Collective Opinion Target Extraction in Chinese Microblogs

Xinjie Zhou, Xiaojun Wan* and Jianguo Xiao
Institute of Computer Science and Technology
The MOE Key Laboratory of Computational Linguistics
Peking University
No. 5, Yiheyuan Road, Beijing, China
{zhouxinjie,wanxiaojun,xiaojianguo}@pku.edu.cn

Abstract

Microblog messages pose severe challenges for current sentiment analysis techniques due to some inherent characteristics such as the length limit and informal writing style. In this paper, we study the problem of extracting opinion targets of Chinese microblog messages. Such fine-grained word-level task has not been well investigated in microblogs yet. We propose an unsupervised label propagation algorithm to address the problem. The opinion targets of all messages in a topic are collectively extracted based on the assumption that similar messages may focus on similar opinion targets. Topics in microblogs are identified by hashtags or using clustering algorithms. Experimental results on Chinese microblogs show the effectiveness of our framework and algorithms.

1 Introduction

Microblogging services such as Twitter[1], Sina Weibo[2] and Tencent Weibo[3] have swept across the globe in recent years. Users of microblogs range from celebrities to ordinary people, who usually express their emotions or attitudes towards a broad range of topics. It is reported that there are more than 340 million tweets per day on Twitter and more than 200 million on Sina Weibo. A tweet means a post on Twitter. Since we mainly focus on Chinese microblogs instead of Twitter in this paper, we will refer to a post as a message. Each message is limited to 140 Chinese characters and usually contains several sentences.

Currently, researches on microblog sentiment analysis have been conducted on polarity classification (Barbosa and Feng, 2010; Jiang el al., 2011; Speriosu et al., 2011) and have been proved to be useful in many applications, such as opinion polling (Tang et al., 2012), election prediction (Tumasjan et al., 2010) and even stock market prediction (Bollen et al., 2011). However, classifying microblog texts at the sentence level is often insufficient for applications because it does not identify the opinion targets. In this paper, we will study the task of opinion target extraction for Chinese microblog messages.

Opinion target extraction aims to find the object to which the opinion is expressed. For example, in the sentence “The sound quality is good!”, “sound quality” is the opinion target. This task is mostly studied in customer review texts in which opinion targets are often referred as features or aspects (Liu, 2012). Most of the opinion target extraction approaches rely on dependency parsing (Zhuang et al., 2006; Jakob and Gurevych, 2010; Qiu et al., 2011) and are regarded as a domain-dependent task (Li et al., 2012a). However, such approaches are not suitable for microblogs because the natural language processing tools perform poorly on microblog texts due to their inherent characteristics. Studies show that one of the state-of-the-art part-of-speech taggers - OpenNLP only achieves the accuracy of 74% on tweets (Liu et al. 2011). The syntactic analysis tool that generates dependency relation may perform even worse. Besides, microblog messages may express opinion in different ways and do not always contain opinion words, which lowers the performance of methods utilizing opinion words to find opinion targets.

In this study, we propose an unsupervised method to collectively extract the opinion targets from opinionated sentences in the same topic.

* Xiaojun Wan is the corresponding author.

[1] https://twitter.com

[2] http://weibo.com/

[3] http://t.qq.com/

Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 1840–1850,
Seattle, Washington, USA, 18-21 October 2013. ©2013 Association for Computational Linguistics

Topics are directly identified by hashtags. We first present a dynamic programming based segmentation algorithm for Chinese hashtag segmentation. By leveraging the contents in a topic, our segmentation algorithm can successfully identify out-of-vocabulary words and achieve promising results. Afterwards, all the noun phrases in each sentence and the hashtag segments are extracted as opinion target candidates. We propose an unsupervised label propagation algorithm to collectively rank the candidates of all sentences based on the assumption that similar sentences in a topic may share the same opinion targets. Finally, for each sentence, the candidate which gets the highest score after unsupervised label propagation is selected as the opinion target.

Our contributions in this study are summarized as follows: 1) our method considers not only the explicit opinion targets within the sentence but also the implicit opinion targets in the hashtag or mentioned in the previous sentence. 2) We develop an efficient algorithm to segment Chinese hashtags. It can successfully identify out-of-vocabulary words by leveraging contextual information and help to improve the segmentation performance of the messages in the topic. 3) We develop an unsupervised label propagation algorithm for collective opinion target extraction. Label propagation (Zhu and Ghahramani, 2002) aims to spread label distributions from a small training set throughout the graph. However, our unsupervised algorithm leverages the connection between two adjacent unlabeled nodes to find the correct labels for both of them. The proposed unsupervised method does not need any training corpus which will cost much human labor especially for fine-grained annotation. 4) To the best of our knowledge, the task of opinion target extraction in microblogs has not been well studied yet. It is more challenging than microblog sentiment classification and opinion target extraction in review texts.

2 Characteristics of Chinese Microblogs

Most of previous microblog sentiment analysis researches focus on Twitter and especially in English. However, the analysis of Chinese microblogs has some differences with that of Twitter: 1) Chinese word segmentation is a necessary step for Chinese sentiment analysis, but the existing segmentation tool performs poorly on microblogs because the microblog texts are much different from regular texts. 2) Wang et al. (2011) find that hashtags in English tweets are used to highlight the sentiment information such as “ #love”, “#sucks” or serve as user-annotated coarse topics such as “#news”, “#sports”. But in Chinese microblogs, most of the hashtags are used to indicate fine-grained topics such as #NBA总决赛第七场# (#NBAFinalG7#). Besides, hashtags in Twitter always appear within a sentence such as “I love #BarackObama!” while hashtags in Chinese microblogs are always isolated and are surrounded by two # symbols such as “#巴拉克奥巴马# 我爱他!” (“#BarackObama# I love him！ ”).

It is noteworthy that topics aggregated by the same hashtag play an important role in Chinese microblog websites. These websites often provide an individual webpage[4] to list hot topics and invite people to participate in the discussion, where each topic consists of tens of thousands of messages with the same hashtag. The hot topics have a wide coverage of timely events and entities. Analyzing the opinion targets of these topics can help to get a deeper overview of the public attitudes towards the entities involved in the hot topics.

3 Motivation

As described above, #hashtags# in Chinese microblogs often indicate fine-grained topics. In this study, we aim to collectively extract the opinion targets of messages with the same hashtag, i.e. in the same topic. Opinion target of a sentence can be divided into two types, one of which called *explicit target* appears in the sentence such as “I love *Obama*”, and the other one called *implicit target*

Topic	Sentence
#官员财产公示# #Property publicity of government offic-ials#	1. 纯属作秀！ (Just for show！)
	2. 财产公示在中国就是作秀。 (Property publicity is just a show in China.)
#菲军舰恶意撞击# #Philippine navy vessel hits Chinese fishing boat#	1. 政府还是不够强硬。 (The government is not tough enough.)
	2. 政府为何不能强硬一些？ (Why cannot the government take a tougher line?)

Table 1. Motivation Examples

[4] http://huati.weibo.com/

may appear out of the sentence, for example, the sentence "Just for show!" in Table 1 directly comments on the target in the hashtag "#Property publicity of government officials#" . Such implicit opinion targets are not considered in previous works and are more difficult to extract than explicit targets. However, we believe that the contextual information will help to locate both of the two kinds of opinion targets because similar sentences in a topic may share the same opinion target, which provides the possibility for collective extraction.

Table 1 shows the motivation examples of two topics and four sentences. The two sentences in each topic are considered to be similar because they share several Chinese words. In the topic #官员财产公示# (#Property publicity of government officials#), the first sentence omits the opinion target. However, the second one contains an explicit target "财产公示" ("property publicity") in the sentence. If we find the correct opinion target for sentence 2, we can infer that sentence 1 may have an implicit opinion target similar to the opinion target in sentence 2. In the second topic, both sentences contain a noun word "政府" ("government"). The similarity between these two sentences may indicate that both of the two sentences are expressing opinion on "政府".

Based on the above observation, we can assume that similar sentences in a topic may have the same opinion targets. Such assumption can help to locate both explicit and implicit opinion targets. Following this idea, we firstly extract all the noun phrases in each sentence as opinion target candidates after applying Chinese word segmentation and part-of-speech tagging. Afterwards, an unsupervised label propagation algorithm is proposed to rank these candidates for all sentences in the topic.

In our methods, hashtags are used to find gold-standard topics. For messages without hashtags, an alternative way is to generate pseudo topics by clustering microblogs messages and then apply the proposed algorithm to each pseudo topic. The detailed discussion of such general circumstance is shown in Section 5.7.

4 Methodology

4.1 Context-Aware Hashtag Segmentation

In our approach, the Chinese word segmentations of hashtags and topic contents are treated separately. Existing Chinese word segmentation tools work poorly on microblog texts. The segmentation errors especially on opinion target words will directly influence the results of part-of-speech tagging and candidate extraction. However, some of the opinion target words in a topic are often included in the hashtag. By finding the correct segments of a hashtag and adding them to the user dictionary of the Chinese word segmentation tool, we can remarkably improve the overall segmentation performance.

The following example can help to understand the idea better. In the topic #90后打老人# (means "A young man hits an old man"), "90后" (literally "90 later" and means a young man born in the 90s) is an important word because it is the opinion target of many sentences. However, existing Chinese word segmentation tools will regard it as two separate words "90" and "后" ("later"). Then in the part-of-speech tagging stage, "90" will be tagged as number and "后" will be tagged as localizer. As we only extract noun phrases as opinion target candidates, the wrong segmentation on "90后" makes it impossible to find the right opinion target. Such error may occur many times in sentences that mention the word "90后" and express opinion on it. In our method, the message texts of the topic are utilized to identify such out-of-vocabulary words based on its frequency in the topic. For example, the high frequency of "90后" is a strong indication that it should be regard as a single word. After segmenting the hashtag correctly into "90后/打/老人", we can add the hashtag segments to the user dictionary of the segmentation tool to further segment the message texts of the topic.

The basic idea for our hashtag segmentation algorithm is to regard strings that appear frequently in a topic as words. Formally, given a hashtag *h* that contains *n* Chinese characters $c_1c_2...c_n$. We want to segment into several words $w_1w_2...w_m$, where each word is formed by one of more characters.

Firstly, we define the stickiness score for a Chinese string $c_1c_2...c_n$ based on the Symmetrical Conditional Probability (SCP) (Silva and Lopes, 1999):

$$SCP(c_1c_2...c_n) = \frac{\Pr(c_1c_2...c_n)^2}{\frac{1}{n-1}\sum_{i=1}^{n}\Pr(c_1...c_i)\Pr(c_{i+1}...c_n)} \quad (1)$$

and $SCP(c_1)$ = $\Pr(c_1)^2$ for string with only one character. $\Pr(c_1c_2...c_n)$ is the occurrence frequency of the string in the topic.

Following (Li et al., 2012b), we smooth the SCP value by taking logarithm calculation. Besides, the length of the string is taken into consideration,

$$SCP'(c_1c_2...c_n) = n \times \log SCP(c_1c_2...c_n) \quad (2)$$

where n is the number of characters in the string.

Then the stickiness score is defined by the sigmoid function as follows:

$$Stickiness(c_1c_2...c_n) = \frac{2}{1+e^{-SCP'(c_1c_2...c_n)}} \quad (3)$$

For the hashtag h = $c_1c_2...c_n$, we want to segment it into m words $w_1w_2...w_m$ which maximize the following equation,

$$\max \sum_{i=1}^{m} Stickness(w_i) \quad (4)$$

The optimization of Equation (4) can be solved efficiently by dynamic programming which iteratively segments a string into two substrings. Different from (Li et al., 2012b) which calculates the SCP value of each string based on Microsoft Web N-Gram, our hashtag segmentation algorithm only uses the topic content and do not need any additional corpus.

4.2 Candidate Extraction

After segmenting the hashtag, all the hashtag segments with length greater than one are added to the user dictionary of the Chinese word segmentation tool ICTCLAS[5] to further segment the message texts of the topic. It also assigns the part-of-speech tag for each word after segmentation. The noun phrases in each sentence is extracted by the following regular expression: $(noun \mid adj)(noun \mid adj \mid 的)*noun$. That means a noun phrase can only include nouns, adjectives and the Chinese word “的” (“of”). It should begin with a noun or adjective and end with a noun. For example, in the following sentence, “中国/n 的/u 教育/n 制度/n 有/v 问题/n 。/w” (“Chinese education system has problems.”), “中国的教育制度” (“Chinese education system”) and “问题” (“problem”) are extracted as noun phrases.

The character number of a noun phrase is limited between two and seven Chinese characters. For each sentence, all phrases that match the regular expression and meet the length restriction are extracted as explicit opinion target candidates. The hashtag segments are regarded as implicit candidates for all sentences. Besides, some opinionated sentences in microblogs do not contain any noun phase, such as “无聊至极！” (“*So boring*!”). These sentences may express opinion on object that has been mentioned before. Therefore, the explicit candidates of the previous sentence in the same message are also taken as the implicit candidates for such sentences.

We do not use any syntactic parsing tool to extract noun phrases because the parsing results on microblogs are not reliable. A performance comparison of our rule based method and the state-of-the-art syntactic parser will be shown in Section 5.

4.3 Unsupervised Label Propagation for Candidate Ranking

We simply assume that each opinionated sentence has one opinion target, which is consistent with

Algorithm 1 Unsupervised Label Propagation

```
Input:
  Graph:                   $G = <V, E, \tilde{W}>$
  Candidate Similarity:    $S \in R_+^{M \times M}$
  Prior labeling:          $Y_v \in R_+^{1 \times M}$ for $v \in V$
  Filtering Matrix:        $F_v \in R_+^{M \times M}$ for $v \in V$
  Probability:             $p^{inj}$ and $p^{cont}$
Output:
  Label vector:            $\hat{Y}_v \in R_+^{1 \times M}$
for all $v \in V$ do
    $\hat{Y}_v \leftarrow Y_v$
end for
repeat
    for all $v \in V$ do
        $D_v \leftarrow \sum_{u \in V, u \neq v} \tilde{W}_{uv} \left(\hat{Y}_u \times S\right) \times F_v$
        $\hat{Y}_v \leftarrow p^{inj} Y_v + p^{cont} D_v$
    end for
until convergence
```

[5] http://www.ictclas.org/

the statistical result of our dataset that over 93% sentences have only one opinion target and each sentence has an average of 1.09 targets. Therefore, the most confident candidate of each sentence will be selected as the opinion target. In this section, we introduce an unsupervised graph-based label propagation algorithm to collectively rank the candidates of all sentences in a topic.

Label propagation (Zhu and Ghahramani, 2002; Talukdar and Crammer, 2009) is a semi-supervised algorithm which spreads label distributions from a small set of nodes seeded with some initial label information throughout the graph. The basic idea is to use information from the labeled nodes to label the adjacent nodes in the graph. However, our idea is to use the connection between different nodes to find the correct labels for all of them. Our unsupervised label propagation algorithm is summarized in Algorithm 1. Sentences are regarded as nodes and candidates of each sentence are regarded as labels. The label vector for each node is initialized based on the results of the candidate extraction step, which means no manually-labeled instances are needed in our model. In each iteration, the label vector of one node is propagated to the adjacent nodes. Both the sentence (node) similarity and the candidate (label) similarity are considered during propagation. Finally, we select the candidate with the highest score in the label vector as the opinion target for each sentence. The details of Algorithm 1 are presented as follows.

Formally, an undirected graph $G = < V, E, \tilde{W} >$ is built for each topic. A node $v \in V$ represents a sentence in the topic and an edge $e = (a, b) \in E$ indicates that the labels of the two vertices should be similar. $\tilde{W}$ is the normalized weight matrix to reflect the strength of this similarity. The similarity between two nodes W_{ab} is simply calculated by using the cosine measure (Salton et al., 1975) of the two sentences.

$$W_{ab} = cos(T_a, T_b) = \frac{T_a \cdot T_b}{\|T_a\| \cdot \|T_b\|} \tag{5}$$

where T_a and T_b are the term vectors of sentences a and b represented by the standard vector space model and weighted by term frequency. After calculating the similarity matrix W, we get the weight matrix $\tilde{W}$ by normalizing each row of W such that $\sum_b \tilde{W}_{ab} = 1$.

For each sentence (node) v, a candidate set C_v is extracted in the previous step. The candidate set CT for the whole topic is the union of all C_v,

$$CT = \bigcup C_v \tag{6}$$

The total number of candidates in the topic is denoted by $M = |CT|$. We calculate the candidate similarity matrix $S \in R_+^{M \times M}$ based on Jaccard Index:

$$S_{ij} = \frac{|A(CT_i) \cap A(CT_j)|}{|A(CT_i) \cup A(CT_j)|} \quad 1 \le i \ne j \le M \tag{7}$$

where $A(CT_i)$ and $A(CT_j)$ are the Chinese character sets of the i-th and j-th candidates in CT respectively.

Candidates are regarded as labels in our model and without loss of generality we assume that the possible labels for the whole topic are $L = \{1 \ldots M\}$ and each label in L corresponds to a unique candidate in CT. For each node $v \in V$, a label vector $Y_v \in R_+^{1 \times M}$ is initialized as

$$\left(Y_v\right)_k = \begin{cases} w & L_k \in C_v \\ 0 & L_k \notin C_v \end{cases} \quad 1 \le k \le M \tag{8}$$

where w is the initial weight of the candidate. We set $w = w_e$ if L_k is an explicit candidate (extracted noun phrase) of v and $w = w_i$ if L_k is an implicit candidate (hashtag segment or inherited from previous sentence) of v. If L_k is not a candidate of the current sentence, then the corresponding value in the label vector is 0. These values which are initialized as zero should always remain zero during the propagation algorithm because the corresponding label does not belong to the candidate set C_v of node v. To reset the values on these positions, a diagonal matrix $F_v \in R_+^{M \times M}$ is created for all nodes v,

$$\left(F_v\right)_{kk} = \begin{cases} 1 & \left(Y_v\right)_k > 0 \\ 0 & \left(Y_v\right)_k = 0 \end{cases} \quad 1 \le k \le M \tag{9}$$

where the subscript kk denotes the k-th position in the diagonal of matrix F_v. We can right-multiply Y_v by F_v to clear the values of the invalid candidates. Figure 1 shows an example of creating the filtering matrix for a label vector.

$$Y_v = \begin{bmatrix}1 & 1 & 0.5 & 0\end{bmatrix} \rightarrow F_v = \begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 0\end{pmatrix}$$

Figure 1. Example of filtering matrix

The propagation process is formalized via two possible actions: inject and continue, with predefined probabilities p^{inj} and p^{cont}. Their sum is unit: $p^{inj} + p^{cont} = 1$. In each iteration, every node is influenced by its adjacent nodes. The propagation influence for each node v is

$$D_v = \sum_{u\in V, u\neq v} \tilde{W}_{uv}\left(\hat{Y}_u \times S\right)\times F_v \qquad (10)$$

where $\hat{Y}_u$ is the label vector of node u at the previous iteration. By multiplying the candidate similarity matrix S, we aim to propagate the score of the i-th candidate of node u not only to the i-th candidate of node v, but also to all the other candidates. W_{uv} measures the strength of such propagation. The filtering matrix F_v is used to clear the values of the invalid candidates as described above.

Then the label vector of node v is updated as follow,

$$\hat{Y}_v = p^{inj} Y_v + p^{cont} D_v \qquad (11)$$

When the positions of the largest values in all label vectors keep unchanged in ten iterations, it is regarded that the algorithm has already converged.

5 Experiments

5.1 Dataset

We use the dataset from the 2012 Chinese Microblog Sentiment Analysis Evaluation (CMSAE)[6] held by China Computer Federation (CCF). There are three tasks in the evaluation: subjectivity classification, polarity classification and opinion target extraction. The dataset contains 20 topics collected from Tencent Weibo, a popular Chinese microblogging website. All the messages in a topic contain the same hashtag. The dataset has a total of 17518 messages and 31675 sentences. In each topic, 100 messages are manually annotated with subjectivity, polarity and opinion targets. A total of 2361opinion targets are annotated for 2152 opinionated sentences.

5.2 Evaluation Metric

Precision, recall and F-measure are used in the evaluation. Since expression boundaries are hard to define exactly in annotation guidelines (Wiebe et al., 2005), both the strict evaluation metric and the soft evaluation metric are used in CMSAE.

Strict Evaluation: For a proposed opinion target, it is regarded as correct only if it covers the same span with the annotation result. Note that, in CMSAE, an opinion target should be proposed along with its polarity. The correctness of the polarity is also necessary.

Soft Evaluation: The soft evaluation metric presented in (Johansson and Moschitti, 2010) is adopted by CMSAE. The *span coverage c* between each pair of the proposed target span *s* and the gold standard span *s'* is calculated as follows,

$$c(s, s') = \frac{|s \cap s'|}{|s'|} \qquad (12)$$

In Equation 12, the operator | | counts Chinese characters, and the intersection ∩ gives the set of characters that two spans have in common.

Using the span coverage, the span set coverage *C* of a set of spans *S* with respect to another set *S'* is

$$C(S, S') = \sum_{s\in S}\sum_{s'\in S'} c(s, s') \qquad (13)$$

The soft precision *P* and recall *R* of a proposed set of spans $\hat{S}$ with respect to a gold standard set *S* is defined as follows:

$$\text{Precision} = \frac{C(\hat{S}, S)}{|\hat{S}|} \quad \text{Recall} = \frac{C(\hat{S}, S)}{|S|} \qquad (14)$$

Note that the operator | | counts spans in Equation 14. The soft F-measure is the harmonic mean of soft precision and recall.

5.3 Comparison Methods

Our proposed approach is first compared with the CMSAE teams.

CMSAE Teams: Sixteen teams participated in the opinion target extraction task of CMSAE. The methods of the top 3 teams are used as baselines

[6] http://tcci.ccf.org.cn/conference/2012/pages/page04_eva.html. The dataset can also be publicly accessed on the website.

here. They are denoted as **Team-1**, **Team-2** and **Team-3** respectively. The average result of all the sixteen teams is also included and is denoted as **Team-Avg**. We will briefly introduce the best team's method. The most important component of their model is a topic-dependent opinion target lexicon which is called object sheet. If a word or phrase in the object sheet appears in a sentence or a hashtag, it is extracted as opinion target. The object sheet is manually built for each topic, which means their method cannot be applied to new topics.

The following models are also used for comparison.

AssocMi: We implement the unsupervised method for opinion target extraction based on (Hu and Liu, 2004), which relies on association mining and a sentiment lexicon to extract frequent and infrequent product features.

CRF: The CRF-based method used in (Jakob and Gurevych, 2010) is also used for comparison. We implement both the single-domain and cross-domain models. Both models are evaluated using 5-fold cross-validation. More specifically, the single-domain model, denoted as **CRF-S**, trains different models for different topics. In each cross-validation round, 80 percent of each topic is used for training and the other 20 percent is used for test. The cross-domain model, denoted as **CRF-C**, uses 16 topics for training and the rest 4 topics for test in each round.

5.4 Comparison Results

CMSAE requires all the teams to perform the subjectivity and polarity classification task in advance. The opinion targets are extracted only for opinionated sentences and should be proposed along with their polarity. To make a fair comparison, we directly use the subjectivity and polarity classification results of Team-1. Then our unsupervised label propagation (**ULP**) method is used to extract the opinion targets for the proposed opinionated sentences. The parameters of our method are simply set as $p^{inj} = p^{cont} = 0.5$, $w_e = 1$ and $w_i = 0.5$.

Table 2 lists the comparison results with CMSAE teams. The average F-measure of all teams is 0.12 and 0.20 in strict and soft evaluation, respectively. It shows that opinion target extraction is a quite hard problem in Chinese microblogs. Our method performs better than all the teams. It increases by 10% and 13% in the two kinds of F-measure compared to the best team. Besides, we do not need any prior information of the topics while Team-1 has to manually build an opinion target lexicon for each topic.

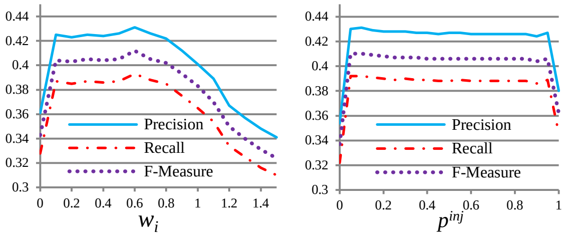
To compare with the other opinion target extraction methods, we only use gold-standard opinionated sentences for evaluation and do not classify the polarity of the opinion targets. Table 3 shows the experimental results of the four models. Our approach achieves the best result among them. AssocMi performs worst in strict evaluation but gets better results than the two CRF-based models in soft evaluation. The two CRF-based models achieve high precision but low recall. We can also observe that CRF-S is much more effective than CRF-C. It achieves high results because it has already seen the opinion targets in the training set. However, it is impossible to build such single-domain model in practical applications because

Method.	Strict			Soft		
	Precision	Recall	F-measure	Precision	Recall	F-measure
Team-Avg	0.17	0.09	0.12	0.29	0.15	0.20
Team-3	0.26	0.16	0.20	0.40	0.25	0.31
Team-2	0.31	0.18	0.23	0.40	0.22	0.29
Team-1	0.30	**0.27**	0.29	0.39	0.36	0.37
ULP	**0.37**	**0.27**	**0.32**	**0.48**	**0.37**	**0.42**

Table 2. Comparison results with CMSAE teams (with subjectivity and polarity classification in advance)

Method	Strict			Soft		
	Precision	Recall	F-Measure	Precision	Recall	F-Measure
AssocMi	0.22	0.20	0.21	0.47	0.43	0.45
CRF-C	0.59	0.15	0.24	0.70	0.18	0.28
CRF-S	**0.61**	0.27	0.35	**0.73**	0.31	0.41
ULP	0.43	**0.39**	**0.41**	0.61	**0.55**	**0.58**

Table 3. Comparison results with baseline methods (only gold-standard opinionated sentences are used)



(a) Initial Candidate Weight (b) Injection Probability

Figure 2. Influence of the parameters

labeled instances are not available for new topics. Our proposed method does not require any training data and gets an increase of 17% over CRF-S and 70% over CRF-C in strict evaluation. In terms of soft evaluation, we achieve an increase of 41% and 107% over the two CRF models.

5.5 Parameter Sensitivity Study

In this section, we study the parameter sensitivity. There are two major parameters in our algorithm: the initial weight w for both explicit and implicit candidates in Equation 8 and the injection probability p^{inj} in Equation 11.

The initial weights of explicit and implicit candidates are set differently because the explicit candidates are more likely to be the opinion targets. These two kinds of initial weights are denoted as w_e and w_i for explicit and implicit candidate, respectively. To study the impact of the initial weights, we fix w_e at 1 and tune w_i because we only care about the relative contribution of them. The injection probability is fixed at 0.5. Figure 2(a) displays the opinion target extraction performance when w_i varies from 0 to 1.5. Due to limited space, we only list the strict F-measure of opinion target extraction evaluated on opinioned sentences (same experimental setup as Table 3).

In particular, when w_i is equal to 0, only explicit candidates are considered. When w_i becomes larger than 1, the implicit candidates become more important than explicit candidates. From the curve in Figure 2(a), we can observe that the implicit candidates help to improve the performance significantly when w_i varies from 0 to 0.1. The performance reaches the peak when w_i = 0.7 and declines rapidly when w_i gets larger than 1.

To study the impact of injection probability p^{inj}, we fix the initial weights for explicit and implicit candidates as 1 and 0.5, respectively. Figure 2(b) shows the results of opinion target extraction with respect to different values of the injection probability. We can observe that the performance keeps steady except for the two extreme values 0 and 1. From the above two figures, we can conclude that our proposed method performs well and robustly with a wide range of parameter values.

Method	Total	Correct	F-Measure of Opinion Target Extraction	
			Strict	Soft
Berkley Parser	4554	877	0.36	0.56
Rule	4105	918	0.37	0.56
HS + Rule	4094	1042	0.41	0.58

Table 4. Performance of candidate extraction and opinion target extraction

5.6 Analysis of Candidate Extraction

Candidate extraction is an important step in our proposed method. If the correct opinion target is not extracted as a candidate, the ranking step will be in vain. As described in Section 3, we develop a hashtag segmentation algorithm and use a rule based method to extract noun phrases from each sentence. We do not use any parsing tool because we believe the performance of these tools is not good enough when applied on microblogs. A quantitative comparison is shown in this section.

We use one of the state-of-the-art syntactic analysis tools **- Berkeley Parser** (Petrov et al., 2006) for comparison here. Noun phrases are directly extracted from the parsing results. Our method **HS+Rule** leverages the hashtag segments to enhance the segmentation result and extracts explicit candidate using a regular expression. To demonstrate the effectiveness of our hashtag segmentation algorithm, the second comparison baseline **Rule** directly uses ICTCLAS to segment the whole topic content and labels each word with its part-of-speech tag. The explicit candidates are extracted by using the same regular expression.

The performance on candidate extraction is compared in Table 4. The second column shows the number of all extracted candidates for all the opinionated sentences by different methods. The third column shows the number of correct opinion targets among them. We can find that the two rule-based models both outperform Berkeley Parser and our **HS+Rule** method finds 14% more correct opinion targets than **Rule**. It proves the effectiveness of our hashtag segmentation algorithm. The total number of candidates extracted by **HS+Rule** is also less than the other two methods. Therefore, the performance of label propagation will be improved when there are fewer candidates to rank. It can be demonstrated by the F-measure of opinion target extraction in the fourth and fifth columns. The experiments are conducted on opinionated sentence only as above. By using **HS+Rule** to extract candidates, our label propagation algorithm gets the highest F-measure in both evaluation metrics.

5.7 Performance on Pseudo Topics by Message Clustering

In our collective extraction algorithm, topics are directly identified by hashtags. For messages without hashtags, we can first employ clustering algorithms to obtain pseudo topics (clusters) and then exploiting the topic-oriented algorithm for collective opinion target extraction. To test the performance of the proposed method in such circumstance, we use the popular clustering algorithm - Affinity Propagation (Frey and Dueck, 2007) to generate topics. The experimental results are shown in Table 5. **APCluster** means that the messages are clustered after removing all the hashtags. **APCluster+HS** means that all the hashtags are retained as normal texts for calculating message similarity. Therefore, the clustering performance can be largely improved. The standard cosine similarity is used to measure the distance between microblog messages for Affinity Propagation in the above two methods. The last method denoted as **GoldCluster** directly uses hashtags to identify the gold-standard topics which shows the upper bound of the performance. After clustering microblogs, the opinion targets of messages in each cluster are collectively extracted by the proposed unsupervised label propagation algorithm. The experiments are conducted on opinionated sentences only.

From the results, we can see that clustering microblogs without hashtags is a quite difficult job which only gets an F-Measure of 0.27. However, the corresponding opinion target extraction performance is still promising, which outperforms the AssocMi and CRF-C methods in Table 3. With the help of hashtags, the clustering performance of **APCluster+HS** is largely improved and the opinion target extraction performance is also increased. It outperforms all the baseline methods in Table 3. The above results reveal that our proposed unsupervised label propagation algorithm works well in pseudo topics and the performance can be increased with better clustering results. Therefore, we can try to incorporate other social network information to improve the message clustering performance, which will be studied in our future work.

Clustering Method	F-Measure of Clustering	F-Measure of Opinion Target Extraction	
		Strict	Soft
APCluster	0.27	0.35	0.50
APCluster+HS	0.71	0.37	0.55
GoldCluster	1.00	0.41	0.58

Table 5. Performance of clustering and opinion target extraction

6 Related Work

Sentiment analysis, a.k.a. opinion mining, is the field of studying and analyzing people's opinions, sentiments, evaluations, appraisals, attitudes, and emotions (Liu, 2012). Most of the previous sentiment analysis researches focus on customer reviews (Pang et al., 2002; Hu and Liu, 2004) and some of them focus on news (Kim and Hovy, 2006) and blogs (Draya et al., 2009). However, sentiment analysis on microblogs has recently attracted much attention and has been proved to be very useful in many applications.

Classification of opinion polarity is the most common task studied in microblogs. Go et.al (2009) follow the supervised machine learning approach of Pang et al. (2002) to classify the polarity of each tweet by distant supervision. The training dataset of their method is not manually labeled but automatically collected using the emoticons. Barbosa and Feng (2010) use the similar pseudo training data collected from three online websites which provide Twitter sentiment analysis services. Speriosu et al. (2009) explore the possibility of exploiting the Twitter follower graph to improve polarity classification.

Opinion target extraction is a fine-grained word-level task of sentiment analysis. Currently, this task has not been well studied in microblogs yet. It is mostly performed on product reviews where opinion targets are always described as product features or aspects. The pioneering research on this task is conducted by Hu and Liu

(2004) who propose a method which extracts frequent nouns and noun phrases as the opinion targets. Jakob and Gurevych (2010) model the problem as a sequence labeling task based on Conditional Random Fields (CRF). Qiu et al. (2011) propose a double propagation method to extract opinion word and opinion target simultaneously. Liu et al. (2012) use the word translation model in a monolingual scenario to mine the associations between opinion targets and opinion words.

7 Conclusion and Future Work

In this paper, we study the problem of opinion target extraction in Chinese microblogs which has not been well investigated yet. We propose an unsupervised label propagation algorithm to collectively rank the opinion target candidates of all sentences in a topic. We also propose a dynamic programming based algorithm for segmenting Chinese hashtags. Experimental results show the effectiveness of our method.

In future work, we will try to collect and annotate data for microblogs in other languages to test the robustness of our method. The repost and reply messages can also be integrated into our graph model to help improve the results.

Acknowledgments

The work was supported by NSFC (61170166), Beijing Nova Program (2008B03) and National High-Tech R&D Program (2012AA011101).

References

Barbosa Luciano and Junlan Feng. 2010. Robust sentiment detection on twitter from biased and noisy data. In Proceedings of the 23rd International Conference on Computational Linguistics: Posters. Association for Computational Linguistics, 2010.

Johan Bollen, Huina Mao and Xiaojun Zeng. 2011. Twitter mood predicts the stock market. Journal of Computational Science 2.1 (2011): 1-8.

Gérard Dray, Michel Plantié, Ali Harb, Pascal Poncelet, Mathieu Roche and François Trousset. 2009. Opinion Mining from Blogs. In International Journal of Computer Informa-tion Systems and Industrial Management Applications.

Brendan J. Frey and Delbert Dueck. 2007. "Clustering by passing messages between data points." Science 315.5814 (2007): 972-976.

Alec Go, Richa Bhayani and Lei Huang. 2009. Twitter sentiment classification using distant supervision. CS224N Project Report, Stanford (2009): 1-12.

Minqing Hu and Bing Liu. Mining and summarizing customer reviews. 2004. In Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 168-177. ACM.

Long Jiang , Mo Yu, Ming Zhou, Xiaohua Liu and Tiejun Zhao. 2011. Target-dependent twitter sentiment classification. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, vol. 1, pp. 151-160.

Niklas Jakob and Iryna Gurevych. Extracting opinion targets in a single-and cross-domain setting with conditional random fields. 2010. In Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics.

Richard Johansson and Alessandro Moschitti. 2010. Syntactic and semantic structure for opinion expression detection. Proceedings of the Fourteenth Conference on Computational Natural Language Learning. Association for Computational Linguistics.

Soo-Min Kim and Eduard Hovy. 2006. Extracting Opinions, Opinion Holders and Topics Expressed in Online News Media Text. In Proceedings of the ACL Workshop on Sentiment and Subjectivity in Text, 2006, pp. 1–8.

Fangtao Li, Sinno Jialin Pan, Ou Jin, Qiang Yang and Xiaoyan Zhu. 2012a. Cross-Domain Co-Extraction of Sentiment and Topic Lexicons. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, pages 410–419, Jeju, Republic of Korea, 8-14 July 2012.

Chenliang Li, Jianshu Weng, Qi He, Yuxia Yao, Anwitaman Datta, Aixin Sun and Bu-Sung Lee. 2012b. Twiner: Named entity recognition in targeted twitter stream. In Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval, pp. 721-730. ACM.

Xiaohua Liu, Kuan Li, Ming Zhou and Zhongyang Xiong. 2011. Collective semantic role labeling for tweets with clustering. In Proceedings of the Twenty-Second international joint conference on Artificial

Intelligence-Volume Volume Three, pp. 1832-1837. AAAI Press.

Bing Liu. 2012. Sentiment analysis and opinion mining. Synthesis Lectures on Human Language Technologies 5.1 (2012): 1-167.

Kang Liu, Liheng Xu and Jun Zhao. 2012. Opinion Target Extraction Using Word-Based Translation Model. In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning.

Bo Pang, Lillian Lee and Shivakumar Vaithyanathan. 2002. 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.

Slav Petrov, Leon Barrett, Romain Thibaux and Dan Klein. Learning accurate, compact, and interpretable tree annotation. 2006. In Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics, pp. 433-440.

Guang Qiu, Bing Liu, Jiajun Bu and Chun Chen. 2011. Opinion word expansion and target extraction through double propagation. Computational linguistics 37, no. 1 (2011): 9-27.

G. Salton, A. Wong and C. S. Yang. 1975. A Vector Space Model for Automatic Indexing, Communications of the ACM, vol. 18, nr. 11, pages 613–620.

J. F. da Silva and G. P. Lopes. 1999. A local maxima method and a fair dispersion normalization for extracting multi-word units from corpora. In Proc. of the 6th Meeting on Mathematics of Language .

Michael Speriosu, Nikita Sudan, Sid Upadhyay and Jason Baldridge. 2011. Twitter polarity classification with label propagation over lexical links and the follower graph. In Proceedings of the First Workshop on Unsupervised Learning in NLP, pp. 53-63. Association for Computational Linguistics, 2011.

Partha Talukdar and Koby Crammer. New regularized algorithms for transductive learning. 2009. Machine Learning and Knowledge Discovery in Databases (2009): 442-457.

Jie Tang, Yuan Zhang, Jimeng Sun, Jinhai Rao, Wenjing Yu, Yiran Chen and A. C. M. Fong. 2012. Quantitative study of individual emotional states in social networks. Affective Computing, IEEE Transactions on 3, no. 2 (2012): 132-144.

Andranik Tumasjan, Timm O. Sprenger, Philipp G. Sandner and Isabell M. Welpe. 2010. Predicting elections with twitter: What 140 characters reveal about political sentiment. In Proceedings of the fourth international aaai conference on weblogs and social media, pp. 178-185.

X. Zhu and Z. Ghahramani. 2002. Learning from labeled and unlabeled data with label propagation. Technical report, CMU CALD tech report.

Li Zhuang, Feng Jing and Xiaoyan Zhu. 2006. Movie review mining and summarization. In Proceedings of the ACM 15th Conference on Information and Knowledge Management, pages 43–50, Arlington, Virginia, USA, November.