Multimedia Systems manuscript No.
(will be inserted by the editor)

Exploring Overall Opinions for Document Level Sentiment Classification with Structural SVM

Xiaojia Pu · **Gangshan Wu** · **Chunfeng Yuan**

Received: date / Accepted: date

Abstract As a fundamental task of sentiment analysis, document level sentiment classification aims to predict user's overall sentiment (e.g. positive or negative) towards the target in a document. The document usually consists of various opinion sentences towards different aspects with different sentiments. Therefore, the overall opinion towards the whole target should play a more important role in document sentiment prediction. However, most existing methods for the task treat all sentences of the document equally. Thus, they are easy to encounter difficulty when the sentiments of most aspect opinion sentences are not coherent with the overall sentiment. To address this, we propose a novel method for document sentiment classification which adequately explores the effect of overall opinion sentences. In our method, firstly, multiple features are exploited to recognize candidate overall opinion sentences, and then a structural SVM is utilized to encode the overall opinion sentences for document sentiment classification. Experiments on several public available datasets including product reviews and movie reviews show the effectiveness of our method.

Keywords Sentiment Classification · Overall Opinion · Structural SVM

1 Introduction

With the advent of Web 2.0 and social networks, there are many kinds of social media data around the world [31] [27]. They are usually user generated contents [22] [32] [29]. Among them, the opinionated contents, e.g. product reviews, are generated rapidly and of great value for decision making. In order to effectively explore the opinions, sentiment analysis has been emerging as an important technique [19] [6]. As a fundamental task of sentiment

Xiaojia Pu, Gangshan Wu, Chunfeng Yuan
State Key Laboratory for Novel Software Technology
Department of Computer Science and Technology
Nanjing University, Nanjing, China
E-mail: puxiaojia@gmail.com

Gangshan Wu
E-mail: gswu@nju.edu.cn

Chunfeng Yuan
E-mail: cfyuan@nju.edu.cn

Table 1 A review of digital camera Canon S100, in which there are many opinion sentences talking about the unsatisfied aspects, however, the overall opinion is positive. The opinion expressions are highlighted in bold.

Review
I want to start off saying that this camera is **small** for a reason. Some people, in their reviews, complain about its **small size**, and how it doesn't compare with larger cameras. I'm in high school, and **this camera is perfect** for what I use it for, carrying it around in my pocket so I can take pictures whenever I want to, of my friends and of funny things that happen. The only thing I **don't like is the small size (8 MEG) memory card** that comes with it. I have to move pictures off of it every day so I have **room for more pictures** the next, and **I don't have enough money to buy the 256 MEG card** that I've had my eye on for a while. A **larger memory card and extra battery** are good things to buy. Other than that **pictures taken in the dark are not as nice as I'd like them**, I'd say that **this camera is perfect**.

analysis, document level sentiment classification aims to automatically determine and extract the overall sentiment (e.g. positive and negative) towards the target in the review [19].

The existing methods for document level sentiment classification can be divided into two categories, i.e. linguistic rule based approach and machine learning based approach [19]. The rule based approach utilizes a predefined sentiment lexicon and some linguistic rules to determine the sentiment of the reviews [37] [33]. This approach is simple, but suffers from scalability, since the sentiment lexicons and linguistic rules are commonly manually defined by experts. Instead, learning based approach takes the task as a special case of text classification, which usually represents documents with bag-of-ngrams features and builds some classifier upon that [28] [38].

With both kinds of the methods, most of them treat the text or the opinion sentences in the document equally informative for final sentiment prediction. Some learning methods simply merge all the text into a flat feature vector. In fact, according to the language and expressive habits, when a user describes the opinion or evaluation towards a target, various aspects or attributes may be mentioned (including the whole target). Some of them are positive, while the others are negative. It's reasonable that the sentiment or polarity of a review mainly depends on the overall opinions rather than those about the specific attributes or aspects. However, this issue is often neglected by most of the existing methods. They are easy to encounter difficulty when the sentiments of most aspect opinion sentences are not coherent with the overall sentiment.

Take the review in Table 1 for example [1], it mentions several aspects, and complains about the small size of the memory card and pictures in the dark, however it gives a positive overall rating to the product. The positive overall opinion sentences lead to a positive overall rating though there are many negative opinions towards detailed aspects.

We can find that overall opinion (OOP) sentences are more informative for predicting document level sentiment, and the aspect opinion sentences whose sentiments are not coherent with the overall sentiment may mislead the classifier. Therefore, the OOP sentences should be sufficiently utilized for document sentiment classification. To accomplish this, firstly, the OOP sentences should be correctly recognized, secondly, the relationship between OOP sentences and document sentiment should be well explored.

In this paper, we propose a novel and effective method called SVM^{eop} to utilize the overall opinions to improve document level sentiment classification. Our method takes advantage of structural SVM [40], in which the OOP sentences are taken as the hidden variables for document sentiment. The structural SVM conducts the hidden variable recognition and the final classification simultaneously. The main difficulty of structural SVM is the initialization

[1] This is a review of Canon S100 in the dataset: http://www.cs.uic.edu/ liub/FBS/Reviews-9-products.rar

of the hidden variables, which influences the accuracy and training time [40]. In order to resolve this, we exploit multiple features to recognize candidate OOP sentences firstly, and then the candidate OOP sentences are incorporated as the initial value, which ensures the accuracy of the result and reduces the training time. We conduct document level sentiment classification on several public available datasets, the results show the importance of the OOP and the effectiveness of our method. The extracted overall opinions also well explain the overall sentiment.

The contribution of our works can be summarized as follows:

1. We explicitly point that for document level sentiment classification, the OOP sentences are more important, and propose an effective method to explore the OOP sentences for document level sentiment classification.

2. Our method takes advantage of structural SVM and directly optimizes document level sentiment classification result, the sentence level annotations are not needed.

3. We combine multiple features to recognize candidate OOP sentences, which is convenient and benefits the subsequent procedures greatly.

The rest of the paper is organized as follows. Section 2 overviews the related work. In section 3, we briefly introduce our method. In section 4 and 5, the details of our method will be presented. In section 6, we discuss the experiments and results. Finally, section 7 concludes the paper and discusses future work.

2 Related work

Document level sentiment classification is a fundamental problem in sentiment analysis, which aims at identifying the sentiment label of a document [27] [19]. There have been plenty of works for this task, and these methods can be grouped into two categories: 1) rule based approach with sentiment lexicon [37] [9] [5] [33]; 2) machine learning based approach [28] [38] [13] [19].

The lexicon based sentiment classification approach utilizes a predefined sentiment lexicon and some linguistic rules to determine the sentiment of the reviews [37] [9] [33]. This approach is simple and interpretable, but suffers from scalability and is inevitably limited by sentiment lexicons that are commonly created manually by experts.

Learning based approach takes sentiment classification as a special case of text classification problem, and use machine learning methods for this task [28] [38]. The documents are usually represented as bag-of-features. [28] firstly investigated machine learning methods including Naive Bayes, Maximum Entropy, and SVM for sentiment classification in movie reviews, and evaluated different features including unigrams, bigrams, adjectives, and part-of-speech tags. Their experimental results suggested that a SVM classifier with unigram presence features outperforms other competitors. The dominant following works aimed to design effective models and features for building a powerful sentiment classifier. The representative features include word ngrams [38], sentiment lexicon features [13], etc. [38] proposed a SVM variant and used Naive Bayes log-count ratios as feature values to classify sentiment polarity. They showed that SVM was better at full-length reviews, and Multinomial Naive Bayes was better at short-length reviews. These methods use local ngram information and do not capture semantic relations between sentences.

As a unsupervised machine learning approach, topic models aim to mine the semantics of the user generated contents [3] [20] [30] [15], which have also been used for sentiment analysis. The Joint Sentiment/Topic model (JST)detects sentiment and topic simultaneously from text [18]. In JST, each document has a sentiment label distribution. Topics are associated to sentiment labels, and words are associated to both topics and sentiment labels. For topic modeling approach, the prior information e.g. the sentiment lexicon is often essential for sentiment detection.

Another line of research concentrates on modeling the semantic relationship between the document and sentences. [26] separated subjective portions from the objective text by finding minimum cuts in graph of sentences to achieve better sentiment classification performance. [23] investigated a global structured model for jointly classifying sentiment polarity at different levels of granularity. [39] used sentence-level latent variables to improve document level sentiment prediction. However, they are still lack of taking into account of overall opinion sentences.

Recently, some deep learning (neural networks) or representation learning methods are emerging, which provide an alternative way to learn continuous text representation [24] [1]. Several studies learn sentiment-specific word embeddings by taking into account sentiment of texts [21] [14] [36]. [16] introduced Paragraph Vector to learn document representation from semantics of words. [17] proposed a hierarchical neural auto encoder for paragraphs and documents. [35] represented document with a gated recurrent neural network, which adaptively encodes semantics of sentences and their relations. The deep learning algorithms usually need a huge amount of data to learn the parameters. That will be time-consuming and sometimes not practical for instant needs.

In summary, though the performance of some existing methods is good in some datasets, and some works have considered the relationship and influence between sentences and document, the special utilization of overall opinion for document sentiment classification is still insufficient. Instead, we directly capture the most important factor i.e. overall opinion sentences for determining the global document polarity, avoiding inducing some unimportant and misleading sentences.

3 Overview of our method

In this section, we give a brief description of our method. The general procedure is shown in Figure 1.

Intuitively, if we could recognize the OOP sentences with some aspect extraction techniques, then simply building the classification model on the OOP sentences is enough for our task. However, it's not practical. First, the state-of-art aspect extraction techniques are still not perfect and usually time-consuming. Supervised learning methods usually need sentence level annotation [11] [10] and unsupervised methods usually produce incoherent aspects [15] [25].

Therefore, we take advantage of structural SVM, and the OOP sentences are taken as the hidden variables for document sentiment. The structural SVM can conduct the hidden variable recognition and the final text classification simultaneously [40]. The difficulty is to give a good initialization of the hidden variable, without which, the structural SVM needs a lot of time to achieve convergence during training and sometimes the result is not very accurate. To resolve this, we exploit multiple features to recognize candidate OOP sentences firstly, and then the candidate OOP sentences will be incorporated as the initial value for our model, which ensures the accuracy and reduces the training time.

The details about the method will be given in the next two sections. For the convenience to understand our method, we illustrate the notations throughout the paper in Table 2.

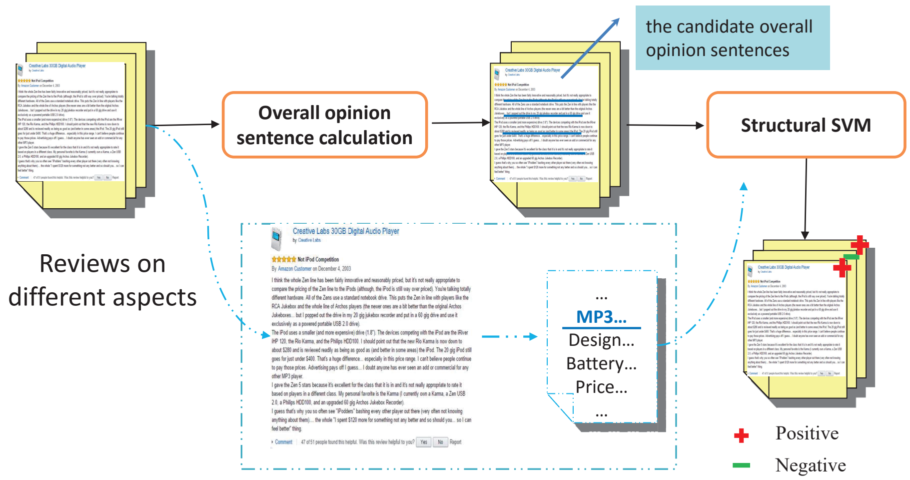

Fig. 1 The overview of our method, firstly, the candidate overall opinion sentences are recognized for model initialization, and then the structural SVM is used to explore the overall opinion for document level sentiment classification.

Table 2 Basic notations used in our paper.

Variable	Description
x	a review/document
$len(x)$	the length(number of sentences) of a review/document
$\hat{y}$	the predicted output of x
y	the correct output of x
x_i	the ith review
y_i	the sentiment polarity of the ith review
$x^i orx^j$	the ith or jth sentence in the review
S_x	the set of total sentences in review x
S_{x_i}	the set of total sentences in review x_i
S	the set of overall opinion sentences in review x
$\|S\|$	the size of the S
S_i	the set of overall opinion sentences in review x_i
$\Psi(x,y,S)$	joint feature map describing quality of predicting sentiment y using S for review x
$\psi_{pol}(x^j)$	the polarity features of sentence x_j
$\psi_{subj}(x^j)$	the subjectivity features of sentences x^j
w	the weight of the model
w_{pol}	the weight of $\psi_{pol}(.)$
w_{subj}	the weight of $\psi_{subj}(.)$
$N(x)$	the normalization factor for document classifier

4 Probabilistic method for recognizing candidate overall opinion sentences

To recognize candidate overall opinion sentences, on the one hand, our method should be efficient, since our final target is document sentiment classification, the time cost in this step should be not very expensive. On the other hand, we should consider all the possibilities of the OOP sentences, and make the candidate OOP sentences as correct as possible.

To meet the requirements, we exploit multiple features and use a function to calculate

Table 3 The features for recognizing candidate overall opinion sentences, which are divided into two kinds, i.e. linguistic features and positional features.

categories	description	example
linguistic features	words for the target	camera, ipod
	conclusion phrases	in a word, overall
positional features	position in the review	$f_{posi}(x^i)$
	is the title ?	$f_{title}\{x^i\}$

the probability of the sentences, and then the sentences with high scores will be taken as the OOP sentences for the initialization for our model.

4.1 Exploiting linguistic and positional features

There are two kinds of clues exploited for the recognition of overall opinion sentences, including the linguistic and positional features. Table 3 illustrates all the used features in our paper.

4.1.1 Linguistic features

The overall opinions imply attitude towards the whole target, so the words or phrases which stand for the whole target are important clues. Take the product 'Cannon 50D ' for example, the phrases, e.g. 'the camera', '50D' and 'Cannon 50D', are all indicative for the 'Cannon 50D' camera. It will be very beneficial for the task, if the user could provide some information similar with these terms. This kind of words or phrases compose a dictionary Dic_{target}.

Besides, conclusive words such as 'overall' directly indicate the overall opinion sentences, which are good heuristic information for our task. In practice, these words or phrases could be easily collected. These conclusive words compose a dictionary Dic_{conc}.

The two dictionaries, i.e. Dic_{target} and Dic_{conc}, constitute the linguistic features for overall opinions.

4.1.2 Positional features

The overall opinion sentences could appear in various positions, e.g. in the title, the beginning, the middle, and the end of the document. It is reasonable that sentences in the beginning or in the end should be of higher probability to be overall opinions. Since the title is the abstraction of the review, it's also likely to be the overall opinion.

Two psycholinguistic and psychophysical experiments showed that in order to efficiently extract polarity of written texts such as customer reviews on the Internet, one should concentrate more computational efforts on messages in the final position of the text [2].

4.1.3 Feature quantification

For the quantification of linguistic features, the cosine similarity is calculated between each sentence and the dictionary. The dictionary is represented as a vector of the indicative words or phrases. The sentence is also represented as a vector, in which each dimension corresponds to the vector of dictionary. If the corresponding word or phrase appears in the sentence, the value of that dimension is set as 1, otherwise, it's set as 0.

Then the cosine similarity of the sentence x^i and dictionary Dic is calculated as follows.

$$Sim(x^i, Dic) = \frac{vec(Dic) \cdot vec(x^i)}{|vec(Dic)|_2 \cdot |vec(x^i)|_2} \tag{1}$$

where $vec(Dic)$ is the vector representation of dictionary Dic, $vec(x^i)$ is the vector representation of x^i.

The cosine similarity measures how likely the sentence can be the overall opinion according to the linguistic features. If the sentence contains more indicative words or phrases, the cosine similarity will be higher, which means it is more likely to be the overall opinion sentence.

For the quantification of positional features, f_{posi} and f_{title} are listed below. With f_{posi}, the sentence x^i that appears in the beginning or the end, will get larger score than those sentences in the middle. $posi(x^i)$ is the position of the sentence x^i in the document x. f_{title} means that the sentence x^i in the title will be with high probability to be the OOP sentences.

$$f_{posi}(x^i) = max\{\frac{posi(x^i)}{len(x)}, 1 - \frac{posi(x^i)}{len(x)}\} \tag{2}$$

$$f_{title(x^i)} = \begin{cases} 1 & x^i \, is title \\ 0 & x^i \, is not title \end{cases} \tag{3}$$

4.2 Probability function

The general idea is to measure the probability about each sentence as overall opinion sentence. Specifically, suppose function $f_{oop}(.)$ parameterized by w_{oop} is used for probability calculation. For a new sentence x^i, $\phi(x^i)$ denotes corresponding feature vector.

$$\phi(x^i) = \{Sim(x^i, Dic_{target}), Sim(x^i, Dic_{conc}), f_{posi}(x^i), f_{title}\{x^i\}\} \tag{4}$$

The two categories of features illustrated above are incorporated as elements of $\phi(x^i)$. Then the probability score for sentence x^i to be overall opinion is

$$f_{oop}(x^i; w_{oop}) = w_{oop} \cdot \phi(x^i). \tag{5}$$

The weight of the probability function, i.e. w_{oop}, measures the importance of these features. Here, the weight of the features in $\phi(x^i)$are set equally. Thus, $\frac{1}{4}$ is assigned as the weight for each dimension of w_{oop} respectively.

With the probability function, the scores of all sentences in a review can be calculated, then the top k sentences will be selected as the candidate OOP sentences. The k is an empirical value, and tuned in our experiments.

5 The proposed structural SVM model

In this section, in order to capture the overall opinion information for document level sentiment classification, we propose a model which takes advantage of structural SVM to explore the connection between the overall opinion sentence and the document sentiment. For convince, the proposed model is called SVM^{eop} (SVM for Exploring Overall Opinions).

5.1 Preliminary of Structural SVM

Structural SVM is an extension of traditional SVM for interdependent and structured output spaces [40].

Given a training set which consists of input-output structure pairs, $\{(x_1, y_1), ..., (x_n, y_n)\} \in (\mathrm{x} \times \mathrm{y})^n$, it want to learn a linear prediction rule of the form

$$f_w(x) = arg \max_{y \in \mathrm{y}} [w \cdot \Phi(x, y)] \tag{6}$$

where Φ is a joint feature vector that describes the relationship between input x and structured output y, with w being the parameter vector.

When training structural SVMs, the parameter vector w is determined by minimizing the loss function $\Delta(y, \hat{y})$ that quantifies how much the prediction $\hat{y}$ differs from the correct output y. Since Δ is typically nonconvex and discontinues, and there are usually exponentially possible structures $\hat{y}$ in the output spaces y, it is usually replaced with a piecewise linear convex upper bound

$$\Delta(y_i, \hat{y}_i(w)) \leq \max_{\hat{y} \in \mathrm{y}} [\Delta(y_i, \hat{y}) + w \cdot \Phi(x_i, \hat{y})] - w \cdot \Phi(x_i, y_i)$$

where $\hat{y}_i(w) = arg \max_{y \in \mathrm{y}} w \cdot \Phi(x_i, y)$

In many applications, the input-output relationship is not completely characterized by $(x, y) \in \mathrm{x} \times \mathrm{y}$ pairs in the training set alone, but also depends on a set of unobserved latent variables $h \in \mathrm{H}$. To generalize the structural SVM formulation, the joint feature vector $\Phi(x, y)$ can be extended with an extra argument h to $\Phi(x, y, h)$ to describe the relation among input x, output y, and latent variable h. We want to learn a prediction rule of the form

$$f_w(x) = arg \max_{(y,h) \in \mathrm{y} \times \mathrm{h}} [w \cdot \Phi(x, y, h)]. \tag{7}$$

5.2 *SVM^eop*

Let x denote a document, $y = \{1, -1\}$ denote the sentiment of a document, and S denote the set of the overall opinion sentences in x. Let $\Psi(x, y, S)$ denote a joint feature map that outputs features describing the quality of predicting sentiment y using S for document x. Here, S is the set of overall opinion sentences. In this paper, we focus on linear models, so give a weight vector w, we can write the quality of predicting y as

$$F(x, y, S) = w^T \Psi(x, y, S), \tag{8}$$

and a document level sentiment classifier as

$$y^* = arg \max_{y} \max_{s \subset S_x} F(x, y, S; w), \tag{9}$$

where S_x denotes the collection of total sentences for x.

Let x^j denote the jth sentence of document x. We propose the following instantiation:

$$\begin{aligned} F(x, y, S; w) &= w^T \Psi(x, y, S) \\ &= \frac{1}{N(x)} \sum_{x^j \in S} y \cdot w_{pol}^T \psi_{pol}(x^j) + w_{subj}^T \psi_{subj}(x^j) \end{aligned} \tag{10}$$

where the first term in the summarization captures the quality of predicting polarity y on sentences in S, the second term captures the quality of predicting S as the overall subjective opinion sentences, and $N(x)$ is a normalization factor.

We represent the weight vector as

$$w = [w_{pol}; w_{subj}], \tag{11}$$

and $\psi_{pol}(x^j)$ denotes the polarity features of sentence x_j, $\psi_{subj}(x^j)$ denotes the subjectivity features of sentences x^j, w_{pol} and w_{subj} are the learned weights of the model.

The unigram features are used in our method, ψ is defined with the bag-of-words feature representation, with one feature corresponding to each word in the lexicon of the corpus.

5.3 Document sentiment prediction

The sentiment classifier is shown below, which is an expansion of Eqn. 9.

$$(y^*, S^*) = arg \max_{y\in\{+1,-1\}} \{\max_{S\subset S_x} w^T \Psi(x, y, S)\} \tag{12}$$

The model aims to predict the sentiment with the best OOP set S, i.e. the extracted overall opinion sentences. S is used to help predicting document sentiment polarity. The detailed inference algorithm is illustrated in Algorithm 1.

Algorithm 1 Inference Algorithm

Input:
x
Output:
(y, s)
$s_+ \leftarrow arg\,max_{s\in S_x} w^T \Psi(x, +1, s)$
$s_- \leftarrow arg\,max_{s\in S_x} w^T \Psi(x, -1, s)$
if $w^T \Psi(x, +1, s_+) > w^T \Psi(x, -1, s_-)$ **then**
    Return $(+1, s_+)$
else
    Return $(-1, s_-)$
end if

The size of OOP set S, i.e. $|S|$, is usually small in comparison with the size of the document x, i.e. $len(x)$. Therefore, during the prediction and learning, we should constrain the searching place for S in S_x. Here, we introduce $f(x)$, which is the upper bound of the size of S, e.g. $f(x) = 0.1 \cdot len(x)$.

$$|S| \leq f(x) \tag{13}$$

For each sentence x^j, we compute the joint score with respect to overall opinion and label y as

$$score(x^j, y) = y \cdot w_{pol}^T \psi_{pol}(x^j) + w_{subj}^T \psi_{subj}(x^j). \tag{14}$$

After calculating, according to the score, the top $|S|$ sentences will be chosen as the new OOP sentences set. In our experiments, we tune the size of S with respect to the number of sentences in x to obtain the optimal performance.

5.4 Learning algorithm

The learning process is to optimize the following problem.

$$\min_{w,\xi \geq 0} \frac{1}{2}||w||^2 + \frac{C}{N}\sum_{i}^{N} \xi_i \tag{15}$$

$$s.t. \forall i \max_{S_i \subset S_{x_i}} w \cdot \Psi(x_i, (y_i, S_i)) \geq \max_{S_i' \subset S_{x_i}} w \cdot \Psi(x_i, (-y_i, S_i')) + 1 - \xi_i$$

where C is the regularization parameter, N is the number of training instances.

This is a SVM style objective function, for each training instance x_i, the corresponding constraint is quantified over the best possible OOP sentence set S_i. The S_i is modeled as a latent variable. Since it is non-convex, we try to solve it using the combination of CCCP algorithm with cutting plane training of structural SVM [12] as proposed in [40].

The initialization of S is set as the candidate OOP sentences calculated in section 4. With the well-generated candidate OOP sentences set, the algorithm tries to make refinement of the set according to the sentiment prediction in the training dataset.

The detailed algorithm of our method is shown in Algorithm 2. This task is accomplished with an iterative approach. With the initial S, the initial parameter w of the model is learned, and then new overall opinion sentences set S is resolved. The training procedure alternates between solving the resulting structural SVM (called SSVMSolve in Algorithm 2) using the currently known best OOP sentences set, and making a guess of new OOP sentences set until the learned w converges.

[40] showed that this alternating procedure is guaranteed to convergence to a local optimum, so the initialization of the S is important since which influences the accuracy of final result. Our candidate OOP sentences set calculated in section 4 for initialization ensures the accuracy and reduces the searching time during training.

In our experiments, we do not train until the convergence, instead, we evaluate the performance in the validation set to choose the halting iteration. The normalizing factor is set as $N(x) = \sqrt{f(x)}$ as described in [39], where $f(x)$ is the upper bound of the size of the extracted candidate overall opinion sentences, i.e. $|S|$, which will be further discussed in the experimental section.

Algorithm 2 The Detailed Training Algorithm

Input:
{ $(x_1, y_1), \ldots, (x_N, y_N)$ }//training data
C //regularization parameter
$(S_1, \ldots, S_N)$ //initial guess
Output:
w
$w \leftarrow SSVMSolve(C, \{(x_i, y_i, S_i)\}_{i=1}^{N})$
while not convergence **do**
  for $i = 1, \cdots, N$ **do**
    $s_i \leftarrow argmax_{S \subset S_{x_i}} w^T \Psi(x_i, y_i, S)$
  end for
  $w \leftarrow SSVMSolve(C, \{(x_i, y_i, S_i)\}_{i=1}^{N})$
end while
Return w

6 Experiments

We conduct experiments on several public available datasets to evaluate our proposed method. We firstly evaluate the performance of document level sentiment classification, and then evaluate the quality of extracted overall opinion sentences. The experiments are implemented using C++, and run in a PC with Intel Core i5 CPU and 8GB RAM.

6.1 Datasets

We evaluate our method on five real-world datasets, **Oscar Data** [34], **Liu Data** [9] [5], **McAuley Data** [22], **IMDB(S)** [26],**IMDB(L)** [21]. The **Oscar Data**, **Liu Data** and **McAuley Data** are all comprised of product reviews from Amazon. The **IMDB(S)** and **IMDB(L)** are all movie reviews. The details are listed below.

- **Oscar Data**[2] contains some reviews about books, DVDs, electronics, music, and videogames.
- **Liu Data** [3] contains fine annotated datasets. For each review, sentences are labeled with aspect information. We choose reviews of several digital products for the experiments.
- **McAuley Data** [4] is a huge dataset of product reviews, we choose more than 10,000 cellphone reviews for our experiments.
- **IMDB(S)** [5] is a benchmark dateset for sentiment classification of movie reviews which contains 2000 reviews.
- **IMDB(L)** [6]consists of 50,000 binary labeled reviews from IMDB. There are 25000 movie reviews in the training set, and another 25000 for testing.

In this paper, we aim to classify the document sentiment into two categories, i.e. positive and negative, so the neutral reviews are removed in our experiments. Table 4 summarizes the datasets statistics.

Table 4 Some statistics of the datasets.

dataset	reviews	positive	negative
Oscar Data	196	97	99
Liu Data	489	301	188
McAuley Data	13448	7615	4233
IMDB(S)	2000	1000	1000
IMDB(L)	50000	25000	25000

All datasets are processed using lowercased stemmed unigram words, and the stop-words are removed. The documents are represented by the bag of words. We choose 0/1 for term weighting, i.e. the occurrence of a word or not, which is widely used in sentiment classification [19].

[2] https://github.com/oscartackstrom/sentence-sentiment-data
[3] https://www.cs.uic.edu/ liub/FBS/sentiment-analysis.html
[4] http://snap.stanford.edu/data/web-Amazon.html
[5] http://www.cs.cornell.edu/people/pabo/movie-review-data/
[6] http://ai.stanford.edu/ amaas/data/sentiment

6.2 Experimental setup

6.2.1 Methods for comparison

To evaluate the document level sentiment classification, we compare our method SVM^{eop} with various methods illustrated below.

- **SVM:** the most common classification model for text classification as well as sentiment classification.
- **MinCut:** the minimum cut algorithm is utilized to extract representative subjective sentences and then some classifier, e.g. SVM is built upon these sentences for sentiment classification [26].
- SVM^{sle}**:** a typical method based on structural SVM for sentiment classification, which considers the interactions between the sentences and documents [39], but without explicit difference between OOP sentences and other opinion sentences.
- **JST:** Joint Sentiment/Topic model [18] is an extension of LDA [3] to detect sentiment and topic simultaneously from text. It's fully unsupervised, therefore for sentiment prediction, a sentiment lexicon is usually needed, in the experiments, the MPQA subjective lexicon [7] is used.
- **LSTM:** Long Short Term Memory [8] is a typical architecture of RNN (Recurrent Neural Networks). It employs the deep neural networks to learn vector representations for sentences and documents. Then the softmax classifier is adopted for sentiment classification.
- **GRU:** Gated Recurrent Unit [4] is also a typical architecture of RNN. It is similar with LSTM unit, however, without a separate memory cell. In the experiments, it's also used to learn the text representation, and then the softmax classifier is adopted for sentiment classification.

The kernel of SVMs is all set as linear kernel, since which has proven its effectiveness in text classification. For LSTM and GRU, the word embeddings are initialized with both random vectors and off-the-shelf word2vec vectors. The dimension of each word vector is 300. The number of hidden layer nodes of recurrent layers is set as 100. The training batch size for IMDB is set as 100.

We also evaluate the performance of our method for overall opinion sentences recognition in product review datasets, since some product data provides the detailed sentence annotation. We compare our method with some methods including the baseline methods.

- **MinCut:** the MinCut algorithm is utilized to extract important and representative sentences, and we take the result as the OOP sentences [26].
- **Position:** a baseline method which solely takes the first 1/10 and last 1/10 sentences of the reviews as OOP sentences.
- **LingRule:** a baseline method based on the linguistic features, we calculate the score of the sentences, and then 2/10 of the top ranked sentences are taken as the OOP sentences.
- *Semi_PLSA***:** a semisupervised topic modeling approach, which is used to model the targets with prior information [20]. In our experiments, the user-provided words for OOP sentence recognition are used as the prior terms for the Overall topic.
- SVM^{sle}**:** With SVM^{sle}, the detected hidden variables of each document are taken as the OOP sentences.

[7] http://mpqa.cs.pitt.edu/lexicons/subj_lexicon

Table 5 Experiment results of sentiment classification in product reviews, the evaluation metric is accuracy.

Methods	Oscar Data	Liu Data	McAuley Data
SVM	0.735	0.840	0.882
MinCut+SVM	0.750	0.820	0.895
SVM^{sle}	0.760	0.857	0.916
SVM^{eop}	**0.765**	**0.896**	**0.932**

Table 6 Experiment results of sentiment classification in movie reviews, the evaluation metric is accuracy.

Methods	IMDB(S)	IMDB(L)
SVM	0.8545 [21]	0.8690
MinCut+SVM	0.8715 [26]	0.8730
SVM^{sle}	0.8722 [39]	0.8833
JST	0.8460 [18]	0.8565
LSTM	-	0.8799
GRU	-	0.8823
SVM^{eop}	**0.8830**	**0.9011**

6.2.2 Evaluation

All the datasets except IMDB(L) are randomly partitioned into 10 parts with equal size, of which 8 parts are taken as the training data, 1 part is the validation set, and 1 part is used for testing. For IMDB(L), we randomly split the training set of 25000 examples into training and validation sets containing 22500 and 2500 examples respectively, as done in [21]. All the parameters as well as the baseline methods are fine tuned in the validation set.

The experiments are repeated 10 times, and all the methods are conducted under the same dataset setup during each time. The performance is measured by the average results. For document sentiment classification, accuracy is adopted as the evaluation metric. For the overall opinion sentence recognition, F-measure is adapted.

6.3 Experiment results

6.3.1 Sentiment classification results

Table 5 shows the document level sentiment classification results in product reviews[8]. In general, the results of our method are better than others in all the datasets.

In comparison with traditional SVM, SVM^{sle} and our method SVM^{eop} try to model the interaction between sentences and the sentiment of the documents, which leads to the improvement of the sentiment prediction. MinCut+SVM is also better than simply using SVM, which verifies that not all sentences in the documents are beneficial to sentiment prediction.

Our method SVM^{eop} is better than both SVM^{sle} and Mincut+SVM, since we specially take advantage of the overall opinion sentences. In comparison with the sentences extracted by SVM^{sle}, the overall opinion sentences are more deterministic for document sentiment prediction.

Table 6 shows the sentiment classification results in movie reviews. Some results of compared methods are collected from papers.

[8] The improvement of our method is significant, since with the paired t-test, $p < 0.05$.

Table 7 Experiment results of overall opinion sentence recognition, the evaluation metric is F-measure.

Methods	Oscar Data	Liu Data	McAuley Data
MinCut	0.38	0.41	0.47
Position	0.20	0.23	0.18
LingRule	0.48	0.51	0.55
Semi_PLSA	0.33	0.42	0.50
SVM^{sle}	0.51	0.59	0.58
SVM^{eop}	**0.62**	**0.66**	**0.70**

The SVM, MinCut+SVM, SVM^{sle}, and SVM^{eop} are based on the original feature representations, i.e. bag-of-words. From Table 6, we can find that, similar with the results in product reviews, both in IMDB(S) and IMDB(L), SVM^{eop} achieves the best accuracy in comparison with SVM, Mincut+SVM and SVM^{sle}. This also demonstrates the impact of overall opinions for document sentiment as well as the effectiveness of our method.

JST is a topic model extended from LDA, and the sentiment of document is inferred through sentiment topic distributions. Since it's unsupervised, a sentiment lexicon is utilized to help make better sentiment prediction. However, the results in Table 6 are still not comparable with all the other supervised methods. That's reasonable, for topic model, it's hard to distinguish positive topics from negative ones very well. Therefore, it's not superior in sentiment classification task.

RNN-based methods aim to learn representations of the documents for sentiment classification. Usually, a large dataset is needed to learn the parameters of them, therefore, we compare our method with LSTM [9] and GRU [10] in the IMDB(L) data. We can observe from Table 6 that GRU is slightly better than LSTM, however they don't show significant advantages in comparison with the methods based on bag-of-words. In other words, the methods which simply use the unigram words as features achieve satisfactory performance on the IMDB dataset, though they lose order information in comparison with LSTM and GRU. This may be because the order information is not very significant for long reviews like the IMDB dataset. Another possible reason is that the learned representation in IMDB with general LSTM or GRU is not competitive enough for sentiment classification, we should design more complex RNN architectures which sufficiently consider the influence of underlying sentences for document sentiment.

It turns out that our method achieves the best performance in both product and movie reviews. The experiment results demonstrate that capturing overall opinion information enhances the classification accuracy. With our method, the overall opinion sentences can be effectively recognized and utilized for sentiment classification.

6.3.2 Overall opinion sentences recognition results

To evaluate our method for overall opinion recognition, we compare our method with some other methods including the baseline methods in product reviews. Table 7 shows the general results. For **Liu Data**, which has the aspect annotations about all sentences, the other two datasets are evaluated by human judgement. In general, our method gets the best results.

MinCut only captures the sentences with higher weights in graph, which does not separate overall and aspect opinion sentences. The *Position* method simply takes the first and last

[9] The hyperparameter tuning has great influence on the performance of LSTM, we follow the work [41] [7], and the result is comparable with [41] and better than [7].

[10] The architecture of GRU differs in many works, e.g. [35] [41] [7], we follow the work [41] [7], and after fine tuning, the result is better than both of them.

Table 8 Examples of the overall opinion sentences in product reviews. The first column is the product, the second column is the example extracted OOPs in the reviews, and the last column is the position of the sentence in that review, e.g. 1/6 means that the sentence is the first sentence in the review, and the review length is 6 sentences.

product	sentences	position
Nokia 6600	1. Seriously an awesome phone!!!	[title]
Noika 6600	2. I've had this beauty for nearly 2 months now and I truely love it.	1/6
iPod	3. Overall, I think the iPod is a good player, although I would note some features that turned me off:	3/23
iPod	4. All in all, this is a wonderful device.	22/25
iPod	5. Overall the iPod really is an almost flawless beast.	64/69
Canon S100	6. I highly recommend the S100 Digital Elph!	8/8
Canon S100	7. Overall this is a great camera.	20/25
Canon S100	8. It's a beautiful thing!	16/16

several opinion sentences as the OOP sentences, which is not precise enough. Though many overall opinions are in the first or the end, however, not all the sentences in these places are OOP sentences. For both *Mincut* and *Position*, little linguistic knowledge has been exploited.

With *LingRule*, those explicit opinion sentences are easy to find, while implicit ones are hard to get. For *Semi_PLSA*, it uses the prior terms for indicating the overall aspect, however, that's still not robust due to the disadvantage of topics models. For both *LingRule* and *Semi_PLSA*, the positional information is not efficiently utilized.

Our method combines the linguistic features and positional features for generating candidate OOP sentences. Moreover, SVM^{eop} formulates the final OOP recognition together with the document level sentiment classification, which leads to a better result. This is also the reason why our method is superior than SVM^{sle}, though it explores the hidden explanations for document sentiment, however, it lacks the explicit discrimination between overall opinions and other opinion sentences.

We also give some example results in Table 8 to verify the quality of the extracted sentences. Obviously, the linguistic clues help the recognition of the sentence 3, 4, 5,7, the phrase 'overall' and 'all in all' are very discriminative. The recognition of sentence 2 and 8 are most possibly relied on the position signals. The recognition of sentence 1 and 6 is based on the combination of the phrases, e.g. 'phone', 'S100', and position information.

6.3.3 Parameter setup and tuning

The parameter $|S|$ adjusts the number of extracted overall opinion sentences for document sentiment analysis and is fine tuned in the validation dataset. In our experiments, the value of $|S|$ ranges in $[1,7]$. The optimal value is obtained when the best performance in development set is achieved. Figure 2 shows the sentiment classification results with different $|S|$ during the parameter tuning in **Liu Data** and **McAulley Data**.

We can find that, if $|S|$ is too large, the classification results gradually decrease, because some sentences are mistaken as OOP ones which have a negative impact in final sentiment classification result. However, if the $|S|$ is too small, it also affects the final classification result. That's reasonable, because sometimes a review may contain several overall opinion sentences, and the polarity of them could also be conflict, therefore, it's crucial to well explore the overall opinion for final sentiment prediction.

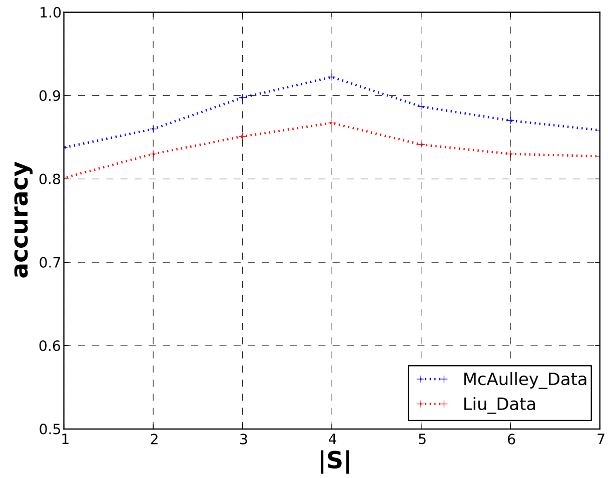

Fig. 2 The accuracy in the development set with different size of the S, i.e the number of expected extracted OOP sentences.

6.3.4 Computation cost for training

We also evaluate the computation cost for our method, since for big data, time efficiency is a very important issue. Theoretically, the initialization with well-generated candidate OOP sentences will make our method convergence more faster than SVM^{sle} since which should search all possible sets during the training.

Here we take the experiment on **McAulley Data** as an example. Figure 3 shows the time cost of our method in comparison with SVM^{sle}. The data size ranges in the interval of 1000*[1,2,...,10]. When the data size becomes large, our method is quite efficient. The main reason is that for each iteration, when the data size become large, the solution space becomes very large, however, with our method, well-generated candidate OOP sentences set greatly reduce the search space. This result also validates the efficiency of our method for big data.

6.3.5 Case study

In order to digest the experiment results more intuitively, we pick several example reviews in Table 9 for further study. The sentiment classification results and the extracted overall opinions are accurate with our method.

For all the reviews, though most of the opinion sentences towards the mentioned aspects are negative, however the overall sentiment about the whole target is positive. The aspect opinion sentences greatly mislead the overall sentiment prediction with existing methods. Take the review of iPod as an example, in the beginning of the review, the user is satisfied with iPod, however, all the remanding texts are the opinions towards the unsatisfied aspects. It's almost impossible to correctly classify sentiment of the reviews with existing methods, however, our method can still make the correct prediction.

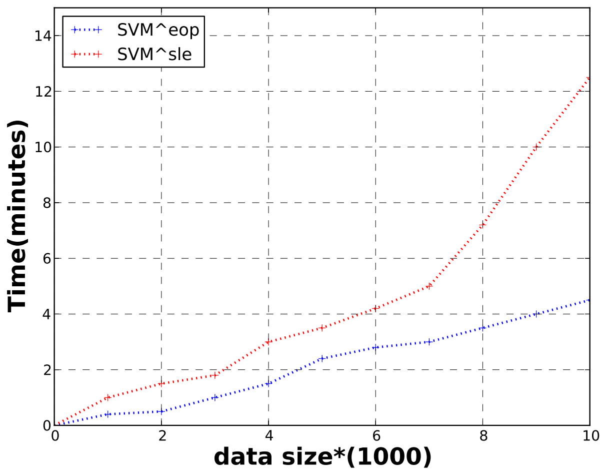

Fig. 3 The running time of our proposed SVM^{eop} in comparison with the SVM^{sle}. The horizontal axis indicates the data size ranging from 10^3 to 10^4, and the vertical axis indicates the computation time.

Table 9 The example reviews and results with our method. The first review is about Nomad MP3 player, the second one is for cellphone, and the second review is about iPod. The extracted overall opinion sentences are in bold. [title] indicates that the sentence is the title. [O] stands for overall opinion, and [A] stands for detailed aspect opinion. The label '+' is for positive opinion, while '-' is for negative opinion.

Reviews
[O, +][title] **excellent product** with a few minor problems. ... [O, +] I have had the nomad jukebox for about three weeks now, and **i am very happy with it**. [A, -] It 's only slightly heavier than the ipod, [A, +] and has a longer battery life. [A, +] the storage capacity is great for me – – i have a large but not huge cd collection and have loaded everything i want to listen to on it and still have 13 gigabytes free. [A, -] The controls are somewhat harder to use than the ipod ... [A, -] Loading cds was somewhat time-consuming ... [A, -] My only reservations ... the tagging process and the way it interacts with the software . ... [A, -] other tagging problems result from the nomad's operating system . ... [A, -] the software does not ignore "the" when it lists the cds in alphabetical order. [A, -] Finally , making playlists from the computer can be complicated ... [O, +] The bottom line for me is that **i am very happy with this product** . ...
[A, -][title] Screen is easily scratched. [O, +] **Overall a great phone.** [A, -] However, the screen is easily scratched. If you purchase this phone, leave plastic protector on screen, until you purchase a protective case. [A, -] My screen was scratched within a week.
[O, +][title] **I love My iPod!** Although iPods have been around for a few years, they didn't really get hot until now. I ended up (surprisingly) getting one for Christmas. [O, +] **I love the features on the iPod and the many things** you can do with it. It deserves a perfect 5-stars. [A, -] The only problem is the battery life. [A, -] After about 3-4 months, you see your battery draining faster than it should. [A, -] After a year, your battery is dead and you need to replace it with a new one. [A, -] Apple's iPod battery replacement service costs $100! Amazing. You pay $300 for the iPod itself, then $100 to get the battery fixed. If you look on Google or around the web's many search engines, you can find a site that will replace the battery at a cheaper price. [A, -] Apple needs to step it up and get better, longer lasting batteries.

6.3.6 Summary and discussion

The experiment results validate the capability of our method for both normal and abnormal reviews. It's is obviously superior to existing methods in abnormal reviews. The results also verify that the exploration of overall opinion sentences can benefit document sentiment classification.

In the movie reviews, we also compare our method with LDA based and RNN-based methods. JST is an extension of LDA, it models the review as a mixture of fine grained sentiment topics. In comparison with supervised methods, it's less effective in sentiment classification. The RNN-based methods aim to learn the representation of reviews and then make sentiment classification. Without special design, the general RNN-based architectures don't show significant advantages. In principle, the framework of our method is compatible with the RNN-based methods. For example, RNN-based sentence representations can be embedded into our model instead of bag-of-words representation.

7 Conclusion

In this paper, we explicitly point that overall opinion sentences play a more important role in determining document level sentiment, and present an effective method extended from structural SVM to utilize overall opinion for document level sentiment classification. With our proposed SVM^{eop} model, the overall opinion sentences are taken as the hidden variables for document sentiment. Multiple features are exploited to recognize candidate overall opinion sentences firstly, and then the candidate overall opinion sentences are incorporated as the initial value for SVM^{eop}, which ensures the accuracy and reduces the training time. Our method directly optimizes document level sentiment classification result, and the sentence level annotations are not needed. Experiments on public sentiment analysis datasets show the effectiveness of our method. In the future, our framework could be extended with some deep learning methods to achieve better results.

References

1. Baroni, M., Dinu, G., Kruszewski, G.: Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In: Proceedings of the ACL 2014, pp. 238–247. Association for Computational Linguistics (2014)
2. Becker, I., Aharonson, V.: Last but definitely not least: On the role of the last sentence in automatic polarity-classification (2010)
3. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)
4. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 (2014)
5. Ding, X., Liu, B., Yu, P.S.: A holistic lexicon-based approach to opinion mining. In: Proceedings of the 2008 International Conference on Web Search and Data Mining, WSDM '08, pp. 231–240. ACM, New York, NY, USA (2008). DOI 10.1145/1341531.1341561
6. Fang, Q., Xu, C., Sang, J., Hossain, M.S., Ghulam, M.: Word-of-mouth understanding: Entity-centric multimodal aspect-opinion mining in social media. IEEE Trans. Multimedia **17**(12), 2281–2296 (2015). URL http://dx.doi.org/10.1109/TMM.2015.2491019
7. Gao, Y., Glowacka, D.: Deep gate recurrent neural network. JMLR:Workshop and Conference Proceedings **6**, 350–365 (2016)
8. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
9. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '04, pp. 168–177. ACM, New York, NY, USA (2004). DOI 10.1145/1014052.1014073
10. Jakob, N., Gurevych, I.: 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, pp. 1035–1045. Association for Computational Linguistics (2010)
11. Jin, W., Ho, H.H., Srihari, R.K.: 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, pp. 1195–1204. ACM (2009)

12. Joachims, T., Finley, T., Yu, C.N.J.: Cutting-plane training of structural svms. Mach. Learn. **77**(1), 27–59 (2009). DOI 10.1007/s10994-009-5108-8. URL http://dx.doi.org/10.1007/s10994-009-5108-8
13. Kiritchenko, S., Zhu, X., Mohammad, S.M.: Sentiment analysis of short informal texts. J. Artif. Int. Res. **50**(1), 723–762 (2014)
14. Labutov, I., Lipson, H.: Re-embedding words. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pp. 489–493 (2013)
15. Lakkaraju, H., Bhattacharyya, C., Bhattacharya, I., Merugu, S.: Exploiting coherence for the simultaneous discovery of latent facets and associated sentiments. In: Proceedings of the 2011 SIAM International Conference on Data Mining (2011)
16. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: Proceedings of The 31st International Conference on Machine Learning, pp. 1188–1196 (2014)
17. Li, J., Luong, M.T., Jurafsky, D.: A hierarchical neural autoencoder for paragraphs and documents pp. 1106–1115 (2015)
18. Lin, C., He, Y.: Joint sentiment/topic model for sentiment analysis. In: Proceedings of the 18th ACM conference on Information and knowledge management, pp. 375–384. ACM (2009)
19. Liu, B.: Sentiment analysis and opinion mining. Synthesis Lectures on Human Language Technologies **5**(1), 1–167 (2012). DOI 10.2200/S00416ED1V01Y201204HLT016
20. Lu, Y., Zhai, C.: Opinion integration through semi-supervised topic modeling. In: Proceedings of the 17th International Conference on World Wide Web, WWW '08, pp. 121–130. ACM, New York, NY, USA (2008). DOI 10.1145/1367497.1367514. URL http://doi.acm.org/10.1145/1367497.1367514
21. Maas, A.L., Daly, R.E., Pham, P.T., Huang, D., Ng, A.Y., Potts, C.: Learning word vectors for sentiment analysis. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pp. 142–150. Association for Computational Linguistics, Portland, Oregon, USA (2011)
22. McAuley, J., Leskovec, J.: Hidden factors and hidden topics: understanding rating dimensions with review text. In: Proceedings of the 7th ACM conference on Recommender systems, pp. 165–172. ACM (2013)
23. McDonald, R., Hannan, K., Neylon, T., Wells, M., Reynar, J.: Structured models for fine-to-coarse sentiment analysis. In: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pp. 432–439. Association for Computational Linguistics, Prague, Czech Republic (2007)
24. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: C. Burges, L. Bottou, M. Welling, Z. Ghahramani, K. Weinberger (eds.) Advances in Neural Information Processing Systems 26, pp. 3111–3119. Curran Associates, Inc. (2013)
25. Moghaddam, S., Ester, M.: On the design of lda models for aspect-based opinion mining. In: Proceedings of the 21st ACM International Conference on Information and Knowledge Management, CIKM '12, pp. 803–812. ACM, New York, NY, USA (2012). DOI 10.1145/2396761.2396863. URL http://doi.acm.org/10.1145/2396761.2396863
26. Pang, B., Lee, L.: A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In: Proceedings of the 42th Annual Meeting of the Association for Computational Linguistics(ACL) (2004)
27. Pang, B., Lee, L.: Opinion mining and sentiment analysis. Foundations and trends in information retrieval **2**(1-2), 1–135 (2008)
28. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up?: sentiment classification using machine learning techniques. In: Proceedings of the ACL-02 conference on empirical methods in natural language processing-Volume 10, pp. 79–86. Association for Computational Linguistics (2002)
29. Sang, J.: User-centric cross-osn multimedia computing. In: Proceedings of the 23rd Annual ACM Conference on Multimedia Conference, MM '15, Brisbane, Australia, October 26 - 30, 2015, pp. 1333–1334 (2015). URL http://doi.acm.org/10.1145/2733373.2807423
30. Sang, J., Xu, C.: Browse by chunks: Topic mining and organizing on web-scale social media. ACM Transactions on Multimedia Computing, Communications, and Applications **7**(1), 30 (2011)
31. Sang, J., Xu, C.: Right buddy makes the difference: An early exploration of social relation analysis in multimedia applications. In: ACM International Conference on Multimedia, pp. 19–28. ACM (2012)
32. Sang, J., Xu, C., Liu, J.: User-aware image tag refinement via ternary semantic analysis. IEEE Transactions on Multimedia **14**(3), 883–895 (2012)
33. Taboada, M., Brooke, J., Tofiloski, M., Voll, K., Stede, M.: Lexicon-based methods for sentiment analysis. Computational linguistics **37**(2), 267–307 (2011)
34. Täckström, O., McDonald, R.: Discovering fine-grained sentiment with latent variable structured prediction models. In: Advances in Information Retrieval, pp. 368–374. Springer (2011)
35. Tang, D., Qin, B., Liu, T.: Document modeling with gated recurrent neural network for sentiment classification. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, p. 1422C1432. Association for Computational Linguistics (2015)

36. Tang, D., Wei, F., Yang, N., Zhou, M., Liu, T., Qin, B.: Learning sentiment-specific word embedding for twitter sentiment classification. In: Proceedings of the ACL 2014, p. 1555C1565 (2014)
37. Turney, P.: Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics(ACL), pp. 417–424 (2002)
38. Wang, S., Manning, C.: Baselines and bigrams: Simple, good sentiment and topic classification. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp. 90–94. Association for Computational Linguistics, Jeju Island, Korea (2012)
39. Yessenalina, A., Yue, Y., Cardie, C.: Multi-level structured models for document-level sentiment classification. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, pp. 1046–1056. Association for Computational Linguistics, Cambridge, MA (2010)
40. Yu, C.N.J., Joachims, T.: Learning structural svms with latent variables. In: Proceedings of the 26th Annual International Conference on Machine Learning, pp. 1169–1176. ACM (2009)
41. Zhang, Y., Er, M.J., Venkatesan, R., Wang, N., Pratama, M.: Sentiment classification using comprehensive attention recurrent models. In: 2016 International Joint Conference on Neural Networks (IJCNN), pp. 1562–1569 (2016). DOI 10.1109/IJCNN.2016.7727384