature engineering [27–34].

Very recently, some researchers applied deep convolutional neural networks to sentence sentiment classification and reported considerably better results than traditional approaches [23,35,36].

In practice, much work has been devoted to perform aspect-based opinion summarization (or sentiment analysis) as a joint system. And topic model is a widely used method. Joint Sentiment/Topic model (JST) [37] is a flat topic model based on LDA. For each polarity, a flat mixture of topics is associated with it and all the words with the polarity are generated from this mixture. The drawback of JST is that finding the different polarities for the same topic is difficult. Reverse JST (RJST) reverses the association direction between topics and polarities in JST. RJST makes it more convenient to find the different polarities for the same topic, but performs poorly on document-level sentiment analysis. More works within topic modelling framework can be found such as Seeded Aspect and Sentiment Model, Multi-grain Topic Model, etc. These topic model based system are fully unsupervised or weakly supervised, and suffered from the drawbacks mentioned above. Also there has been a great deal of research into discovering both aspects and the related sentiments outside of the topic modeling framework. A machine learning framework was proposed by Jin, Ho, and Srihari [38] to discover aspects and sentimental polarities related to each aspect. Their lexicalized HMMs based framework naturally integrates multiple linguistic features (e.g., part-of-speech, phrases' internal formation patterns, and surrounding contextual clues of words/phrases) into automatic learning of potential product entities and opinion orientations. The experimental results demonstrated the effectiveness of their approach. While according to the analysis delivered above, this linguistic patterns based machine learning framework is limited in extracting implicit aspects and grouping similar aspects expressions into aspect categories. Approaches and frameworks mentioned above don't consider the usage of deep learning techniques, e.g., Convolutional neural network (CNN) which is proved to be an effective approach in many traditional NLP tasks. Limited work that use deep learning for aspect-based opinion summarization can be found. Most of them use CNN only in sentiment analysis but not the whole AOS system.

Convolutional neural network (CNN) is currently underpinning the cutting edge in computer vision [39,40]. It has also achieved state-of-the-art results in many traditional NLP tasks [41] and other NLP areas such as information retrieval [42,43] and relation classification [44,45]. Words are encoded as low-dimensional word vectors in CNN, instead of high dimensional one-hot representations. Word vector representations capture semantic information, so semantically close words are likewise close in low dimensional vector space. CNN models for specific NLP tasks often use unsupervised pre-trained word vectors [46] as initialization, which are then improved by optimizing specific supervised objectives.

3 Methodology

3.1 System Overview

An architectural overview of our aspect-based summary system is given in Fig. 2. The input to the system is a set of crawled reviews for a particular product. The sentence segmenter divides review texts into a set of sentences. The aspect mapper maps these sentences into pre-defined aspect categories. In this step only sentences belonging to the pre-defined aspects are extracted and retained. The sentiment classifier then predicts the polarity of each of these extracted sentences as positive or negative. After annotating each sentence with aspect and

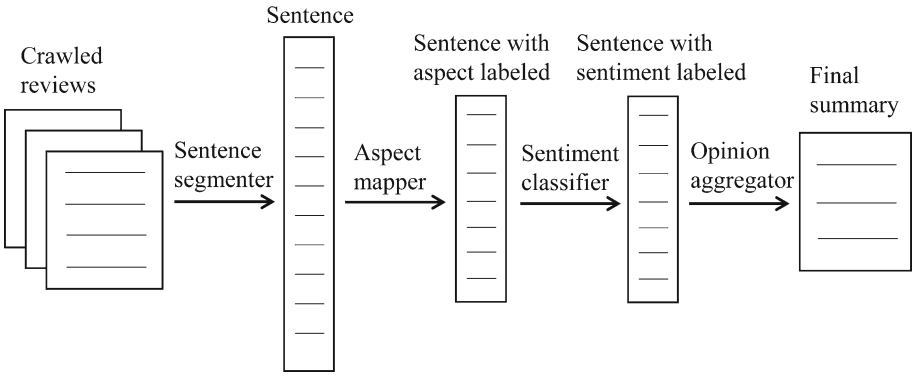

Fig. 2 An overview of our aspect-based opinion summary system

sentiment, the final opinion aggregator counts the number of positive and negative opinionated sentences corresponding to each aspect, and gives the hyperlinks to these sentences.

3.2 Cascaded CNN

The architecture of our C-CNN is shown is Fig. 3. The network contains C CNN aspect mappers and a CNN sentiment classifier. Aspect-mapping CNN and sentiment-classification CNN are organized in a cascaded way. Each mapper determines whether the input sentence belongs to its corresponding aspect. If that is the case, the sentiment classifier predicts sentiment polarity as positive or negative.

We address two considerations about the cascaded network. (1) The network only contains one sentiment classifier. One may think it is problematic as a single sentence (e.g. "This phone runs fast, but loses its charge too quickly!") could contain different aspects, and sentiments towards these aspects could be opposite. We do not train a separate sentiment classifier for each aspect category since in practice only a few sentences imply opposite sentiments for different aspects. (2) The sentiment classifier only deals with sentences belonging to at least one pre-defined aspect categories as practical applications only care the sentiments of aspect related sentences. In addition, sentences not belonging to any pre-defined aspect could be objective. It is not suitable classifying the sentiments of objective sentences as positive or negative.

Each CNN contains a word embedding layer, a convolutional and pooling layer, and a fully-connected layer.

Word embedding. This layer encodes each word in the input sentence as a word vector. Let be the sentence length, $|\boldsymbol{D}| \in R$ be the vocabulary size and $\boldsymbol{W}^{(1)} \in R^{k \times |\boldsymbol{D}|}$ be the embedding matrix of k-dimensional word vectors. The i-th word in a sentence is transformed into a k-dimensional vector $\boldsymbol{w}_i$ by matrix-vector product:

$$\boldsymbol{w}_i = \boldsymbol{W}^{(1)} \boldsymbol{x}_i \tag{1}$$

Here xi is the one-hot $\boldsymbol{x}_i$ representation for the i-th word.

Convolution. After encoding the input sentence with word vectors, the convolution operations are applied on top of these vectors to produce new features. A convolution operation involves a filter $\boldsymbol{u} \in R^{hk}$ applied to a window of $h = 2r + 1$ words. For example, a feature f_i is produced from a window of words $\boldsymbol{w}_{i-r:i+r}$ by

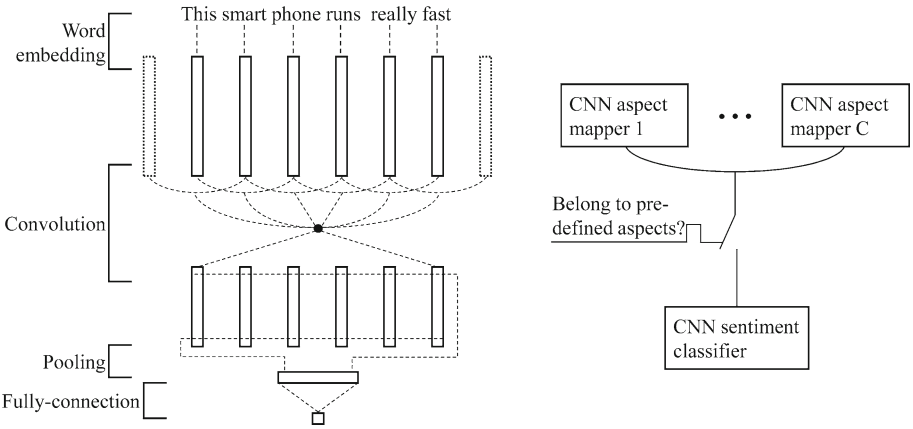

Fig. 3 C-CNN for aspect mapping and sentiment classification

$$f_i = g(\boldsymbol{w}_{i-r:i+r} \cdot \boldsymbol{u}) \tag{2}$$

Here g denotes a non-linear activation function. This filter is applied to every possible windows of the input sentence to generate a feature map.

$$\boldsymbol{f} = [f_1, f_2, \ldots, f_l] \tag{3}$$

The above describes the process that one feature map is extracted from one filter. The network uses $m_i (i = 1, 2, \ldots, C)$ filters to generate m_i feature maps for the i-th aspect mapper and m_{C+1} filters for the sentiment classifier. The filter weights for i-th aspect mapper are stored in a $hk \times m_i$ - dimensional matrix $\boldsymbol{W}_i^{(2)} \in \boldsymbol{R}^{hk \times m_i}$. For sentiment classifier, $\boldsymbol{W}_{C+1}^{(2)} \in \boldsymbol{R}^{hk \times m_2}$.

Pooling. This layer applies max-over-time pooling [41] to each of the feature maps produced by convolutional layers:

$$\hat{f} = \max(f_1, f_2, \ldots, f_l) \tag{4}$$

Max-over-time pooling takes the maximum element in each feature map and naturally deals with variable sentence lengths. It produces a fixed-sized feature vector $\boldsymbol{v}_i \in \boldsymbol{R}^{m_i}$ for the i-th task.

Fully-connection. The fixed-sized feature vectors produced by pooling layers are fed into fully-connected layers. Concretely, $\boldsymbol{v}_i$ is passed to a binary logistic regression classifier.

$$a_1 = 1/\left(1 + e^{-\boldsymbol{W}_i^{(3)} \boldsymbol{v}_i}\right), i = 1, 2, \ldots, C + 1 \tag{5}$$

Here $\boldsymbol{W}_i^{(3)} \in \boldsymbol{R}^{n \times m_i}$ is the weight matrix for i-th task, and a_i is the aspect output vector. For aspect mapper, a_i $(i = 1, 2, \ldots, C)$ is the probability of the input sentence belonging to the i-th aspect category; for sentiment classifier a_i $(i = C + 1)$ is the positive-sentiment probability.

Table 1 The number of sentences belonging to each aspect category

Amazon smartphone review dataset		Taobao skirt review dataset	
Aspects	#sentences	Aspects	#sentences
Battery	352	Price	1128
Run speed	370	Design	1296
Speaker	158	Quality	961
Screen	434	Fabric	549
Camera	344	Express delivery	647
None of the above	11,042	Service	532
All	12,700	None of the above	13,200
		All	18,314

4 Dataset and Experimental Setup

4.1 Datasets

To train our C-CNN, we need a collection of sentences labeled with aspects and sentiments. As there is no such benchmark corpus, we create two datasets and will make them publicly available for research purpose. The first one is Amazon Smartphone Review (ASR) dataset. ASR contains 12,700 smartphone review sentences crawled from amazon.com.[1] Each review sentence is labeled with respect to five pre-defined aspects, {battery, screen, camera, speaker, running speed}. Sentences belonging to at least one aspect are also labeled as expressing positive or negative sentiment.

The second is Taobao Skirt Review (TSR) dataset. TSR contains 18,314 labelled skirt review sentences and one million unlabeled. Reviews are crawled from taobao.com.[2] Each labelled review sentence is labeled with respect to six pre-defined aspects, {price, design, quality, fabric, express delivery, service}. Also, the sentiments for sentences belonging to at least one aspect are labeled as positive or negative. The number of sentences belonging to each aspect for ASR and TSR is shown in Table 1. Three people in our group crawled and labeled the reviews in sentence level which constitute the current version of corpus.

Table 2 shows some example sentences in ASR dataset that belong to aspect camera. Note that sentences 1, 3, 5, 6, 7, 8 do not explicitly mention camera, but they are still labeled as camera. This enables our system to handle both explicit and implicit aspects, and simultaneously extracts and categorizes different aspect expressions into the same aspect category.

4.2 Experimental Setup

Baselines. The baselines exploit Support Vector Machine (SVM) as classifiers. Specially, we adopt the L2-regularized L2-loss linear SVM. The implementation software is scikit-learn.[3] Multiple SVMs are cascaded in the way like C-CNN. One-hot representation of each word (or term) is employed as feature for training SVM. The terms we use are unigrams and bigrams.

[1] http://www.amazon.com.

[2] http://www.taobao.com.

[3] http://scikit-learn.org.

Table 2 Example sentences belonging to aspect camera in ASR

1.	The auto-focus/focus is middling at best on this phone
2.	The regular back camera is good
3.	Can take some pretty good quality photos (although not top quality)
4.	Camera works surprisingly well
5.	By far the best photos and speed I have experiencing on a phone
6.	Some expected noise on pictures taken in low-light
7.	It takes great pictures and video
8.	The colors aren't incredibly vivid, but the pictures do look pretty nice

The local weight of each term in the one-hot representation is simply assigned term presence (*tp*), i.e. 1 for presence and 0 for absence. The most commonly used weighting scheme, term frequency (*tf*), is not used as it produces very close results to *tp*. The reason may be that in our experiments most words in a sentence only occur one time, so weights assigned by *tp* and *tf* are almost the same with each other. We also use three global term weighting methods, *no* (no global term weighting), *idf* (inverse document frequency), and *re* (regularized entropy)[34].

Network settings. We use rectified linear units [47] as activation functions for convolutional layer, and sigmoid function for output layer. Network models are trained using stochastic mini-batch gradient descent with batch size of 1000, momentum of 0.9, learning rate of 0.5. The weights in all layers are initialized from a zero-mean Gaussian distribution with 0.1 as standard deviation and the constant 0 as the neuron biases. We use filter windows (h) of 1, 2, 3 with 100 feature maps each. The use of a momentum term is a technique that can help the network out of local minima and keep gradient pointing in the same direction.

Word2vec (w2v). Besides random initialization, we also pre-train word embeddings using word2vec tool, which implements continuous bag-of-words and skip-gram architectures for learning word vector representations [46]. We train skip-gram model with context window size of 9 on corpus of December 2013 English Wikipedia for ASR. For TSR, word embeddings are pre-trained using the one million unlabeled review sentences of TSR dataset. The embedding dimension is 30.

Regularization. For regularization we employ dropout on the input into max-pooling layer. Dropout based max-pooling randomly picks activation based on a multinomial distribution at training time, and employs probabilistic weighted pooling to act as model averaging at test time [48]. Use of dropout can prevent the network from overfitting problem. The dropout rate is set to 0.5.

Evaluation metric. We use F1-measure for performance evaluation of aspect mapping, and classification accuracy for sentiment classification. All comparisons are done using ten-fold cross-validation. That is, the overall results are averaged over ten folds.

Parameters. All model parameters are selected either according to the experience of common deep convolutional model setups, which is recognized by reseachers (e.g., momentum and dropout rate) [23,35,36,39,48] or considering both the performance and complexity of the model via cross-validations (e.g., mini-batch size, word embedding size, size and number of filters). Table 3 shows some of the main parameters used in the proposed C-CNN model.

4.3 Cross Validation

Cross-validation is used to generalize all the evaluation results, i.e., F1-measure for aspect mapping and classification accuracy for sentiment classification, which is a model validation

Table 3 Parameters used in the proposed C-CNN model

Parameter	Value
Mini-batch size	1000
Word embedding dimension	30
Momentum	0.9
Learning rate	0.5
Dropout rate	0.5
Filter size	1, 2, 3
#feature maps	100 for each filter size

technique for assessing how the trained model will perform on an independent data set. It is mainly used in settings where the goal is prediction, and one wants to estimate how accurately a predictive model will perform in practice.

One complete round of cross-validation involves partitioning the dataset into complementary subsets, training the model on one subset (i.e., training set), and validating the model on the other subset (i.e., validation set or testing set). To reduce variability, multiple rounds of cross-validation are performed using different partitions, and the validation results are averaged over the rounds.

In this paper, ten-fold cross-validation is used to generalize the evaluation results. The original dataset is partitioned into 10 subsets with equal size in random. Of the 10 subsets, a single subset is retained as the testing (or validation) set for testing the model, and the remaining 9 subsets are used as training set in combination. F1-measure and classification accuracy are both averaged over ten rounds of validations for every aspect and sentiment polarity.

5 Experimental Results

5.1 Computational Complexity

In practice, we use the elapsed time of training and test as the estimation of computational complexity. For most deep learning algorithms training is notoriously time consuming, especially for task on images. Though our C-CNN is designed for mapping and summarization task on text and less complicated, it still requires a lot more time to train the model compared with traditional machine learning methods. Without any acceleration method and parallel computing, C-CNN on ASR dataset takes around 20 min for training on an i7-4790K CPU. For SVM, one-vs-rest model is used in aspect mapping, so six SVM classifiers are trained on ASR dataset (including one sentiment classifier). The average time taken for training these classifiers is around 1.5 min, on the same computer. Training time of C-CNN could be greatly reduced by using acceleration techniques, such as parallel computing and GPU computing.

Different from the huge gap in training time between C-CNN and SVM, C-CNN takes less time than SVM based methods in test. Benefit from the cascaded structure and multi-output property of neural networks, C-CNN can complete the aspect mapping and sentiment classification within one single test. The elapsed time is shorter than 1 s. While SVM based method with one-vs-rest model have to test every aspects and sentiment separately, which leads to the increment of total time in test. Around 10 s are needed for SVM based model to accomplish both aspect mapping and sentiment classification.

Table 4 F1-measure for aspect mapping on ASR dataset via ten-fold cross-validation

Methods	Aspects				
	Battery	Run speed	Speaker	Screen	Camera
SVM+*no*	74.30	69.36	81.15	73.68	80.80
SVM+*idf*	73.50	71.22	79.78	73.61	79.72
SVM+*re*	74.62	71.29	81.15	73.68	81.55
SVM+*no*+bigram	74.67	68.72	81.11	74.76	80.76
SVM+*idf*+bigram	70.21	65.34	76.44	72.76	77.12
SVM+*re*+bigram	75.93	70.14	81.32	75.00	81.78
C-CNN	73.66	69.83	80.38	73.84	80.06
C-CNN+w2v	75.52	72.50	81.77	75.48	81.75
C-CNN+w2v+dropout	**76.03**	**72.67**	**82.22**	**75.83**	**83.74**

Bold values indicate the largest ones among each column

Table 5 Classification accuracy for sentiment classification on ASR dataset via ten-fold cross-validation

Methods	Accuracy
SVM+*no*	82.36
SVM+*idf*	81.28
SVM+*re*	82.53
SVM+*no*+bigram	82.97
SVM+*idf*+bigram	83.31
SVM+*re*+bigram	84.19
C-CNN	82.40
C-CNN+w2v	84.26
C-CNN+w2v+dropout	**84.87**

Bold value indicates the largest ones among column

5.2 Result for ASR

Table 4 presents the results of CNN based methods against SVM methods for aspect mapping on ASR dataset. For SVM based methods, global term weighting scheme *idf* provides poorer results than *no*, with or without bigrams as the feature terms. But *re* always improves F1-measure. The benefits of bigrams depend on aspects and global term weighting schemes. For example, using *idf* as the term weighting scheme, adding bigrams always harms the performance. But if feature terms are weighted with *re*, adding bigrams gives better results.

For CNN-based methods, C-CNN with randomly initialized word embeddings does not show clear superiority SVM based methods, and even underperforms SVM+*re*+bigram for all aspects. Pre-training word embeddings using word2vec provides significant gains of F1-measurefor C-CNN on all aspect mapping tasks. The improvement of F1-measureranges from +1.39% (80.38 vs. 81.77%) to +2.76% (69.83 vs. 72.50%). With dropout as the regularizer, C-CNN achieves the best results for all aspect mapping tasks.

Table 5 presents the classification accuracy of CNN-based methods against SVM based methods for sentiment classification task on ASR dataset. As with aspect mapping tasks, *idf* also underperforms *no*, and *re* outperforms *no*. Different from aspect mapping, adding bigrams always improves the performance of sentiment classification. The reason may be that bigrams could capture sentiment polarity shift caused by negation words. Again, C-

Table 6 F1-measure for aspect mapping on TSR dataset via ten-fold cross-validation

Methods	Aspects					
	Price	Design	Quality	Fabric	Express delivery	Service
SVM+*no*	91.13	88.50	96.17	90.84	92.37	88.60
SVM+*idf*	90.52	90.52	90.52	90.52	90.52	90.52
SVM+*re*	91.65	91.65	91.65	91.65	91.65	91.65
SVM+*no*+bigram	92.39	91.35	96.24	90.86	92.10	89.66
SVM+*idf*+bigram	91.79	89.37	95.32	90.13	91.53	88.53
SVM+*re*+bigram	**93.03**	91.38	96.71	91.91	92.15	90.06
C-CNN	90.10	89.53	95.30	90.01	91.57	88.40
C-CNN+w2v	92.38	91.25	96.73	91.54	92.57	89.76
C-CNN+w2v+dropout	92.77	**92.93**	**97.34**	**92.26**	**92.89**	**90.53**

Bold values indicate the largest ones among each column

Table 7 Classification accuracy for sentiment classification on TSR dataset via ten-fold cross-validation

Methods	Accuracy
SVM+*no*	96.98
SVM+*idf*	96.88
SVM+*re*	97.08
SVM+*no*+bigram	97.81
SVM+*idf*+bigram	97.87
SVM+*re*+bigram	97.85
C-CNN	97.14
C-CNN+w2v	98.03
C-CNN+w2v+dropout	**98.26**

Bold value indicates the largest ones among column

CNN underperforms SVM+*re*+bigram by large margins, but pre-training word embeddings using word2vec provides significant gains and provides similar results with SVM+*re*+bigram. Finally, C-CNN+w2v+dropout gives the best result, 84.87%, for the sentiment classification task.

5.3 Results for TSR

Table 6 presents the results of CNN based methods against SVM methods for aspect mapping on TSR dataset. Generally, theF1-measure for this dataset is much higher than TSR. This is due to that the expressions and keywords of reviews on Taobao are very similar, while reviews on Amazon are diverse. As with ASR, *idf* underperforms *no*, and *re* outperforms *no*, with or without bigrams as the feature terms. Different from ASR, adding bigrams almost always improves the performance of aspect mapping. Again, for CNN-based methods, C-CNN with randomly initialized word embeddings underperforms SVM+*re*+bigram for all aspects by large margins. Pre-training word embeddings with word2vec narrows the margins. Finally, with dropout as the regularizer, C-CNN achieves the best results on 5 of 6 aspect mapping tasks.

Table 7 presents the classification accuracy of CNN-based methods against SVM based methods for sentiment classification task on TSR dataset. Similar to the performance on

ASR, *idf* harms the performance and re provides benefits. Adding bigrams gives significant improvements for sentiment classification. The reason may be that bigrams could capture sentiment polarity shift caused by negation words. C-CNN with randomly initialized word embeddings provides similar result with SVM+no, and underperforms C-SVM with bigrams. With word2vec and dropout, C-CNN gives the best performance.

6 OpiSum

We build an aspect-based opinion summary system called OpiSum. Given the url of clothes on tmall.com, the system automatically crawls all customer reviews within the webpages. Then the system performs procedures in Fig. 2 to generate and visualize final aspect-based opinion summary. The demo of OpiSum can be found at http://114.215.167.42.

7 Conclusions

In this paper we have presented an aspect-based opinion summary system for particular products. Our system directly maps each review sentence into pre-defined aspects. This is particularly suitable for some vertical e-commerce websites that only sell particular products, or if users only care about opinions on particular product aspects. Meanwhile, two corpuses containing reviews labeled with aspects and polarity in sentence level are proposed in this paper, as no existing data set suitable for such task of aspect-based opinion summary can be found. The ASR corpus contains labeled reviews of smartphones in English crawled from Amazon. And the TSR corpus contains labeled reviews of skirts in Chinese from a chinses e-commerce platform, Taobao.com. Both review corpuses will be made public for research purpose. To attack aspect mapping and sentiment classification tasks, we have proposed a convolutional network based approach, C-CNN. C-CNN contains multiple aspect mappers and a single sentiment classifier, and aspect mappers and sentiment classifier are combined in a cascaded way. Empirical results imply the superiority of CNN based methods over SVM based methods in F1-measure and classification accuracy. A CNN based method is able to capture word relations of varying size. Also external features provided by parsers or other techniques are not required, as the model has the ability of learning and capturing features itself through the procedure of back and forward propagations in network training. Comparatively, feature designing and engineering are among the most important factors which can significantly influence the performance of traditional machine learning methods (e.g., SVM) for NLP problems. One common drawback of deep learning based methods is that they usually require a lot more time and computational resources in training for better performance, so as the proposed C-CNN in this paper. And parameters should be tuned carefully to protect the model from over-fitting (or under-fitting) problem. However, when it comes to test, the cascaded structure of C-CNN presents a remarkable reduction of elapsed time, compared to SVM. Using the processing procedures in Fig. 2 and C-CNN, we also build an aspect-based summary system called OpiSum, which can be accessed at http://114.215.167.42.

Acknowledgements This work was supported in part by National Natural Science Foundation of China under Grant 61371148.

References

1. Gao Z-K, Cai Q, Yang Y-X, Dang W-D, Zhang S-S (2016) Multiscale limited penetrable horizontal visibility graph for analyzing nonlinear time series. Sci Rep 6:35622. doi:10.1038/srep35622
2. Gao Z-K, Cai Q, Yang Y-X, Dong N, Zhang S-S (2017) Visibility graph from adaptive optimal kernel time-frequency representation for classification of epileptiform EEG. Int J Neural Syst 27:1750005
3. Gao Z-K, Jin N-D (2012) A directed weighted complex network for characterizing chaotic dynamics from time series. Nonlinear Anal Real World Appl 13(2):947–952. doi:10.1016/j.nonrwa.2011.08.029
4. Gao Z-K, Yang Y-X, Fang P-C, Zou Y, Xia C-Y, Du M (2015) Multiscale complex network for analyzing experimental multivariate time series. EPL (Europhys Lett) 109(3):30005
5. Hu M, Liu B (2004) Mining opinion features in customer reviews. AAAI 4:755–760
6. Blair-Goldensohn S, Hannan K, McDonald R, Neylon T, Reis GA, Reynar J (2008) Building a sentiment summarizer for local service reviews. In: Proceedings of WWW workshop on NLP in the information explosion era, pp 339–348
7. Kim S, Zhang J, Chen Z, Oh AH, Liu S (2013) A hierarchical aspect-sentiment model for online reviews. In: AAAI
8. Kobayashi N, Inui K, Matsumoto Y (2007) Extracting aspect-evaluation and aspect-of relations in opinion mining. In: EMNLP-CoNLL, Citeseer, pp 1065–1074
9. Sauper C, Barzilay R (2013) Automatic aggregation by joint modeling of aspects and values. J Artif Intell Res 46:89–127
10. Yang B, Cardie C (2012) Extracting opinion expressions with semi-Markov conditional random fields. In: Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning, Association for Computational Linguistics, pp 1335–1345
11. Zhao WX, Jiang J, Yan H, Li X (2010) Jointly modeling aspects and opinions with a MaxEnt-LDA hybrid. In: Proceedings of the 2010 conference on empirical methods in natural language processing, Association for Computational Linguistics, pp 56–65
12. Joshi M, Penstein-Rosé C (2009) Generalizing dependency features for opinion mining. In: Proceedings of the ACL-IJCNLP 2009 conference short papers, Association for Computational Linguistics, pp 313–316
13. Qiu G, Liu B, Bu J, Chen C (2011) Opinion word expansion and target extraction through double propagation. Comput Linguist 37(1):9–27
14. Wu Y, Zhang Q, Huang X, Wu L (2009) Phrase dependency parsing for opinion mining. In: Proceedings of the 2009 conference on empirical methods in natural language processing: volume 3, Association for Computational Linguistics, pp 1533–1541
15. Zhuang L, Jing F, Zhu X-Y (2006) Movie review mining and summarization. In: Proceedings of the 15th ACM international conference on information and knowledge management, ACM, pp 43–50
16. Breck E, Choi Y, Cardie C (2007) Identifying expressions of opinion in context. In: Proceedings of the 20th international joint conference on artificial intelligence, pp 2683–2688
17. Irsoy O, Cardie C (2014) Opinion mining with deep recurrent neural networks. In: Proceedings of the 2014 conference on empirical methods in natural language processing, pp 720–728
18. Jakob N, Gurevych I (2010) Extracting opinion targets in a single-and cross-domain setting with conditional random fields. In: Proceedings of the 2010 conference on empirical methods in natural language processing, Association for Computational Linguistics, pp 1035–1045
19. Jin W, Ho HH, Srihari RK (2009) A novel lexicalized HMM-based learning framework for web opinion mining. In: Proceedings of the 26th annual international conference on machine learning, Citeseer, pp 465–472
20. Brody S, Elhadad N (2010) An unsupervised aspect-sentiment model for online reviews. In: Paper presented at the human language technologies: the 2010 annual conference of the North American Chapter of the Association for Computational Linguistics. Los Angeles, California
21. Chen Z, Mukherjee A, Liu B (2014) Aspect extraction with automated prior knowledge learning. In: Proceedings of the 52nd annual meeting of the Association for Computational Linguistics, pp 347–358
22. Jo Y, Oh AH (2011) Aspect and sentiment unification model for online review analysis. In: Proceedings of the fourth ACM international conference on Web search and data mining, ACM, pp 815–824
23. Kim Y (2014) Convolutional neural networks for sentence classification. In: Proceedings of the 2014 conference on empirical methods in natural language processing. pp 1746–1751
24. Moghaddam S, Ester M (2012) On the design of LDA models for aspect-based opinion mining. In: Proceedings of the 21st ACM international conference on Information and knowledge management, ACM, pp 803–812
25. Mukherjee A, Liu B (2012) Aspect extraction through semi-supervised modeling. In: Proceedings of the 50th annual meeting of the Association for Computational Linguistics: Long Papers-vol 1, Association for Computational Linguistics, pp 339–348

26. Sauper C, Haghighi A, Barzilay R (2011) Content models with attitude. In: Proceedings of the 49th annual meeting of the Association for Computational Linguistics: human language technologies, volume 1, Association for Computational Linguistics, pp 350–358
27. Bespalov D, Bai B, Qi Y, Shokoufandeh (2011) A Sentiment classification based on supervised latent n-gram analysis. In: Proceedings of the 20th ACM international conference on information and knowledge management, ACM, pp 375–382
28. Davidov D, Tsur O, Rappoport (2010) A Enhanced sentiment learning using twitter hashtags and smileys. In: Proceedings of the 23rd international conference on computational linguistics: posters, Association for Computational Linguistics, pp 241–249
29. Nakagawa T, Inui K, Kurohashi S (2010) Dependency tree-based sentiment classification using CRFs with hidden variables. In: Human language technologies: the 2010 annual conference of the North American chapter of the Association for Computational Linguistics, Association for Computational Linguistics, pp 786–794
30. Ng V, Dasgupta S, Arifin S (2006) Examining the role of linguistic knowledge sources in the automatic identification and classification of reviews. In: Proceedings of the COLING/ACL on main conference poster sessions, Association for Computational Linguistics, pp 611–618
31. Paltoglou G, Thelwall M (2010) A study of information retrieval weighting schemes for sentiment analysis. In: Proceedings of the 48th annual meeting of the Association for Computational Linguistics, Association for Computational Linguistics, pp 1386–1395
32. Pang B, Lee L, Vaithyanathan S (2002) Thumbs up: sentiment classification using machine learning techniques. In: Proceedings of the ACL-02 conference on empirical methods in natural language processing, vol 10, Association for Computational Linguistics, pp 79–86
33. Riloff E, Patwardhan S, Wiebe J (2006) Feature subsumption for opinion analysis. In: Proceedings of the 2006 conference on empirical methods in natural language processing, Association for Computational Linguistics, pp 440–448
34. Wu H, Gu X, Gu Y (2016) Balancing between over-weighting and under-weighting in supervised term weighting. Inf Process Manag. doi:10.1016/j.ipm.2016.10.003
35. Kalchbrenner N, Grefenstette E, Blunsom P (2014) A convolutional neural network for modelling sentences. In: Proceedings of the 52nd annual meeting of the Association for Computational Linguistics, pp 655–665
36. Santos CND, Gatti M (2014) Deep convolutional neural networks for sentiment analysis of short texts. In: Proceedings of the 25th international conference on computational linguistics, pp 69–78
37. Lin C, He Y (2009) Joint sentiment/topic model for sentiment analysis. In: Proceedings of the 18th ACM conference on Information and knowledge management, ACM, pp 375–384
38. Jin W, Ho HH, Srihari RK (2009) OpinionMiner: a novel machine learning system for web opinion mining and extraction. In: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, pp 1195–1204
39. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. Adv Neural Inf Process Syst 2012:1097–1105
40. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1–9
41. Collobert R, Weston J, Bottou L, Karlen M, Kavukcuoglu K, Kuksa P (2011) Natural language processing (almost) from scratch. J Mach Learn Res 12(Aug):2493–2537
42. Shen Y, He X, Gao J, Deng L, Mesnil G (2014) A latent semantic model with convolutional-pooling structure for information retrieval. In: Proceedings of the 23rd ACM international conference on conference on information and knowledge management, ACM, pp 101–110
43. Shen Y, He X, Gao J, Deng L, Mesnil G (2014) Learning semantic representations using convolutional neural networks for web search. In: Proceedings of the 23rd international conference on world wide web, ACM, pp 373–374
44. Santos CND, Xiang B, Zhou B (2015) Classifying relations by ranking with convolutional neural networks. In: Proceedings of ACL-IJCNLP 2015, ACL, pp 626–634
45. Zeng D, Liu K, Lai S, Zhou G, Zhao J (2014) Relation classification via convolutional deep neural network. In: Proceedings of the 25th international conference on computational linguistics, pp 2335–2344
46. Le QV, Mikolov T (2014) Distributed representations of sentences and documents. In: Proceedings of The 31st international conference on machine learning, pp 1188–1196
47. Nair V, Hinton GE (2010) Rectified linear units improve restricted Boltzmann machines. In: Proceedings of the 27th international conference on machine learning (ICML-10), pp 807–814
48. Wu H, Gu X (2015) Towards dropout training for convolutional neural networks. Neural Netw 71:1–10