Sentiment Analysis with the Exploration of Overall Opinion Sentences

Xiaojia Pu, Gangshan Wu and Chunfeng Yuan
State Key Laboratory for Novel Software Technology
Department of Computer Science and Technology
Nanjing University, Nanjing 210023, China
Email: puxiaojia@gmail.com, {gswu, cfyuan}@nju.edu.cn

***Abstract*—With the rapid growth of opinionated contents, e.g. product reviews, sentiment analysis has drawn much attention from the researchers. The most fundamental task of sentiment analysis is document sentiment classification which aims to predict the overall sentiment (e.g. positive or negative) towards the opinion target in a review. There are usually various opinion sentences towards different aspects with different sentiments. Among them, the overall opinion towards the whole target should be more deterministic in document sentiment prediction. However, most existing methods treat all the sentences equally, thus, they may encounter difficulty especially when the sentiments of most aspect opinion sentences differ from the overall sentiment. To address this, we propose a novel method for document sentiment classification which adequately explores the effect of overall opinion sentences. The method is extended from structural SVM, and the overall opinion sentences are taken as the hidden variables for document sentiment. Experiments on several standard product review datasets show the effectiveness of our method.**

I. Introduction

With the advent of Web 2.0 and social networks, people become more and more convenient to express and share their opinions on web. These large volume of opinionated contents, e.g. product reviews, are generated rapidly and of great value. In order to effectively explore these data, sentiment analysis has been emerging as a hot research area [1] [2]. As a fundamental task of sentiment 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 [2].

The existing methods for document level sentiment classification could be divided into two categories, i.e. unsupervised approach and supervised approach [2]. The unsupervised approach utilizes a predefined sentiment lexicon and some linguistic rules to determine the sentiment of the reviews [3] [4]. This approach is simple, but suffers from scalability, since the sentiment lexicons and linguistic rules are commonly manually defined by experts. Instead, supervised approach takes the task as a special case of text classification, which usually represents documents with bag-of-ngrams features and build SVM classifier upon that [5] [6].

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

Take the review in Table I for example, it mentions several aspects, and complains about the small size of the memory card and pictures in the dark, however 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 [7], in which the OOP sentences are taken as the hidden variables for document sentiment. The structural SVM could conduct the hidden variable recognition and final classification simultaneously. The main difficulty of structural SVM is the initialization of the hidden variable, which greatly influence the training time and the accuracy [7]. In order to resolve this problem, we exploit multiple features to recognize candidate OOP sentences firstly, and then the candidate OOP sentences will be incorporated as the initial value which dramatically reduces the training time for structural SVM. We conduct document sentiment classification on several benchmark datasets, and the experiments demonstrate the effectiveness of our method.

The contribution of our work could be summarized as follows:

978-1-5090-6182-2/17/$31.00 ©2017 IEEE

TABLE I
A REVIEW OF DIGITAL CAMERA CANON S100. 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**.

1. We propose an effective method to explore the OOP sentences for document level sentiment classification which improve the classification results.

2. We combine multiple features to recognize candidate OOP sentences, which ensures the accuracy for subsequent procedures.

The rest of the paper is organized as follows. Section II overviews the related work. In section IV, IV and V, the detailed methods are described. In section VI, the experiments and results are presented. Finally, section VII concludes the paper and discusses future work.

II. RELATED WORK

Document level sentiment classification is a fundamental problem in sentiment analysis, which aims to identifying the sentiment label of a document [1] [2]. 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 [3] [8] [9] [4]; 2) machine learning based approach [5] [10] [11] [12] [13].

The lexicon based sentiment classification approach utilizes a predefined sentiment lexicon and some linguistic rules to determine the sentiment of the reviews [2] [3] [8] [4]. 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 [5] [6]. The documents are usually represented as bag of features. [5] 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. Dominant studies follow [5] and work on designing effective models and features for building a powerful sentiment classifier. Representative features include word ngrams [6], sentiment lexicon features [14]. [6] 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.

Another line of research concentrates on modeling the semantic relationship between the document and sentences. [10] separated subjective portions from the objective text by finding minimum cuts in graph of sentences to achieve better sentiment classification performance. [11] investigated a global structured model for jointly classifying sentiment polarity at different levels of granularity. [12] used sentence-level latent variables to improve document level sentiment prediction. However, they are still lack of taking into account of overall opinion sentences.

III. 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 a classifier on the OOP sentences is enough for our task. However, it's not that easy. First, the state-of-art aspect extraction techniques are still not perfect and usually time consuming. Supervised learning methods usually need sentence level annotation [15] [16] and unsupervised methods usually produce incoherent aspects [17] [18]. Second, not all the reviews contain the OOP sentences, relying entirely on OOP sentences is also dangerous.

Therefore, we take advantage of structural SVM, and the OOP sentences are taken as the hidden variables for document sentiment. The structural SVM could conduct the hidden variable recognition and the final classification simultaneously [7]. The difficulty is to give a good initialization of the hidden variables, without which, the structural SVM needs a lot of time to achieve convergence and sometimes the result is not accurate enough. 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.

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 II.

TABLE II
BASIC NOTATIONS USED IN THE PAPER.

Variable	Description
x	a review document
$len(x)$	the length (number of sentences) of a review document
y	the predicted polarity of the document
x_i	a review sentence
y_i	sentiment polarity of a sentence
S_x	the set of total sentences in x
S	the set of overall opinion sentences
w	the parameter of the structural SVM model

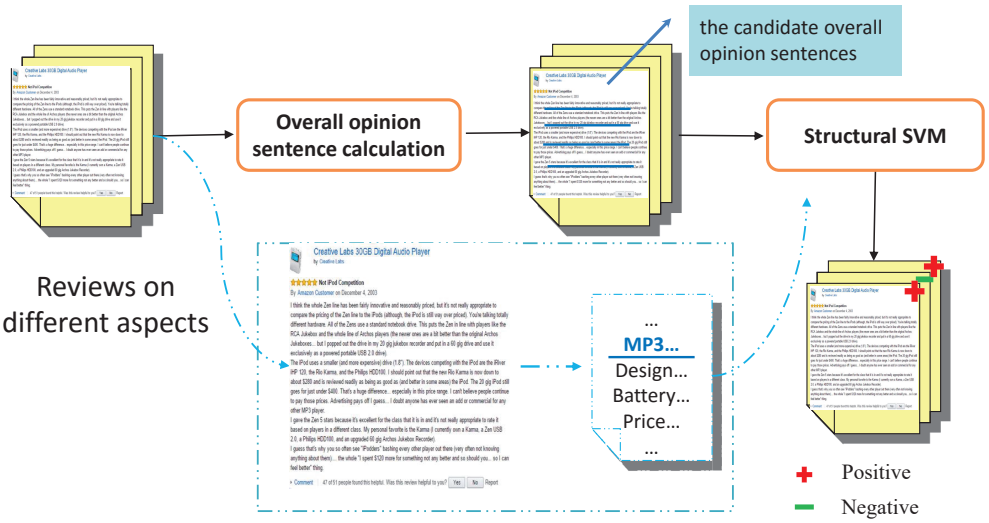

Fig. 1. The overview of our method, firstly, the candidate overall opinion sentences are recognized for model initialization, and then the structural SVM is used to explore the overall opinion for document level sentiment classification.

TABLE III
THE FEATURES FOR RECOGNIZING CANDIDATE OVERALL OPINION SENTENCES.

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

IV. PROBABILISTIC METHOD FOR RECOGNIZING CANDIDATE OVERALL OPINION SENTENCES

To recognize candidate overall opinion sentences, all the possibilities of the OOP expressions should be considered. Besides, since our final target is the document sentiment classification, the method should be efficient.

To this end, multiple features are exploited including the linguistic features and positional features. A function is utilized to calculate the probability of each sentence, and then the sentences with high scores will be selected as the candidate overall opinions.

A. Exploiting linguistic and positional features

There are two kinds of clues could be exploited for the recognition of overall opinion sentences, including the linguistic and positional features. Table III illustrates the all used features.

1) Linguistic features: The overall opinions imply the 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', '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.

Besides, conclusive words such as 'overall' directly indicate the OOP sentences, which are good heuristic information for our task. In practice, these words or phrases could be easily collected.

All the linguistic features compose a dictionary Dic of indicative words or phrases.

2) Positional features: The overall opinion sentences could appear in various positions, e.g. in the title, in the beginning of the document, in the middle, and in the end of the document. It is reasonable that the sentences in the beginning or in the end should be of higher probability to be as overall opinion as well as the title.

Some 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 [19].

3) Feature quantification: For linguistic features, the KL divergence is calculated between each sentence and the provided indicative words or phrases, and then the values are normalized.

For positional features, f_{posi} and f_{title} are listed below. With f_{posi}, the sentence that appears in the beginning or the end, will get larger score than those sentences in the middle. f_{title} means that the sentence in the title will be with high probability to be the OOP sentences.

$$f_{posi}(x_i) = max\{\frac{posi(x_i)}{len(x)}, 1 - \frac{posi(x_i)}{len(x)}\} \quad (1)$$

$$f_{title(x_i)} = \begin{cases} 1 & x_i\, is\, title \\ 0 & x_i\, is\, not\, title \end{cases} \quad (2)$$

B. Probability function

The general idea is to measure the probability about each sentence as overall opinion sentence. Specifically, 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, i.e.

$$\phi(x_i) = \{KL(x_i, Dic_1), \ldots, f_{posi}(x_i), f_{title}\{x_i\}\} \quad (3)$$

We calculate the score as

$$f_{oop}(x_i; w_{oop}) = w_{oop} \cdot \phi(x_i) \quad (4)$$

In the function, the two kinds of features illustrated in last section will be incorporated as the parameters. The parameter $\phi(x_i)$ will be numerated feature vector, and the dimension is the amount of all features.

Here, we set the total weight of the two kinds of features equally as $\frac{1}{2}$. Then, for each category, the feature are quantified, and the weight will divided equally for all possible elements. For linguistic features, that will be $\frac{1/2}{|Dic|}$. For the two position features, that will be $\frac{1}{4}$ respectively.

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

V. 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).

A. Preliminary of Structural SVM

Structural SVM is an extension of traditional SVM for interdependent and structured output spaces [7].

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

$$f_w(x) = arg \max_{y \in \mathrm{y}} [w \cdot \Phi(x, y)] \quad (5)$$

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 many possible structures $\hat{y}$ in the output spaces y, it is usually replaced with a piecewise linear convex upper bound

$$\Delta(y_i, \hat{y}_i(w)) \leq \max_{\hat{y} \in \mathrm{y}} [\Delta(y_i, \hat{y}) + w \cdot \Phi(x_i, \hat{y})] - w \cdot \Phi(x_i, y_i) \quad (6)$$

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, we extend our joint feature vector $\Phi(x, y)$ 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

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

B. 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

$$F(x, y, S) = w^T \Psi(x, y, S), \quad (8)$$

and a document level sentiment classifier as

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

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 of $F(x, y, S; w) = w^T \Psi(x, y, S)$,

$$= \frac{1}{N(x)} \sum_{j \in S} y \cdot w_{pol_{oj}}^T \psi_{pol}(x^j) + w_{subj}^T \psi_{subj_o^j}(x^j), \quad (10)$$

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 opinion sentences, and $N(x)$ is a normalization factor.

We represent the weight vector as

$$w = [w_{pol_o}; w_{subj_o}], \quad (11)$$

and $\psi_{pol_o} x^j$ denotes the polarity features of sentence x_j with overall opinion, $\psi_{subj_o} x^j$ denotes the subjectivity features of sentences x^j with overall opinion.

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.

C. Document sentiment prediction

The model will try to predict the sentiment with the best S, i.e. the extracted overall opinion sentences. S will be used for help predicting document sentiment polarity, we have the document level sentiment classifier as

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

Initialization of the S

The initialization of the overall opinion sentences set S will be the candidate OOP sentences calculated in section 4.

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

D. Updating of overall opinion sentences

For the document sentiment prediction, the finding of S is essential. This task is accomplished with an iterative approach.

With the initial S, the document prediction could be done and the initial parameter w could be learned as described in section 5.5, and then new overall opinion sentences set S could be resolved.

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

$$score(x^j, S, y) = y \cdot w_{pol_o}^T \psi_{pol}(x^j) + w_{subj_o}^T \psi_{subj}(x^j). \quad (13)$$

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

E. Learning algorithm

The learning process is to optimize the following problem below.

$$\begin{aligned} &\min_{w, \xi \geq 0} \frac{1}{2}||w||^2 + \frac{C}{N}\sum_i^N \xi_i \\ &s.t. \forall i \max_{S_i \subset S_x} w_s \cdot \Psi(x_i, (y_i, S_i)) \\ &\geq \max_{S_i' \in S(x)} w_s \cdot \Psi(x_i, (-y_i, S_i')) + \Delta(y_i, -y_i, S_i') - \xi_i \end{aligned} \quad (14)$$

where C is the regularization parameter.

This is a SVM style objective function, for each training instance, the corresponding constraint is quantified over the best possible OOP sentence sets S_i. The S_i is modeled as a latent variable. Since it is non-convex, we try to solve it using CCCP algorithm [7].

The candidate sentence set are consisting of those sentences with high scores as talked in section 4. With the well generated candidate OOP sentence set, the algorithm will try to make refinement of the set according to the sentiment prediction in the training dataset.

Starting with candidate OOP sentences for each training instance, 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.

The detailed algorithm of our method is shown in Algorithm 2.

The normalizing factor is set as $N(x) = \sqrt{(|S|)}$ as described in [12], where $|S|$ is the size of the extracted candidate overall opinion sentences, and will be further discussed in the experimental section.

Algorithm 2 The Detailed Training Algorithm

Input:
{ (x_1, y_1), ..., (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**
        $s_i \leftarrow arg\,max_{S \subset S(x_i)}\, w^T \Psi(x_i, y_i, S)$
    end for
    $w \leftarrow SSVMSolve(C, \{(x_i, y_i, S_i)\}_{i=1}^N)$
end while
Return w

VI. Experiments

We conduct experiments on several benchmark sentiment classification datasets to evaluate our proposed method. The algorithms are implemented using C++, and run in a PC with Intel Core i5 CPU and 8GB RAM.

A. Datasets

Three benchmark datasets are used for our experiments, i.e. **Oscar Data** [20], **Liu Data** [8] [9], and **McAuley Data** [21], which are all comprised of product reviews from Amazon. The details are listed below.

Oscar Data[1]: The dataset contains some reviews about books, dvds, electronics, music, and videogames.

Liu Data [2]: It contains fine annotated datasets, for each review, the sentences are labeled with aspect information. We choose digital product reviews for our experiments.

McAuley Data [3]: This is a huge dataset of product reviews, we choose more than 10,000 cellphone reviews for our experiments.

Table IV summarizes the datasets statistics. All datasets are processed using lowercased stemmed unigram words, and the stop-words are removed. The documents are represented by the vectors of words. We choose 0/1 for term weighting, which is widely used in sentiment classification [2].

[1]https://github.com/oscartackstrom/sentence-sentiment-data
[2]https://www.cs.uic.edu/ liub/FBS/sentiment-analysis.html
[3]http://snap.stanford.edu/data/web-Amazon.html

TABLE IV
SOME STATISTICS OF THE DATASETS.

dataset	reviews	positive	negative
Oscar Data	196	97	99
Liu Data	489	301	188
McAuley Data	13448	7615	4233

TABLE V
EXPERIMENT RESULTS OF DOCUMENT LEVEL SENTIMENT CLASSIFICATION (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

B. Experimental setup

1) Methods for comparison: For the document level sentiment classification, we compared our method SVM^{eop} with various methods illustrated below.

- SVM This is the most common method for this task.
- MinCut The minimum cut algorithm is utilized to extract important and representative sentences and then SVM could be applied for sentiment classification [10].
- SVM^{sle} This is a typical method based on structural SVM for sentiment classification, which also considers the interactions between the sentences and documents [12], but has no explicit difference between OOP sentences and other opinion sentences .

The kernel of SVM in these methods is set as linear kernel, which is effective in text classification.

To evaluate the performance of our method for overall opinion sentences recognition, we compare our method with some baselines as well as some existing methods.

- MinCut The minimum cut algorithm is utilized to extract important and representative sentences, and we take the result as the OOP sentences [10].
- Position This is the baseline method, the first 1/10 and last 1/10 sentences of the reviews as taken as OOP sentences.
- LingRule This is also the 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.
- $Semi\_PLSA$ The $Semi\_PLSA$ is used to model the targets with supervision information [22]. In this paper, the user provided words for OOP sentence recognition are used as the prior terms for Overall aspect.
- SVM^{sle} We take the final outputed hidden sentences as the OOP sentences.

2) Evaluation: Each dataset is randomly equally partitioned into 10 parts, of which 8 parts are taken as the training data, 1 part is the development set, and 1 part is used for testing. All the parameters as well as the baseline methods are fine tuned in the development set.

For each dataset, the experiments are repeated 10 times, and all the methods are conducted under the same setup during each time. The performance is measured by the average results of 10 times.

For document sentiment classification, accuracy is adopted as the evaluation metric. For the overall opinion sentence recognition, F-measure is adapted.

C. Experiment results

1) Document level sentiment classification results: Table V shows the document level sentiment classification results of our method[4]. In general, we can find that the results of our method are better than others in all the datasets.

In comparison with traditional SVM, SVM^{sle} and our method try to model the interaction between sentences and the sentiment of the documents, which lead to the improvement of the sentiment prediction. MinCut is also litter better than directly using SVM, which also indicates that not all sentences in the documents are useful for sentiment prediction.

Our method SVM^{eop} is better than both SVM^{sle} and Mincut, 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 polarity prediction.

This results demonstrate that the overall opinion is highly useful for document level sentiment classification. With our method, the overall opinion sentences could be effectively recognized and utilized, which could resolve the problem we proposed in the introduction section effectively.

2) Overall opinion sentences recognition results: To evaluate our method for overall opinion recognition, we compare our method with some baselines as well as some existing methods.

Table VI show the general results from different methods, e.g. $MinCut$, $Position$, $LingRule$, $Semi\_PLSA$. For **Liu Data**, which has the aspect annotation about each sentence, the other two datasets are evaluated by human judgement. In general, our method could get better results.

$MinCut$ could only capture those sentences with higher weights in the graph, which does not separate overall (general) and aspect sentences. The $Position$ method simply takes the first and last several opinion sentences as the OOP, 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 OOPs. For both $Mincut$ and $Position$, little linguistic knowledge has been exploited for the task.

With $LingRule$, the explicit opinion sentence could be found, while the implicit ones are easy to be ignored. 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.

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

TABLE VI
EXPERIMENT RESULTS OF OVERALL OPINION SENTENCE RECOGNITION (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

TABLE VII
EXAMPLES OF THE OVERALL OPINION SENTENCES.

sentences	position
1. Seriously an awesome phone!!!	[title]
2. ... and I truely love it.	1/6
3. Overall, I think the iPod is a good player...	3/23
4. All in all, this is a wonderful device.	22/25
5. Overall the iPod really is an almost flawless beast.	64/69
6. I highly recommend the S100 Digital Elph!	8/8
7. Overall this is a great camera.	20/25
8. It's a beautiful thing!	16/16

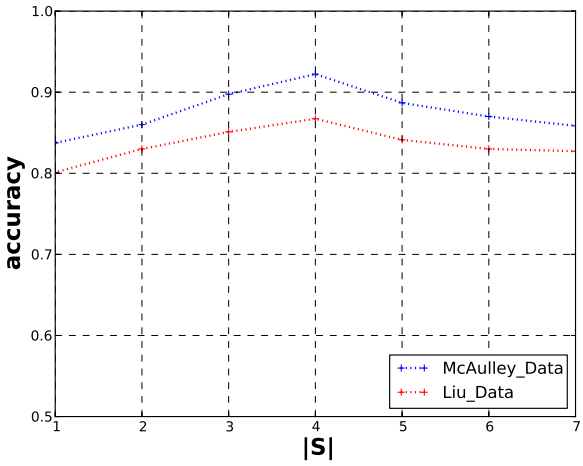

Fig. 2. The accuracy in the development set with different size of the S.

Our method combines the linguistic features and positional features for generating OOP sentence recognition. 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 has tried to explore the hidden explains for document sentiment, however, it lacks the explicit discrimination between overall opinions and other opinion sentences.

We also give some example results in Table VII to verify the quality of the extracted sentences. The first column is the extracted OOPs in the reviews, and the second 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. 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 maybe is the combination of the phrases, e.g. 'phone', 'S100', and position information.

3) Parameter setup and tuning: The parameter $|S|$ adjusts the number of extracted overall opinion sentences for our model and is fine tuned in the development dataset.

In the 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 will gradually become worse, since some sentences which may be not OOP sentences are utilized for determining the polarity of the document. This may have a negative impact in final sentiment classification result. However, if the $|S|$ is too small, it will also affect the final classification result. That's reasonable, because sometimes a review may contain several overall opinion sentences, and the polarity of the overall opinions could also be conflict, therefore, it's crucial to well explore the overall opinion for final sentiment prediction.

4) Case study: Several example reviews are collected in Table VIII for further study. The first review is about Nomad MP3 player and the second review is about iPod. The extracted overall opinion sentence/expressions 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.

We can find that our method gives accurate results for both reviews. Most of the aspect opinion sentences in the reviews are negative, however the overall sentiment is positive. The aspect opinion sentences greatly mislead the overall sentiment prediction. It's almost impossible to correctly classify sentiment of the reviews with existing methods, however, our method can still make the correct prediction. In fact, our method could perform well in both normal and abnormal reviews.

VII. CONCLUSION

In this paper, we explicitly pointed that overall opinion sentences play a more important role in determining document level sentiment, and presented an effective method based on structural SVM to utilize overall opinion for sentiment analysis. With our method, the overall opinion sentences are taken as the hidden variables for document sentiment. Multiple features are exploited to recognize candidate overall opinion sentences for the model initialization which ensures the accuracy. Experiments on benchmark sentiment analysis datasets showed improved performance over previous results. In the future, we will provide an automatic approach to decide the numbers of candidate overall opinion sentences to make our method more efficient.

TABLE VIII
THE EXAMPLE REVIEWS AND CORRESPONDING RESULTS.

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** . ...
[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.

ACKNOWLEDGMENT

This research is supported by the National Science Foundation of China under Grant No.61321491 and No.61223003, and Collaborative Innovation Center of Novel Software Technology and Industrialization.

REFERENCES

[1] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Foundations and trends in information retrieval*, vol. 2, no. 1-2, pp. 1–135, 2008.

[2] B. Liu, "Sentiment analysis and opinion mining," *Synthesis Lectures on Human Language Technologies*, vol. 5, no. 1, pp. 1–167, 2012.

[3] P. Turney, "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)*, 2002, pp. 417–424.

[4] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede, "Lexicon-based methods for sentiment analysis," *Computational linguistics*, vol. 37, no. 2, pp. 267–307, 2011.

[5] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: sentiment classification using machine learning techniques," in *Proceedings of the ACL-02 conference on empirical methods in natural language processing-Volume 10*. Association for Computational Linguistics, 2002, pp. 79–86.

[6] S. Wang and C. Manning, "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)*. Jeju Island, Korea: Association for Computational Linguistics, July 2012, pp. 90–94.

[7] C.-N. J. Yu and T. Joachims, "Learning structural svms with latent variables," in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009, pp. 1169–1176.

[8] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '04. New York, NY, USA: ACM, 2004, pp. 168–177.

[9] X. Ding, B. Liu, and P. S. Yu, "A holistic lexicon-based approach to opinion mining," in *Proceedings of the 2008 International Conference on Web Search and Data Mining*, ser. WSDM '08. New York, NY, USA: ACM, 2008, pp. 231–240.

[10] B. Pang and L. Lee, "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.

[11] R. McDonald, K. Hannan, T. Neylon, M. Wells, and J. Reynar, "Structured models for fine-to-coarse sentiment analysis," in *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Prague, Czech Republic: Association for Computational Linguistics, June 2007, pp. 432–439.

[12] A. Yessenalina, Y. Yue, and C. Cardie, "Multi-level structured models for document-level sentiment classification," in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Cambridge, MA: Association for Computational Linguistics, October 2010, pp. 1046–1056.

[13] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA: Association for Computational Linguistics, June 2011, pp. 142–150.

[14] S. Kiritchenko, X. Zhu, and S. M. Mohammad, "Sentiment analysis of short informal texts," *J. Artif. Int. Res.*, vol. 50, no. 1, pp. 723–762, May 2014.

[15] W. Jin, H. H. Ho, and R. K. Srihari, "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, 2009, pp. 1195–1204.

[16] N. Jakob and I. Gurevych, "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, 2010, pp. 1035–1045.

[17] H. Lakkaraju, C. Bhattacharyya, I. Bhattacharya, and S. Merugu, "Exploiting coherence for the simultaneous discovery of latent facets and associated sentiments." in *Proceedings of the 2011 SIAM International Conference on Data Mining*, 2011.

[18] S. Moghaddam and M. Ester, "On the design of lda models for aspect-based opinion mining," in *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, ser. CIKM '12. New York, NY, USA: ACM, 2012, pp. 803–812. [Online]. Available: http://doi.acm.org/10.1145/2396761.2396863

[19] I. Becker and V. Aharonson, "Last but definitely not least: On the role of the last sentence in automatic polarity-classification," 2010.

[20] O. Täckström and R. McDonald, "Discovering fine-grained sentiment with latent variable structured prediction models," in *Advances in Information Retrieval*. Springer, 2011, pp. 368–374.

[21] J. McAuley and J. Leskovec, "Hidden factors and hidden topics: understanding rating dimensions with review text," in *Proceedings of the 7th ACM conference on Recommender systems*. ACM, 2013, pp. 165–172.

[22] Y. Lu and C. Zhai, "Opinion integration through semi-supervised topic modeling," in *Proceedings of the 17th International Conference on World Wide Web*, ser. WWW '08. New York, NY, USA: ACM, 2008, pp. 121–130. [Online]. Available: http://doi.acm.org/10.1145/1367497.1367514