Automated Rule Selection for Aspect Extraction in Opinion Mining

Qian Liu[1,2], **Zhiqiang Gao**[1,2], **Bing Liu**[3] and **Yuanlin Zhang**[4]
[1]School of Computer Science and Engineering, Southeast University, China
[2]Key Laboratory of Computer Network and Information Integration
(Southeast University) Ministry of Education, China
[3]Department of Computer Science, University of Illinois at Chicago, USA
[4]Department of Computer Science, Texas Tech University, USA
[1,2]{qianliu, zqgao}@seu.edu.cn, [3]liub@cs.uic.edu, [4]y.zhang@ttu.edu

Abstract

Aspect extraction aims to extract fine-grained opinion targets from opinion texts. Recent work has shown that the syntactical approach, which employs rules about grammar dependency relations between opinion words and aspects, performs quite well. This approach is highly desirable in practice because it is unsupervised and domain independent. However, the rules need to be carefully selected and tuned manually so as not to produce too many errors. Although it is easy to evaluate the accuracy of each rule automatically, it is not easy to select a set of rules that produces the best overall result due to the overlapping coverage of the rules. In this paper, we propose a novel method to select an effective set of rules. To our knowledge, this is the first work that selects rules automatically. Our experiment results show that the proposed method can select a subset of a given rule set to achieve significantly better results than the full rule set and the existing state-of-the-art CRF-based supervised method.

1 Introduction

Aspect extraction is a fundamental task of opinion mining or sentiment analysis. It aims to extract fine-grained opinion targets from opinion texts. For example, in the sentence "This phone has a good screen," we want to extract "screen." In product reviews, an aspect is basically an attribute or feature of a product. Aspect extraction is important for opinion mining because without knowing the aspects that the opinions are about, the opinions are of limited use [Liu, 2012].

In recent years, aspect extraction has been studied extensively. There are two main approaches: *syntactical* and *statistical*. Some existing work has shown that syntactical dependency based methods such as *double propagation* (DP) [Qiu *et al.*, 2011] can perform better than the statistical learning based method Conditional Random Fields (CRF) [Lafferty *et al.*, 2001]. The key idea of the syntactical approach is that opinions have targets and there are often explicit syntactic relations between opinion words (e.g., "good") and target aspects (e.g., "screen"). By exploiting such relations, the DP method can use a set of seed opinion words to extract aspects and new opinion words, and then use them to extract more aspects and opinion words, and so on, through a propagation process.

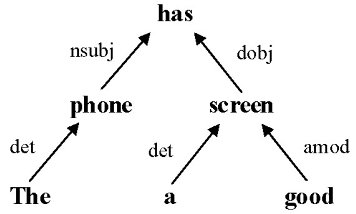

Figure 1: Dependency tree of "The phone has a good screen."

Figure 1 shows the dependency relations between words in the sentence "The phone has a good screen." The word "good" is an opinion word, which is known (given or extracted). The word "screen," a noun modified by "good" (i.e., they have a dependency relation *amod*), is clearly an aspect (or opinion target). Therefore, from a given set of opinion words, we can extract a set of aspects using some syntactic relations, e.g., *amod*. Similarly, one can also use syntactic relations to extract new aspects and new opinion words from the extracted aspects. For example, using the extracted aspect "screen," we can extract "speaker" from "Both the screen and the speaker are what I expected" using the conjunction "and." Note that, an aspect term can be a single word or a multi-word phrase (e.g., "battery life"). We will explain how phrases are handled in the experiment section.

Clearly, syntactic patterns can extract wrong terms. For example, in "It is a good idea to get this phone," the words "good" and "idea" also have the dependency relation *amod*, but "idea" is not an aspect of the phone. In order to produce good results, one has to carefully choose rules. Apart from choosing the right rules, some heuristics were also proposed in [Popescu and Etzioni, 2005; Qiu *et al.*, 2011] to prune the extracted aspects to reduce errors. However, simple heuristics such as frequency-based pruning method in [Hu and Liu, 2004] have the problem of pruning many low frequency correct aspects, and complicate heuristics such as the methods in [Qiu *et al.*, 2011] need external knowledge in pruning.

As we will see later, a large number of rules with different qualities can be used for extraction. In the existing works, researchers typically manually examine a set of rules, e.g., [Qiu *et al.*, 2011], and then choose a small subset of more reliable rules. However, since they work by trial and error, no reasons are provided on why some rules are chosen while others are not, how to decide whether a rule is good, and how to combine a set of rules. It is important to have the reasons in order to select rules automatically as the manual approach is very difficult to guarantee quality. This paper addresses this difficulty and proposes an automated rule selection algorithm to choose a subset of good rules from a given set of rules.

Formally, our problem is stated as follows. Given a set of aspect extraction rules $\mathcal{R}$ (each has an id), a set of seed opinion words $\mathcal{O}$, and a set of reviews $\mathcal{D}$ with labeled aspects, we want to select a subset of rules in $\mathcal{R}$ that can be used to extract aspects from reviews across domains. This cross domain capability is important because we do not want to label data in each domain, which is highly labor-intensive. The selected set of rules are obviously domain independent because they are syntactic rules, and thus can be used across domains.

The proposed algorithm can take any number of rules of any quality, and automatically select a good subset of rules for extraction. Selecting the optimal set of rules would involve evaluating all possible subsets of $\mathcal{R}$ on $\mathcal{D}$ and selecting the subset with the right rule sequence that performs the best. However, as the rule number m grows, the number of subsets grows exponentially, i.e., 2^m. Thus, we propose a greedy algorithm which is inspired by rule learning in machine learning and data mining [Liu *et al.*, 1998].

Our experiment is conducted using a popular aspect extract evaluation dataset that contains annotated reviews of five products [Hu and Liu, 2004] as well as some additional datasets annotated by us. Our experimental results show that the proposed method is able to select a subset of rules from those rules used in DP to perform extraction much more accurately than the original rule set used in DP. Furthermore, since our method can take rules of any quality as input, we add many more new rules to the rule set in DP. The proposed method selects a subset of rules that produces even better results than that selected from the DP rules only. It also outperforms the state-of-the-art CRF-based supervised learning method by a large margin.

2 Related Work

There are two main approaches for aspect extraction: *syntactical* and *statistical*. The former is mainly based on dependency relations [Hu and Liu, 2004; Zhuang *et al.*, 2006; Wang and Wang, 2008; Wu *et al.*, 2009; Zhang *et al.*, 2010; Zhao *et al.*, 2010b; 2014; Liu *et al.*, 2013b], while the latter on CRF [Jakob and Gurevych, 2010; Choi and Cardie, 2010; Mitchell *et al.*, 2013] and topic modeling [Mei *et al.*, 2007; Titov and McDonald, 2008; Li *et al.*, 2010; Brody and Elhadad, 2010; Moghaddam and Ester, 2011; Sauper *et al.*, 2011; Mukherjee and Liu, 2012; Chen *et al.*, 2014].

On the statistical approach, we will show that our method outperforms the CRF-based method [Jakob and Gurevych, 2010] by a large margin. Topic modeling often only gives some rough topics in a document corpus rather than precise aspects as a topical term does not necessarily mean an aspect [Lin and He, 2009; Lu *et al.*, 2009; Zhao *et al.*, 2010a; Jo and Oh, 2011; Fang and Huang, 2012]. For example, in a battery topic, a topic model may find topical terms such as "battery," "life," "day," and "time," etc, which are related to battery life, but each individual word is not an aspect.

There are also frequency-based methods for aspect extraction, which extract frequent noun phrases as aspects [Hu and Liu, 2004; Popescu and Etzioni, 2005; Ku *et al.*, 2006; Zhu *et al.*, 2009]. Additionally, researchers have also proposed different joint models leveraging the relations between opinion words and aspects as well as other information to extract aspects, opinion words, and their relations [Klinger and Cimiano, 2013; Lazaridou *et al.*, 2013]. Other similar works include label propagation [Zhou *et al.*, 2013] and word alignment based approaches [Liu *et al.*, 2013a].

In this paper, we focus on the syntactical approach. Our work is most related to the DP approach [Qiu *et al.*, 2011], which we will discuss further below. Since our rule selection method uses some training data, it can be regarded as an integration of both supervised and unsupervised learning because its resulting rule set can be applied to the test data from any domain. Our work is also related to associative classification [Liu *et al.*, 1998], which uses association rule mining algorithms to generate the complete set of association rules, and then selects a small set of high quality rules for classification, and the pattern mining method in [Kobayashi *et al.*, 2007]. However, we do not need to generate rules as we already have such rules based on syntactical dependency relations. Our rule selection method is also very different.

3 Syntactical Extraction Rules

This work uses dependency relations between opinion words and aspects, as well as between opinion words and aspects themselves to extract aspects. For example, one extraction rule could be "*if a word A, whose part-of-speech (POS) is a singular noun (nn), has the dependency relation amod with (i.e., modified by) an opinion word O, then A is an aspect*", which can be formulated by the following rule r_1:
IF $depends(amod, A, O) \wedge pos(A, nn) \wedge opinionword(O)$
THEN $aspect(A)$
where $depends(amod, A, O)$ means that A and O have a dependency relation *amod*, $pos(A, nn)$ means that A is a singular noun, $opinionword(O)$ means that O is an opinion word, $aspect(A)$ means that A is an aspect. For example, from "This phone has a great screen," we can extract the aspect "screen" as there is a *amod* relation between "great" and "screen," "great" is an opinion word, and "screen" is a noun.

As discussed in [Qiu *et al.*, 2011], there are many possible dependency relations that can be exploited for aspect extraction. To reduce incorrect aspects caused by propagation, [Qiu *et al.*, 2011] used only a small subset of manually selected rules based on 8 dependency relations. Since our proposed approach can automatically select a subset of good rules for aspect extraction, it can take as input any kind and any number of syntactical rules regardless of their qualities.

We group rules into three types based on whether they can extract aspects by themselves given a set of opinion words. For some rules, the propagation mechanism in DP is needed before they can be used in the extraction.

Type 1 rules (**$\mathcal{R}^1$**): using opinion words to extract aspects (based on some dependency relations between them), e.g., rule r_1. A set of seed opinion words are given a priori.

Type 2 rules (**$\mathcal{R}^2$**): using aspects to extract aspects. The known aspects are extracted in the previous propagation, e.g.,
IF $depends(conj, A_i, A_j) \wedge pos(A_i, nn) \wedge aspect(A_j)$
THEN $aspect(A_i)$
For example, if "screen" has been extracted by a previous rule as an aspect, this rule can extract "speaker" as an aspect from "This phone has a great screen and speaker" because "screen" and "speaker" has the $conj$ dependency relation.

Type 3 rules (**$\mathcal{R}^3$**): using aspects and opinion words to extract new opinion words. The given aspects are extracted in the previous propagation, and the given opinion words are the known seeds or extracted in the previous propagation. The following is an example of such rules:
IF $depends(amod, A, O) \wedge pos(O, jj) \wedge aspect(A)$
THEN $opinionword(O)$
where $pos(O, jj)$ means that O is an adjective. For example, if "screen" has been extracted as an aspect and "nice" was not a seed opinion word, then "nice" will be extracted as an opinion word by this rule from "This phone has a nice screen."

4 Rule Set Selection Algorithm

We are now ready to present the proposed algorithm. The first subsection gives the main idea and steps of the algorithm. The subsequent subsections detail each step.

4.1 The Main Idea and Steps

As mentioned in the introduction, finding the best subset of rules is an infeasible problem, we thus propose a greedy algorithm to perform the task, which has three steps:

Step 1: Rule evaluation. Clearly, rule quality is a key criterion for rule selection. This step first evaluates each rule to assess its quality. Specifically, given a set of rules, a set of seed opinion words, and a training data $\mathcal{D}$ with labeled aspects in its sentences, for each type of rules, this step applies each rule to $\mathcal{D}$ and outputs the precision and recall values of the rule. We use precision and recall because we want the rules with high precision and recall.

Step 2: Rule ranking. This step ranks the rules in each type first based on their precisions. If two rules have the same precision, the one with the higher recall is ranked higher. We use precision first because high precision rules are more desirable. The recall can be improved by using more rules. This step thus produces three rankings for the three types of rules to be used in the next step for rule selection.

Step 3: Rule selection. Given the ranked rules of each type and training data $\mathcal{D}$, this step adds rules from the ranked rule set one by one in the descending order into the current output rule subset. Once a rule is added, the current rule subset is applied to and evaluated on $\mathcal{D}$, and the F$_1$-score of the current rule subset is recorded. This process continues until all rules in the ranked list are added to the output rule set and evaluated. The algorithm then prunes the rules to produce the final set of rules that gives the best result on $\mathcal{D}$. In this step, F$_1$-score is used as the performance evaluation measure because we want the final rule set to produce overall good aspect extraction result. F$_1$-score is also the final evaluation measure in our experiments.

4.2 Rule Evaluation

Given the training data (e.g., product reviews) $\mathcal{D}$ with aspect labels, the evaluation of each rule in $\mathcal{R}^1$ is straight forward. We simply apply each rule $r_i^1 \in \mathcal{R}^1$ with seed opinion words $\mathcal{O}$ to $\mathcal{D}$ to extract all possible aspects, denoted by $\mathcal{A}_i^1$ ($i \in [1, N]$, where N is the number of rules of type 1). The set of correct aspects in $\mathcal{A}_i^1$ is denoted by $\mathcal{T}_i^1$, the precision of r_i^1 is defined by $r_i^1.pre = |\mathcal{T}_i^1|/|\mathcal{A}_i^1|$, and the recall of r_i^1 is defined by $r_i^1.rec = |\mathcal{T}_i^1|/|All_{lab}|$, where $|.|$ means the size of a set, All_{lab} is the set of all labeled aspects in $\mathcal{D}$. The set of all aspects extracted by rules of type 1 is denoted by $\mathcal{A}^1 = \bigcup_{i=1}^{N} \mathcal{A}_i^1$. Note that, the rules in $\mathcal{R}^1$ are independent of each other. Also, given some opinion words a priori, the evaluation of rules in $\mathcal{R}^1$ is independent to other types of rules.

Rules of type 2 are dependent on rules of type 1 in the sense that they need some known aspects as input in order to extract new aspects. We apply each rule $r_j^2 \in \mathcal{R}^2$ to $\mathcal{D}$ on the condition that $\mathcal{A}^1$ is given, to get new aspects extracted by r_j^2, denoted by $\mathcal{A}_j^2$ ($j \in [1, M]$, where M is the number of rules of type 2, $\mathcal{A}_j^2 \cap \mathcal{A}^1 = \emptyset$). The set of correct aspects in $\mathcal{A}_j^2$ is denoted by $\mathcal{T}_j^2$, the precision of r_j^2 is defined by $r_j^2.pre = |\mathcal{T}_j^2|/|\mathcal{A}_j^2|$, and the recall of r_j^2 is definded by $r_j^2.rec = |\mathcal{T}_j^2|/|All_{lab}|$. The set of all aspects extracted by rules of type 2 is denoted by $\mathcal{A}^2 = \bigcup_{j=1}^{M} \mathcal{A}_j^2$.

Rules of type 3 are dependent on rules of both type 1 and type 2 in the sense that rules of type 3 do not extract aspects directly and have to resort to rules of type 1 and type 2 to produce new aspects. Given $\mathcal{O}$, we apply each rule $r_k^3 \in \mathcal{R}^3$ to $\mathcal{D}$ together with all rules in $\mathcal{R}^1$ and $\mathcal{R}^2$, to get new aspects extracted by adding r_k^3 into $\mathcal{R}^1 \cup \mathcal{R}^2$, denoted by $\mathcal{A}_k^3$ ($k \in [1, L]$, where L is the number of rules of type 3, $\mathcal{A}_k^3 \cap (\mathcal{A}^1 \cup \mathcal{A}^2) = \emptyset$). The set of correct aspects in $\mathcal{A}_k^3$ is denoted by $\mathcal{T}_k^3$, the precision of r_k^3 is defined by $r_k^3.pre = |\mathcal{T}_k^3|/|\mathcal{A}_k^3|$, the recall of r_k^3 is defined by $r_k^3.rec = |\mathcal{T}_k^3|/|All_{lab}|$.

4.3 Rule Ranking

After all the rules are evaluated, they are ranked according to precision and recall. The three types of rules are ranked separately, i.e., rules of each type are ranked among themselves. The *order* of rules in each type is defined as follows: Given two rules r_i and r_j of the same type, $r_i, r_j \in \mathcal{R}^k (k = 1, 2, 3)$, we have $r_i \succ r_j$, called r_i is *higher* than r_j, if

1. $r_i.pre > r_j.pre$, or
2. $r_i.pre = r_j.pre$ and $r_i.rec > r_j.rec$, or
3. $r_i.pre = r_j.pre$, $r_i.rec = r_j.rec$ and $r_i.id < r_j.id$.

The order is a total order and defines a ranking on the rules.

4.4 Rule Selection

To select the best subset is impossible as we discussed in the introduction section. We thus propose a greedy selection algorithm based on the ranking of the rules of each type. The algorithm is called RS-DP (short for Rule Selection-DP) and given in Algorithm 1. RS-DP has four sub-steps:

Sub-step 1 (lines 1-2): Initialize the sequence $\mathcal{S} = <>$ (line 1), and discard the rules in the rule set whose precision and recall are both zero (line 2). It guarantees that the remaining rules can correctly extract at least one aspect.

Sub-step 2 (lines 3-7): Select rules from $\mathcal{R}^1$ following the descending order defined by the ranking. Since the rules in $\mathcal{R}^2$ and $\mathcal{R}^3$ depend on the rules in $\mathcal{R}^1$, the rules in $\mathcal{R}^1$ should be selected and added into $\mathcal{S}$ first. For each rule $r_i \in \mathcal{R}^1$, since r_i can correctly extract at least one aspect, it will be a potential rule in $\mathcal{S}$ (line 4). We apply $\mathcal{S}$ to $\mathcal{D}$ and $\mathcal{O}$ to find those cases covered by $\mathcal{S}$, i.e., they satisfy the rules in $\mathcal{S}$, and then compute the F$_1$-score of $\mathcal{S}$, called F$_1$-score *derived* from r_i, and denoted by $score(r_i) = \frac{2 \times \mathcal{S}.pre \times \mathcal{S}.rec}{\mathcal{S}.pre + \mathcal{S}.rec}$ (lines 5-6).

After inserting all the rules in $\mathcal{R}^1$ into $\mathcal{S}$, we perform pruning (line 7) by discarding those rules in $\mathcal{S}$ that do not improve the F$_1$-score of the rule set. Since when a rule r is added to the end of $\mathcal{S}$, $score(r)$, i.e., the F$_1$-score of the current $\mathcal{S}$ is recorded, we simply find the first rule in $\mathcal{S}$ from which the highest F$_1$-score is obtained. All the rules after this rule can be discarded because they only produce more errors.

Sub-step 3 (lines 8-12): Select rules from $\mathcal{R}^2$ following the descending order defined by the ranking. Given $\mathcal{O}$, rules in $\mathcal{R}^2$ cannot extract any new aspects directly, because they rely on known aspects. However, $\mathcal{S}$ now contains the selected rules from $\mathcal{R}^1$, which can extract aspects using the seed opinion words. The extracted aspects can be fed to the rules in $\mathcal{R}^2$. Thus, like that in sub-step 2, for each rule $r_j \in \mathcal{R}^2$, we first add it to the end of $\mathcal{S}$ (lines 8-9), and apply $\mathcal{S}$ to $\mathcal{D}$ and $\mathcal{O}$ to find those cases covered by $\mathcal{S}$, and then compute $score(r_j)$, i.e., F$_1$-score of $\mathcal{S}$ after adding r_j (lines 10-11).

After inserting all the rules in $\mathcal{R}^2$ into $\mathcal{S}$, line 12 performs rule pruning by discarding the rules that do not improve the F$_1$-score of $\mathcal{S}$. This sub-step works in the same way as line 7.

Sub-step 4 (lines 13-19): Select rules from $\mathcal{R}^3$ following the descending order defined by the ranking. Unlike sub-steps 2 and 3, not all the rules in $\mathcal{R}^3$ are potential rules in the final rule set, only those rules that improve the performance of $\mathcal{S}$ should be selected. If we use the same evaluation strategy in this sub-step as in sub-steps 2 and 3, it can result in many useless rules, which generate no additional extractions. Thus, we use a different evaluation strategy here. That is, for each rule $r_k \in \mathcal{R}^3$, we evaluate $\mathcal{S} \cup \{r_k\}$ on $\mathcal{D}$ and $\mathcal{O}$ (lines 13-14). If $\mathcal{S} \cup \{r_k\}$ performs better than $\mathcal{S}$, then r_k is inserted into $\mathcal{S}$ (lines 15-18). Since every added rule can improve the results in this sub-step, it has no pruning.

Algorithm 1 RS-DP

```
Input: Ranked rule sets $\mathcal{R}^1$, $\mathcal{R}^2$ and $\mathcal{R}^3$, training data $\mathcal{D}$,
    seed opinion words $\mathcal{O}$
Output: The best subset $\mathcal{S}$ of rules
 1: $\mathcal{S} = <>$;   // $\mathcal{S}$ is represented as a sequence
 2: Discard all the rules in $\mathcal{R}^1$, $\mathcal{R}^2$ and $\mathcal{R}^3$ whose precision
    and recall are both zero;
 3: for each rule $r_i \in \mathcal{R}^1$ in descending order do
 4:     insert $r_i$ at the end of $\mathcal{S}$;
 5:     compute F$_1$-score $x$ of $\mathcal{S}$ on $\mathcal{D}$ and $\mathcal{O}$, $score(r_i) = x$;
 6: end for
 7: Find the first rule $p$ in $\mathcal{S}$ with the highest derived F$_1$-score
    among all the rules of $\mathcal{S}$, drop all the rules after $p$ in $\mathcal{S}$;
 8: for each rule $r_j \in \mathcal{R}^2$ in descending order do
 9:     insert $r_j$ at the end of $\mathcal{S}$;
10:     compute F$_1$-score $x$ of $\mathcal{S}$ on $\mathcal{D}$ and $\mathcal{O}$, $score(r_j) = x$;
11: end for
12: Find the first rule $q$ in $\mathcal{S}$ with the highest derived F$_1$-score
    and drop all the rules after $q$ in $\mathcal{S}$, let $\max F = score(q)$;
13: for each rule $r_k \in \mathcal{R}^3$ in descending order do
14:     compute F$_1$-score $x$ of $\mathcal{S} \cup \{r_k\}$ on $\mathcal{D}$ and $\mathcal{O}$;
15:     if ($x > \max F$) then
16:         insert $r_k$ at the end of $\mathcal{S}$, $\max F = x$;
17:     end if
18: end for
19: Output $\mathcal{S}$ as the final rule set.
```

5 Experiments

We now evaluate the proposed technique to assess the performance of aspect extraction of the selected rules.

5.1 Datasets

We use two customer review collections in our experiments. One is from [Hu and Liu, 2004], which contains five review datasets of four domains: digital cameras (D1, D2), cell phone (D3), MP3 player (D4), and DVD player (D5). The other one is built by us in order to further verify the effectiveness of our approach in more domains. It contains three review datasets of three domains: computer (D6), wireless router (D7), and speaker (D8). Aspects in these review datasets are annotated manually. The first collection has been widely used in aspect extraction evaluation by researchers [Hu and Liu, 2004; Popescu and Etzioni, 2005; Qiu *et al.*, 2011; Liu *et al.*, 2013a]. The second one is annotated by two annotators. Since they did not identify the same number of aspects, instead of using Kappa statistics, we use Dice coefficient to measure the inter-annotator agreement for our annotations. The average Dice coefficient for three domains is 0.7, which indicates a reasonable high degree of inter-annotator agreement. The detailed information about these datasets is shown in Table 1. For seed opinion words, we used all (and only) the adjective opinion words in the opinion lexicon of [Hu and Liu, 2004][1].

Table 1: Detailed information of the datasets.

Data	Product	# of Sentences	# of Aspects
D1	Digital camera	597	237
D2	Digital camera	346	174
D3	Cell phone	546	302
D4	MP3 player	1716	674
D5	DVD player	740	296
D6	Computer	531	354
D7	Wireless router	879	307
D8	Speaker	689	440

5.2 Evaluation Metrics

Precision, recall, and F$_1$-score are employed as our evaluation metrics. There are two ways to compute the results: (1) based

[1] http://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html

on multiple occurrences of each aspect term, and (2) based on distinct occurrence of each aspect term.

In a dataset, an important aspect often occurs many times, e.g., the aspect "picture" occurred 10 times in a set of camera reviews. For (1), if any occurrence of "picture" is extracted, then all occurrences of "picture" are considered extracted, i.e., 10. If none of its occurrences is extracted, it is considered as 10 losses. In (2), if any occurrence of "picture" is extracted, it is considered as one extraction. If none is extracted, it is considered as one loss. (2) clearly makes sense, but (1) also makes good sense because it is crucial to get those important aspects extracted. Extracting (or missing) a more frequent aspect term is rewarded (or penalized) more heavily than extracting (or missing) a less frequent one.

Let an extraction method return a set $\mathcal{A}$ of distinct aspect terms, and the set of distinct aspect terms labeled by human annotators be $\mathcal{T}$. TP (true positives) is $|\mathcal{A} \cap \mathcal{T}|$, FP (false positives) is $|\mathcal{A} \setminus \mathcal{T}|$, FN (false negatives) is $|\mathcal{T} \setminus \mathcal{A}|$.

For (2), the evaluation metrics are defined as follows:

$Precision = \frac{TP}{TP+FP}$

$Recall = \frac{TP}{TP+FN}$

$F_1\text{-}score = \frac{2 \times Precision \times Recall}{Rrecision+Recall}$

For (1), F$_1$-score is computed in the same way, but precision and recall computations need to change because we now consider multiple occurrences of the same aspect:

$$Precision = \frac{\sum_{i=1}^{|\mathcal{A}|} f_i \times E(a_i,\mathcal{A})}{\sum_{i=1}^{|\mathcal{A}|} f_i}$$

$$Recall = \frac{\sum_{i=1}^{|\mathcal{T}|} f_i \times E(a_i,\mathcal{T})}{\sum_{i=1}^{|\mathcal{T}|} f_i}$$

where f_i is the term frequency of a_i, $E(a_i, \mathcal{A})$ (or $E(a_i, \mathcal{T})$) equals to 1 if a_i is an element of $\mathcal{A}$ (or $\mathcal{T}$), otherwise $E(a_i, \mathcal{A})$ (or $E(a_i, \mathcal{T})$) equals to 0.

5.3 Compared Approaches

In the experiments, we compare our approach with DP [Qiu *et al.*, 2011] and CRF [Jakob and Gurevych, 2010]. The reason to compare with DP is that we use DP rules as the input of our algorithm. We compare also with CRF as our approach is supervised. In total, we consider the following approaches, DP, DP$^+$, RS-DP, RS-DP$^+$, CRF and CRF$^+$.

DP denotes the original double propagation algorithm in [Qiu *et al.*, 2011]. It uses 8 aspect extraction patterns, which can be expanded into rules after instantiating the relation variables in each pattern with 8 dependency relations (*mod*, *pnmod*, *subj*, *s*, *obj*, *obj2*, *desc* and *conj*) defined in MiniPar[2]. Since Stanford Parser[3] is employed in our experiments, we use the corresponding dependency relations (*amod*, *prep*, *nsubj*, *csubj*, *xsubj*, *dobj*, *iobj* and *conj*) defined in Stanford Parser.

DP$^+$ still uses the 8 aspect extraction patterns as in DP. The difference is that DP$^+$ uses more dependency relations in the patterns. DP$^+$ uses 18 dependency relations, i.e., *amod*, *prep*, *nsubj*, *csubj*, *xsubj*, *dobj*, *iobj*, *conj*, *advmod*, *dep*, *cop*, *mark*, *nsubjpass*, *pobj*, *acomp*, *xcomp*, *csubjpass*, and *poss*.

RS-DP implements the proposed Algorithm 1 using all the rules of DP.

RS-DP$^+$ implements the proposed Algorithm 1 using all the rules of DP$^+$.

CRF is the supervised CRF-based aspect extraction method proposed in [Jakob and Gurevych, 2010].

CRF$^+$ is the same CRF-based method but with much more dependency features than CRF. The dependency relations between opinion words and other words used in CRF$^+$ are the same as those in DP$^+$ and RS-DP$^+$.

DP$^+$, RS-DP$^+$ and CRF$^+$ are designed to explore the effectiveness of adding more rules into the rule set.

Cross domain test. Unlike traditional supervised learning where when a model is learned from a domain, it is also tested in the same domain, we test across domains because syntactical extraction rules are meant to be domain independent. Otherwise, it defeats the purpose of using such rules. Since the datasets D1 to D5 are widely used, and D6 to D8 are newly built, we evaluate these approaches on D1 to D5, and D6 to D8 separately. In testing RS-DP, RS-DP$^+$, CRF and CRF$^+$, to reflect cross domain aspect extraction, we use leave-one-out cross validation for D1 to D5, i.e., the algorithm selects rules based on the annotated data from four products, and tests the selected rules using the unseen data from the remaining product; for D6 to D8, the algorithm selects rules based on the annotated data from D1 to D5, and tests the selected rules using each of the data from D6 to D8. This simulates the situation that the selected rules can be applied to any domain (or in a domain independent matter).

In our experiments, all the approaches use Stanford Parser for part-of-speech tagging and dependency parsing. Note that, we extract not only single noun aspects but also noun phrases (multi-word expressions) in all the six approaches. For the approaches DP, DP$^+$, RS-DP and RS-DP$^+$, noun phrases are identified based on the dependency relation *nn* (noun compound modifier) defined in Stanford Parser, then the identified noun phrases are treated as ordinary nouns in the rules. In CRF and CRF$^+$, noun phrases are identified by the CRF algorithm, which is the same as in [Jakob and Gurevych, 2010].

5.4 Experimental Results

Table 2 shows the results of DP, DP$^+$, RS-DP, RS-DP$^+$, CRF and CRF$^+$ tested on D1 to D8 and evaluated based on multiple occurrences of each aspect term (evaluation (1)). Table 3 shows the corresponding results of the approaches evaluated based on distinct aspect terms (evaluation (2)).

In our experiments, DP and DP$^+$ are directly tested on each dataset as the rules are domain or dataset independent. For RS-DP, RS-DP$^+$, CRF and CRF$^+$, as mentioned earlier, to test across domains D1 to D5, we use leave-one-out cross validation, and to test D6 to D8, we use the rules selected from D1 to D5.

From the tables, we observe that the F$_1$-scores of RS-DP and RS-DP$^+$ are markedly better than those of DP, DP$^+$, CRF and CRF$^+$. RS-DP$^+$ is also better than RS-DP because RS-DP$^+$ uses more rules although some of the additional rules were not of high quality, which can be seen from the average F$_1$-scores of DP and DP$^+$.

Specifically, as shown in Table 2 and Table 3, the average precision of DP is higher than that of DP$^+$, and the average

[2] http://webdocs.cs.ualberta.ca/~lindek/minipar.htm

[3] http://nlp.stanford.edu:8080/parser/

Table 2: Precision, Recall and F$_1$-score of DP, DP$^+$, RS-DP, RS-DP$^+$, CRF and CRF$^+$ evaluated based on multiple aspect term occurrences.

Data	DP			DP$^+$			RS-DP			RS-DP$^+$			CRF			CRF$^+$		
	P	R	F$_1$	P	R	F$_1$	P	R	F$_1$	P	R	F$_1$	P	R	F$_1$	P	R	F$_1$
D1	70.69	90.95	79.55	66.23	96.30	78.48	86.17	88.07	87.11	85.21	90.54	87.79	79.26	67.90	73.14	86.58	75.72	80.79
D2	73.63	89.53	80.81	65.67	95.93	77.97	88.64	87.79	88.21	86.99	93.60	90.18	83.96	74.42	78.90	84.11	83.72	83.91
D3	76.53	90.21	82.81	65.56	95.10	77.61	84.16	82.87	83.51	83.33	89.51	86.31	80.00	51.05	62.33	76.74	75.87	76.31
D4	69.73	88.72	78.09	62.21	95.55	75.36	81.01	81.45	81.23	80.75	90.36	85.28	87.01	70.33	77.78	83.78	73.29	78.19
D5	63.04	89.61	74.01	58.82	94.27	72.44	84.06	78.85	81.37	85.47	89.96	87.66	74.40	69.89	72.08	75.14	73.84	74.48
Avg	70.73	89.80	79.05	63.70	95.43	76.37	84.81	83.81	84.29	84.35	90.79	**87.44**	80.93	66.72	72.85	81.27	76.49	78.73
D6	73.80	88.78	80.60	66.27	95.05	78.09	83.90	82.18	83.03	83.06	86.51	84.75	75.34	67.33	71.11	76.56	72.71	74.59
D7	65.53	91.63	76.42	55.76	97.36	70.91	74.29	84.58	79.10	76.92	86.78	81.55	76.22	76.65	76.44	83.72	74.01	78.57
D8	70.98	91.44	79.92	62.11	96.26	75.50	79.87	81.55	80.70	80.89	84.62	82.71	84.73	73.54	78.74	88.08	74.06	80.47
Avg	70.10	90.62	78.98	61.38	96.22	74.83	79.35	82.77	80.94	80.29	85.97	**83.01**	78.76	72.51	75.43	82.79	73.59	77.87

Table 3: Precision, Recall and F$_1$-score of DP, DP$^+$, RS-DP, RS-DP$^+$, CRF and CRF$^+$ evaluated based on distinct aspect terms.

Data	DP			DP$^+$			RS-DP			RS-DP$^+$			CRF			CRF$^+$		
	P	R	F$_1$	P	R	F$_1$	P	R	F$_1$	P	R	F$_1$	P	R	F$_1$	P	R	F$_1$
D1	60.00	83.87	69.96	46.59	91.40	61.72	82.67	74.19	78.20	80.85	75.27	77.96	63.79	52.69	57.71	72.73	58.06	64.57
D2	59.62	78.79	67.87	46.32	89.39	61.02	82.76	75.76	79.10	83.10	83.33	83.22	73.53	54.55	62.63	69.81	75.76	72.66
D3	58.14	81.44	67.85	45.90	87.63	60.25	76.06	67.01	71.25	76.29	73.20	74.71	70.59	45.36	55.23	75.81	55.67	64.20
D4	53.94	74.67	62.63	46.11	88.00	60.51	71.94	66.00	68.84	68.53	78.00	72.96	80.22	50.00	61.60	70.34	56.67	62.77
D5	52.78	76.34	62.41	45.55	87.10	59.82	74.47	58.06	65.25	78.33	67.74	72.65	65.57	50.54	57.08	63.49	53.76	58.22
Avg	56.89	79.02	66.14	46.09	88.70	60.66	77.58	68.21	72.53	77.42	75.51	**76.30**	70.74	50.63	58.85	70.44	59.98	64.48
D6	63.41	78.46	70.14	52.16	88.46	65.62	77.91	69.23	73.31	76.40	74.00	75.18	62.69	53.08	57.48	64.29	56.15	59.94
D7	55.32	84.76	66.95	42.55	94.29	58.63	69.23	74.29	71.67	72.09	79.05	75.41	58.33	65.71	61.80	70.91	60.95	65.55
D8	56.47	80.79	66.48	44.21	90.73	59.45	71.00	66.89	68.88	72.55	68.87	70.66	65.52	55.63	60.17	68.92	56.95	62.37
Avg	58.40	81.34	67.86	46.31	91.16	61.23	72.71	70.14	71.29	73.68	73.97	**73.75**	62.18	58.14	59.82	68.04	58.02	62.62

recall of DP is lower than that of DP$^+$. This is because DP$^+$ uses more rules (due to 10 additional dependency relations) than DP. The new rules bring more correct aspects (higher recall) but also more errors (low precision). Since the average F$_1$-score of DP is higher than that of DP$^+$, we can see that adding rules arbitrarily can harm the overall results for the existing approach in [Qiu *et al.*, 2011].

From Table 2 and Table 3, we can also see that the average precisions of RS-DP and RS-DP$^+$ are dramatically higher than those of DP and DP$^+$ respectively. Although there is some loss in recall, this is expected because RS-DP and RS-DP$^+$ have less rules. However, the final F$_1$-scores of RS-DP and RS-DP$^+$ are markedly better than those of DP and DP$^+$ respectively.

We also observe that the average precisions of RS-DP and RS-DP$^+$ are almost the same, but the average recall of RS-DP$^+$ is higher. This shows that the proposed method is able to take good advantage of the additional rules (in RS-DP$^+$), maintaining a high recall with almost no loss in precision, which finally translates to the high F$_1$-score of RS-DP$^+$.

From the tables, we further see that the recall of CRF$^+$ is improved on almost every dataset compared with CRF, although the precision of CRF$^+$ decreases slightly on some datasets. The overall F$_1$-score of CRF$^+$ is improved by about 6% on D1 to D5, and about 3% on D6 to D8 compared with CRF. This shows that the CRF-based approach is also able to make good use of additional rules. Note that, CRF$^+$ uses the same dependency relations as in RS-DP$^+$. However, both CRF and CRF$^+$ are much poorer than RS-DP and RS-DP$^+$. We believe one of the key reasons is that the rule-based approach performs propagation and make improvements iteratively, which the CRF-based method does not do.

In summary, we can conclude that the proposed approach can take rules of any quality and select a good subset to produce much better results than existing state-of-the-art rule-based and CRF-based approaches.

6 Conclusion

This paper proposed an automated rule set selection/learning method with the goal of improving the syntactical rule-based approach to aspect extraction in opinion mining. The original set of rules can be user-designed or learned by a system and the rules can be of any quality. The proposed technique can select a good subset of the given rules to perform much better extraction than the original rule set because our method can select rules which work best when used together. The experimental results demonstrated its superior performance. We also compared it with the state-of-the-art supervised statistical/learning method CRF. The proposed technique is much more effective. In our future work, we plan to employ semantic rule patterns, which can be learned or designed based on semantic parsing in addition to syntactic rule patterns as in the DP method. We also plan to explore other possible algorithms for rule selection, such as simulated annealing strategies and genetic algorithms.

Acknowledgments

Zhiqiang Gao's research was supported by the 863 program grant 2015AA015406 and the NSF of China grant 61170165. Bing Liu's research was partially supported by the NSF grants IIS-1111092 and IIS-1407927, and a Google faculty award. Yuanlin Zhang's work was partially supported by the NSF grant IIS-1018031. We would also like to thank Yun Lou and Pooja Sampelly for valuable discussions.

References

[Brody and Elhadad, 2010] Samuel Brody and Noemie Elhadad. An unsupervised aspect-sentiment model for online reviews. In *NAACL '10*, pages 804–812, 2010.

[Chen *et al.*, 2014] Zhiyuan Chen, Arjun Mukherjee, and Bing Liu. Aspect extraction with automated prior knowledge learning. In *ACL '14*, pages 347–358, 2014.

[Choi and Cardie, 2010] Yejin Choi and Claire Cardie. Hierarchical sequential learning for extracting opinions and their attributes. In *ACL '10*, pages 269–274, 2010.

[Fang and Huang, 2012] Lei Fang and Minlie Huang. Fine granular aspect analysis using latent structural models. In *ACL '12*, pages 333–337, 2012.

[Hu and Liu, 2004] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *KDD '04*, pages 168–177, 2004.

[Jakob and Gurevych, 2010] Niklas Jakob and Iryna Gurevych. Extracting opinion targets in a single- and cross-domain setting with conditional random fields. In *EMNLP '10*, pages 1035–1045, 2010.

[Jo and Oh, 2011] Yohan Jo and Alice H. Oh. Aspect and sentiment unification model for online review analysis. In *WSDM '11*, pages 815–824, 2011.

[Klinger and Cimiano, 2013] Roman Klinger and Philipp Cimiano. Bi-directional inter-dependencies of subjective expressions and targets and their value for a joint model. In *ACL '13*, 2013.

[Kobayashi *et al.*, 2007] Nozomi Kobayashi, Kentaro Inui, and Yuji Matsumoto. Extracting aspect-evaluation and aspect-of relations in opinion mining. In *EMNLP-CoNLL '07*, pages 1065–1074, 2007.

[Ku *et al.*, 2006] Lun-Wei Ku, Yu-Ting Liang, and Hsin-Hsi Chen. Opinion extraction, summarization and tracking in news and blog corpora. In *AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*, pages 100–107, 2006.

[Lafferty *et al.*, 2001] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML '01*, pages 282–289, 2001.

[Lazaridou *et al.*, 2013] Angeliki Lazaridou, Ivan Titov, and Caroline Sporleder. A bayesian model for joint unsupervised induction of sentiment, aspect and discourse representations. In *ACL '13*, pages 1630–1639, 2013.

[Li *et al.*, 2010] Fangtao Li, Minlie Huang, and Xiaoyan Zhu. Sentiment analysis with global topics and local dependency. In *AAAI '10*, 2010.

[Lin and He, 2009] Chenghua Lin and Yulan He. Joint sentiment/topic model for sentiment analysis. In *CIKM '09*, pages 375–384, 2009.

[Liu *et al.*, 1998] Bing Liu, Wynne Hsu, and Yiming Ma. Integrating classification and association rule mining. In *AAAI '98*, pages 80–86, 1998.

[Liu *et al.*, 2013a] Kang Liu, Liheng Xu, Yang Liu, and Jun Zhao. Opinion target extraction using partially-supervised word alignment model. In *IJCAI '13*, pages 2134–2140, 2013.

[Liu *et al.*, 2013b] Qian Liu, Zhiqiang Gao, Bing Liu, and Yuanlin Zhang. A logic programming approach to aspect extraction in opinion mining. In *WI-IAT '13*, pages 276–283, 2013.

[Liu, 2012] Bing Liu. *Sentiment Analysis and Opinion Mining*. Morgan & Claypool Publishers, 2012.

[Lu *et al.*, 2009] Yue Lu, ChengXiang Zhai, and Neel Sundaresan. Rated aspect summarization of short comments. In *WWW '09*, pages 131–140, 2009.

[Mei *et al.*, 2007] Qiaozhu Mei, Xu Ling, Matthew Wondra, Hang Su, and Chengxiang Zhai. Topic sentiment mixture: Modeling facets and opinions in weblogs. In *WWW '07*, pages 171–180, 2007.

[Mitchell *et al.*, 2013] Margaret Mitchell, Jacqui Aguilar, Theresa Wilson, and Benjamin Van Durme. Open domain targeted sentiment. In *ACL '13*, pages 1643–1654, 2013.

[Moghaddam and Ester, 2011] Samaneh Moghaddam and Martin Ester. ILDA: interdependent lda model for learning latent aspects and their ratings from online product reviews. In *SIGIR '11*, pages 665–674, 2011.

[Mukherjee and Liu, 2012] Arjun Mukherjee and Bing Liu. Aspect extraction through semi-supervised modeling. In *ACL '12*, volume 1, pages 339–348, 2012.

[Popescu and Etzioni, 2005] Ana-Maria Popescu and Oren Etzioni. Extracting product features and opinions from reviews. In *EMNLP '05*, pages 339–346, 2005.

[Qiu *et al.*, 2011] Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. Opinion word expansion and target extraction through double propagation. *Computational Linguistics*, 37(1):9–27, 2011.

[Sauper *et al.*, 2011] Christina Sauper, Aria Haghighi, and Regina Barzilay. Content models with attitude. In *ACL '11*, pages 350–358, 2011.

[Titov and McDonald, 2008] Ivan Titov and Ryan McDonald. A joint model of text and aspect ratings for sentiment summarization. In *ACL '08*, pages 308–316, 2008.

[Wang and Wang, 2008] Bo Wang and Houfeng Wang. Bootstrapping both product features and opinion words from chinese customer reviews with cross-inducing. In *IJCNLP '08*, pages 289–295, 2008.

[Wu *et al.*, 2009] Yuanbin Wu, Qi Zhang, Xuanjing Huang, and Lide Wu. Phrase dependency parsing for opinion mining. In *EMNLP '09*, pages 1533–1541, 2009.

[Zhang *et al.*, 2010] Lei Zhang, Bing Liu, Suk Hwan Lim, and Eamonn O'Brien-Strain. Extracting and ranking product features in opinion documents. In *COLING '10*, pages 1462–1470, 2010.

[Zhao *et al.*, 2010a] Wayne Xin Zhao, Jing Jiang, Hongfei Yan, and Xiaoming Li. Jointly modeling aspects and opinions with a maxent-lda hybrid. In *EMNLP '10*, pages 56–65, 2010.

[Zhao *et al.*, 2010b] Yanyan Zhao, Bing Qin, Shen Hu, and Ting Liu. Generalizing syntactic structures for product attribute candidate extraction. In *NAACL '10*, pages 377–380, 2010.

[Zhao *et al.*, 2014] Yanyan Zhao, Wanxiang Che, Honglei Guo, Bing Qin, Zhong Su, and Ting Liu. Sentence compression for target-polarity word collocation extraction. In *COLING '14*, pages 1360–1369, 2014.

[Zhou *et al.*, 2013] Xinjie Zhou, Xiaojun Wan, and Jianguo Xiao. Collective opinion target extraction in Chinese microblogs. In *EMNLP '13*, pages 1840–1850, 2013.

[Zhu *et al.*, 2009] Jingbo Zhu, Huizhen Wang, Benjamin K. Tsou, and Muhua Zhu. Multi-aspect opinion polling from textual reviews. In *CIKM '09*, pages 1799–1802, 2009.

[Zhuang *et al.*, 2006] Li Zhuang, Feng Jing, and Xiao-Yan Zhu. Movie review mining and summarization. In *CIKM '06*, pages 43–50, 2006.