



Efficient Utilization of Dependency Pattern and Sequential Covering for Aspect Extraction Rule Learning

Fariska Zakhralativa Ruskanda*, Dwi Hendratmo Widyantoro & Ayu Purwarianti

School of Electrical Engineering and Informatics, Institut Teknologi Bandung
Jalan Ganesha 10, Bandung 40132, Indonesia

*E-mail: fariska@informatika.org

Abstract. The use of dependency rules for aspect extraction tasks in aspect-based sentiment analysis is a promising approach. One problem with this approach is incomplete rules. This paper presents an aspect extraction rule learning method that combines dependency rules with the Sequential Covering algorithm. Sequential Covering is known for its characteristics in constructing rules that increase positive examples covered and decrease negative ones. This property is vital to make sure that the rule set used has high performance, but not inevitably high coverage, which is a characteristic of the aspect extraction task. To test the new method, four datasets were used from four product domains and three baselines: Double Propagation, Aspectator, and a previous work by the authors. The results show that the proposed approach performed better than the three baseline methods for the F-measure metric, with the highest F-measure value at 0.633.

Keywords: *aspect-based sentiment analysis; aspect extraction; dependency rule; rule learning; sequential covering.*

1 Introduction

The Internet is used very much to convey opinions and suggestions regarding products, organizations, services, persons, or topics/news. Online media used include blogs, forums, social media, and review websites. The abundance of opinion data can be utilized by organizations/companies to find out feedback from customers and then to improve the quality of products and services. Opinions are often expressed on aspects of a product. Aspects are features or parts of a product. Aspect-based sentiment analysis (ABSA) studies the relationship between aspects of a product and the polarity of opinions on it. One crucial step in ABSA is aspect extraction. In this step, the aspect-opinion pair is extracted, for example, a sentence from a review of a digital camera product: "My friends were impressed with the quality of the pictures I took!" The aspect-opinion pair for this sentence is 'picture quality' and 'impressed'. The results of aspect-opinion pair extraction are then utilized in the next ABSA stage, the sentiment polarity classification.

Received June 28th, 2019, Revised June 25th, 2020, Accepted for publication June 30th, 2020.

Copyright © 2020 Published by IRCS-ITB, ISSN: 2337-5787, DOI: 10.5614/itbj.ict.res.appl.2020.14.1.4

Several approaches can be applied to the aspect extraction task. One of the most natural ones is by using linguistic rules. The most widely used linguistic features are dependency relations, which represent syntactic dependency between terms. The dependency feature is used to find the relationship between the aspect and the opinion term. The advantage of this approach is the ability to trace rule inference. Reasons for selecting rules can be outlined and can be corrected if something goes wrong.

Researchers have begun studying aspect extraction using dependency rule methods decades ago. Some of these are Double-Propagation [1] and Aspectator [2]. The problem with these methods is that the dependency rules used are too general and incomplete, which causes difficulties when extracting sentences with varying structure. This incompleteness problem is indicated by the low recall rate of the extraction result. Besides being incomplete, another problem is the manual process of constructing dependency rules for aspect extraction. This process is done by examining each review sentence and determining the pattern of relations between the aspect and the opinion term. With so many variations of review sentences that exist today, this process has a higher risk of being wrong. The rule construction process can also be done through automatic rule learning, as proposed by Liu, *et al.* [3]. This method has succeeded in improving system performance, but it is still insufficiently flexible because it depends on seed opinion words and a seed rule set.

The problem of the incompleteness of dependency rules motivated us to study rule learning methods for the aspect extraction of ABSA. Our goal was to find a rule learning method that provides better performance, encompasses various sentence structures, and cross-domain usability. Therefore, we propose a new rule learning method for aspect extraction. The proposed method combines the dependency rules with the Sequential Covering rule learning algorithm for aspect extraction. Sequential Covering was used with the consideration that this algorithm has the characteristic of producing rules that include as many positive examples as possible while minimizing negative examples. This algorithm is an improvement of our previous work [4], which also used dependency patterns and Sequential Covering to get the rule set. A modification was performed by adding the use of negative examples to the rule construction process and adding a rule pruning stage at the end of the process. This method is more effective because it does not require seed opinion words or a seed rule set. Moreover, the aspect extraction rules can be applied to products from various domains with more accurate aspect extraction results because it can handle opinion phrase expressions.

2 Related Work

Aspect extraction in ASBA has become an intensive research area over the last 15 years. The main characteristic of this task is that each opinion sentence usually describes a particular aspect or feature. Many approaches have been proposed to complete this task. Sequence models is one of the commonly used methods in this regard. Specific methods included in this model are the Hidden Markov Model (HMM) [5] and Conditional Random Field (CRF) [6-10]. Jakob and Gurevych [7] used the concept of CRF to extract aspects of sentences that contain expressions of opinion. This method represents the possibility of labeling with a begin-inside-outside (BIO) scheme, namely: B-target, which identifies the beginning of an aspect; I-target, which identifies continuation aspects; and O for other tokens. This sequence model is often superior to rule-based linguistic methods, but lifetime depends heavily on manual selection of features and annotated training data [11].

The topic model is another approach that is widely used. The idea is that a set of documents is usually a combination of many topics and each topic contains a probability distribution of words. In aspect extraction, where the topic model is used, it assumes each aspect to be a unigram language model, which is a multinomial distribution of words. It is utilized to group similar words into one aspect group [12-17]. Titov, *et al.* [13] applied Multi-grain Latent Dirichlet Allocation (MG-LDA) to find aspects, using two forms of topic models: global topics and local topics. The distribution of global topics is usually a fixed value for each (review) document, whereas the distribution of local topics varies for each document.

Another classic approach that is still used today is the linguistic rule-based approach. This approach uses word frequency [18], part-of-speech [19], dependency relations [12], or phrases [20,21]. Dependency is one of the language features that is widely used to perform aspect extraction. Pioneers of this approach were Hu and Liu [18]. They proposed an aspect extraction method using rules based on the occurrence frequency of aspect/feature terms and dependency relations. Further development of this method was carried out by Popescu and Etzioni [22], Blair-Goldensohn, *et al.* [23], and Qiu, *et al.* [1]. The algorithm proposed by Qiu is Double Propagation (DP). This algorithm proposes a bootstrapping method using dependency relations to extract aspects simultaneously. The input of this method is a set of opinion words as a seed to extract aspect words and other opinion words in a sentence. There are eight handcrafted rules that the DP uses to extract the aspect and opinion terms. The DP method was extended in Aspectator [2]. This algorithm has the advantage of

not requiring a seed word, using ten handmade rules for the extraction of aspect-opinion pairs. Rule-based methods can also be combined with deep learning techniques for ABSA, as performed by Ray and Chakrabarti [24]. They used deep-CNN architecture to extract aspects and polarity, and they used a set of rules to improve the performance. The combination of BiLSTM and CRF deep learning methods was used by Luo, *et al.* to obtain syntactic information more efficiently through bottom-up and top-down propagation on the dependency tree [24]. The last two methods provide a high F-measure but require a huge dataset for training the deep-learning model.

The dependency rules for aspect extraction can also be obtained by conducting the learning process automatically, as has been carried out by Liu, *et al.* [25]. The linguistic rules used are syntactic rules and the approach used is to choose the best set of rules from a set of existing rules using three steps, namely: rule evaluation, rule ranking, and rule selection. The input of this method is a set of initial rules, a set of seed opinion words, and annotated training data. In the field of sentiment analysis, rule learning, especially by using Sequential Covering, has been done on several tasks. In the comparative sentence-mining task, Sequential Covering was deployed by Jindal and Liu [26]. This method extracts comparative relations based on two types of rules, namely class sequential rules and sequential rule labels. The linguistic features used with Sequential Covering are part-of-speech tags and keywords. Sequential Covering has also been utilized in subjective and objective text classification tasks by Chao and Jiang [27]. This method uses Sequential Covering to construct classification rules for informal sentences in Chinese review sentences. The contribution of this research was to combine dependency relation features with Sequential Covering rule learning to obtain aspect extraction rules that can include an optimal variation of review sentences.

3 The Proposed Method

Aspect extraction using dependency rules (Figure 1) is performed in 3 steps, namely: (1a) extraction of aspect-opinion expressions; and (1b) filtering out aspect-opinion pairs, and (2) aspect and opinion expression expansion. The first step is done by using a set of extraction rules to capture aspect terms and their opinion term pairs, adopted from the Aspectator method [2]. Then, every aspect-opinion pair is filtered by the sentiment lexicon and aspect and opinion list to get an aspect-opinion pair that matches the product domain. The last step is to expand the results so that we can capture the aspect and opinion phrase. This extraction mechanism is also fully explained in our previous work [4].

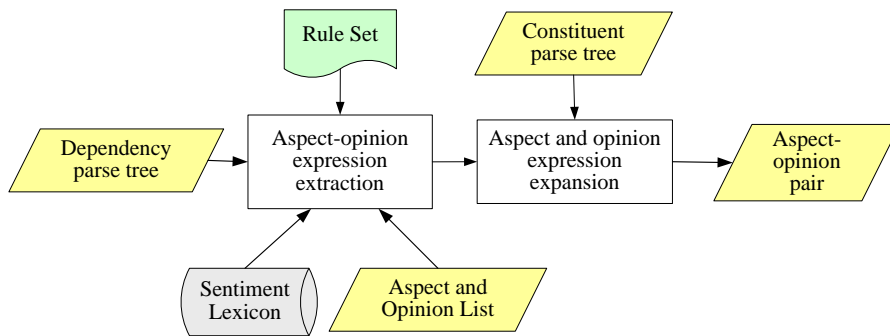


Figure 1 Aspect-opinion extraction process.

In this paper, we propose a method of learning the aspect extraction rules using the Sequential Covering algorithm. Sequential Covering is a basic algorithm that learns rules directly from a set of examples [28,29]. In principle, this algorithm accepts input in the form of a set of positive and negative examples, then gives the result in the form of a proposition logic rule, i.e. disjunctive normal form (DNF). Each rule includes several positive examples and at the same time minimize the negative examples. This rule is then used to classify other cases that are not in the example.

Input	: Dependency parse tree, annotated aspect-opinion pair
Output	: Aspect extraction rule set
1	Construct positive and negative pattern examples, each pattern is a feature; then Create positive occurrence table and negative occurrence table,
2	Calculate performance for each feature: Performance(f) = Ppos(f) - w*Pneg(f)
3	Prune features below the performance threshold
4	Update features; Sort features based on decimal value
5	Construct DNF rule using SC traversal
6	Prune rules

Figure 2 Dependency pattern-sequential covering algorithm.

We adopted the Sequential Covering algorithm mechanism to learn the aspect-opinion extraction rules. The positive example we use is the dependency relation that connects each aspect in the dataset to its opinion pair in the sentence. Meanwhile, the negative example is the non-aspect-opinion pair relation, namely the dependency relation that connects the aspect/opinion term with other words around it. These positive and negative examples are used by the algorithm to learn extraction rules that include as many examples of aspect-opinion pairs as possible while minimizing examples of non-aspect-opinion pairs. These extraction rules

are used to extract the aspect-opinion pairs from other review sentences. The adaptation of this algorithm was inspired by the ESCAPE algorithm [30], which uses Sequential Covering to generate an explanation of a subset of data in the form of DNF rules.

Figure 2 shows a complete description of the dependency pattern-sequential covering algorithm. In the first step, this method reads each review sentence and makes annotated aspect-opinion pairs dependency relations become positive pattern examples. Then for each annotated aspect term, all relations to the other terms around that aspect term are identified. The relation pattern then becomes a negative pattern example. The same applies to the pattern of relations around the opinion term. The result is a set of positive examples and a set of negative examples.

Sentence ID	Sentence Example
S1	<p>This (OTHER) camera (NOUN) is (VERB) perfect (ADJ) for an enthusiastic amateur photographer (NOUN)</p> <p><asp> <op></p>
S2	<p>settings for just about everything (NOUN), beautiful (ADJ) pics (NOUN) and beautiful prints</p> <p><op> <asp></p>
S3	<p>the (OTHER) pictures (NOUN) are (VERB) razor sharp (ADJ) even (ADV) in (OTHER) macro (NOUN)</p> <p><asp1> <op> <asp2></p>

Figure 3 Examples of review sentences and their dependency patterns.

Every different pattern in these examples becomes a feature. For example, check out the three review sentences in Figure 3, which contain an aspect-opinion pair annotation (<asp>, <op>), a POS-tag, and interword dependency relations. There are nine different patterns (P1-P9) around the aspect-opinion pair (Figure 4) and some examples of positive patterns and examples of negative patterns for each sentence (Figure 5(a)). Each aspect-opinion pair in the review sentence that we used is the result of an annotation process performed manually by an expert by selecting the most critical aspects in a sentence and its opinion pair. After that, a positive occurrence table and a negative occurrence table are created.

In the positive occurrence table, each row is an aspect-opinion pair in each review sentence, while each column is a pattern. A row is '1' if the pattern appears for specific aspect-opinion pairs and '0' if it does not. The same is done for negative patterns to produce a negative occurrence table. Examples of positive and negative occurrence tables can be seen in Figure 5(b)-(c). Then, for each pattern feature, the occurrence frequency in the positive emergence table and the negative emergence table is calculated.

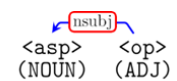


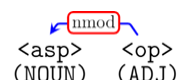
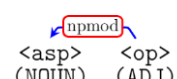
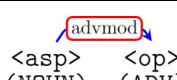
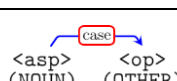
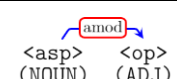
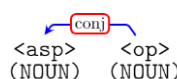
ID	Pattern	Extracted From	Performance
P1	 <asp> <op> (NOUN) (ADJ)	S1	0.5
P2	 <asp> <op> (NOUN) (OTHER)	S1	<u>-3.75</u>
P3	 <asp> <op> (VERB) (ADJ)	S1	<u>-5.625</u>
P4	 <asp> <op> (NOUN) (ADJ)	S1	-1.625
P5	 <asp> <op> (NOUN) (ADJ)	S3	<u>-3.75</u>
P6	 <asp> <op> (NOUN) (ADV)	S3	-1.875
P7	 <asp> <op> (NOUN) (OTHER)	S3	-1.875
P8	 <asp> <op> (NOUN) (ADJ)	S2	0.25
P9	 <asp> <op> (NOUN) (NOUN)	S2	-1.875

Figure 4 Extracted patterns from review sentences.

The second step is to calculate the performance of each pattern feature using the current frequency value. This performance value is calculated to get the

quantification value of a pattern feature in the positive occurrence table. The performance formula is given in Eq. (1):

$$\text{Performance}(f) = P_{\text{pos}}(f) - w * P_{\text{neg}}(f) \quad (1)$$

where f is a pattern feature, P_{pos} is the probability of a pattern appearing in the positive occurrence table, and P_{neg} is the probability of a pattern appearing in the negative occurrence table. The value w is added to give weight as a comparison to the occurrence in the positive and negative occurrence tables. Figure 4 (performance column) shows each pattern's performance value by using the w value of 7.5.

In the next step, the calculated performance value is used to determine which pattern is suitable as part of the conjunction rules. This feasibility can be seen from whether the performance value is quite significant, i.e. above the threshold limit. If the performance value is insufficient, the pattern is discarded. This leaves a reasonably exclusive set of patterns. After removing non-exclusive patterns, the positive and negative occurrences tables need to be updated so that they do not contain non-exclusive patterns.

For example, by using a threshold value of -2.0, three patterns are eliminated: P2, P3, and P5. The fourth step is to sort each remaining positive occurrence table row based on the decimal value. Every row should be translated to a single binary value. Each binary value is then converted to decimals and sorted according to the decimal value. This sorting is done by using the principle of adjacent decimal values, which means having more proximity for each bit in the binary value. The result of this step is a positive occurrence table sorted by decimal value. With the completion of step 4, the drafting of DNF rules is ready (Figure 5(d)).

In the next step, DNF rules are constructed using the Sequential Covering traversal algorithm. The technique is to read the lines of the positive appearance table one by one from beginning to end, then compare each row with the adjacent rows. In principle, if two different lines contain several bits that have a common value, then some of these bits are conjunctions. This is continued for the following lines. When an inequality is found, the last conjunction is stored as a rule. Later the same process begins again for new lines and neighbors to get other rules. For example, all four rows in the occurrence table produce a set of rules [P1, P4, P8] (Figure 5(e)). The final step is to eliminate the rules with low performance. Pruning rules were also created to avoid overlapping rules. The process of reviewing overlapping and low-performing rules and determining some rules that need to be eliminated is done by hand.

	Positive Examples	Negative Examples
Sentence ID	Pattern ID	Pattern ID
S1	P1	P2, P3, P4
S2	P8	P9
S3a	P1	P2, P3, P5
S3b	P4	P3, P5, P6, P7

(a)

	P1	P2	P3	P4	P5	P6	P7	P8	P9
S1	1	0	0	0	0	0	0	0	0
S2	0	0	0	0	0	0	0	1	0
S3a	1	0	0	0	0	0	0	0	0
S3b	0	0	0	1	0	0	0	0	0

(b)

	P1	P2	P3	P4	P5	P6	P7	P8	P9
S1	0	1	1	1	0	0	0	0	0
S2	0	0	0	0	0	0	0	0	1
S3a	0	1	1	0	1	0	0	0	0
S3b	0	0	1	0	1	1	1	0	0

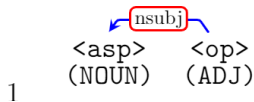
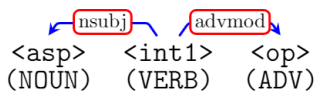
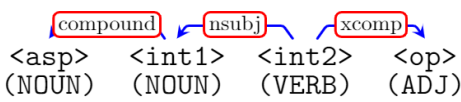
(c)

	P1	P4	P6	P7	P8	P9
S1	1	0	0	0	0	0
S3a	1	0	0	0	0	0
S3b	0	1	0	0	0	0
S2	0	0	0	0	1	0

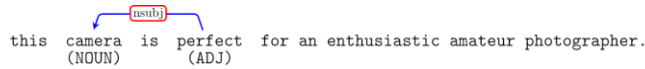
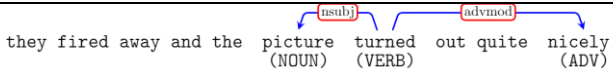
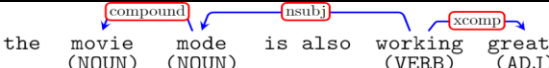
(d)

Step	Current Row	Conjunction	Intersection	Stored Conjunction
0	-	[]	[]	[]
1	S1: [P1]	[P1]	[P1]	-
2	S3a: [P1]	[P1]	[P1]	-
3	S3b: [P4]	[P4]	[]	[P1]
4	S2: [P8]	[P8]	[]	[P1, P4]
5	-	-	[]	[P1, P4, P8]

Figure 5 (a) Positive and negative pattern example; (b) positive occurrence table; (c) negative occurrence table; (d) updated and sorted positive occurrence table; (e) rule covering example.

Rule ID	Rule (Dependency graph form)	Rule (DNF form)
R1	 $\langle \text{asp} \rangle$ (NOUN) $\langle \text{op} \rangle$ (ADJ)	IF isNOUN(w1) AND isADJ(w2) AND nsubj(w2, w1) THEN $\langle w1, w2 \rangle$
R4	 $\langle \text{asp} \rangle$ (NOUN) $\langle \text{int1} \rangle$ (VERB) $\langle \text{op} \rangle$ (ADV)	IF isNOUN(w1) AND isNOUN(w2) AND isADJ(w3) AND nsubj(w3, w2) AND compound(w2, w1) THEN $\langle w1, w3 \rangle$
R10	 $\langle \text{asp} \rangle$ (NOUN) $\langle \text{int1} \rangle$ (NOUN) $\langle \text{int2} \rangle$ (VERB) $\langle \text{op} \rangle$ (ADJ)	IF isNOUN(w1) AND isNOUN(w2) AND isVERB(w3) AND isADJ(w4) AND xcomp(w3, w4) AND nsubj(w3, w2) AND compound(w2, w1) THEN $\langle w1, w4 \rangle$

(a)

Rule applied	Review Sentence	Pair extracted
R1	 this camera is perfect for an enthusiastic amateur photographer.	$\langle \text{"camera"}, \text{"perfect"} \rangle$
R4	 they fired away and the picture turned out quite nicely.	$\langle \text{"picture"}, \text{"nicely"} \rangle$
R10	 the movie mode is also working great.	$\langle \text{"movie mode"}, \text{"great"} \rangle$

(b)

Figure 6 (a) Rule examples and rule conversion to IF-THEN form; (b) Application of rules to review sentences.

We implemented this rule learning process to a review datasets from the digital camera domain: Canon G3 and Nikon Coolpix 4300 datasets obtained from [18], which contained 450 annotated aspects (Table 1). These datasets had previously been re-annotated to support the process of rule learning, which requires information on aspect-opinion pairs (Figure 7) that was not contained in the original dataset. Our previous work [4] provides a full description of the re-annotation process. In the training process, we conducted experiments to

determine the combination of w and *threshold* values that would give the optimum result.

We used variations of w values of 5.0, 7.5, and 10.0 combined with *threshold* values of 0.0, -0.5, and -1.0. The result showed that the most optimum aspect extraction performance was obtained with the combination of $w = 7.5$ and *threshold* = -1.0. As for the rules, only the rules with performance above the *threshold* were taken, which means that the rule occurs more often in the positive examples than in the negative examples. For instance, the rule with the best performance was R1, appearing 161 times in the positive examples and 7 times in the negative examples. The rule set produced in this learning process consisted of 113 rules. A rule example and its application on review sentences can be seen in Figure 6.

Table 1 Dataset statistics.

Review Dataset		Number of Sentences	Number of Aspects
Canon G3	Training	569	262
Nikon Coolpix 4300	Data	329	188
Nokia 6610		518	295
Creative Labs Nomad Jukebox	Testing	1611	716
Apex AD2600 Progressive-scan	Data	659	339
ABSA16_Restaurants_Train_SB1		1708	1880
Total		5394	3680

```

camera[perfect,+2]##this camera is perfect for an enthusiastic amateur
photographer .
picture[beautiful,+2],print[beautiful,+2]##settings for just about
everything , beautiful pics and beautiful prints .
picture[razor-sharp,+3], macro[razor-sharp,+3]##the pictures are
razor-sharp , even in macro .
    
```

Figure 7 Example of the content of re-annotated Nikon Coolpix 4300 dataset.

4 Experiments and Results

The purpose of this research, as stated in Section 2, was to obtain extraction rules that include optimal variations of review sentences. Experiments were carried out by applying them to various domains of review sentences. As Table 1 shows, for testing we used four datasets from different domains: Nokia 6610, Nomad Jukebox Creative Labs, Apex AD2600 Progressive-scan, and ABSA16_Restaurants_Train_SB1 [18, 31]. Testing in various domains was carried out to investigate whether the resulting rule set was flexible toward different types of sentences.

The experimental steps are shown in Figure 8. For constituent and dependency parsing, we used Stanford Parser and Stanford Dependency Parser, which have more than 90% accuracy [32]. For the sentiment lexicon, we used SentiWordNet [33]. We used the aspect and opinion lists from our previous work [4]. Evaluation of the results was carried out using the metrics: recall, and F-measure [33].

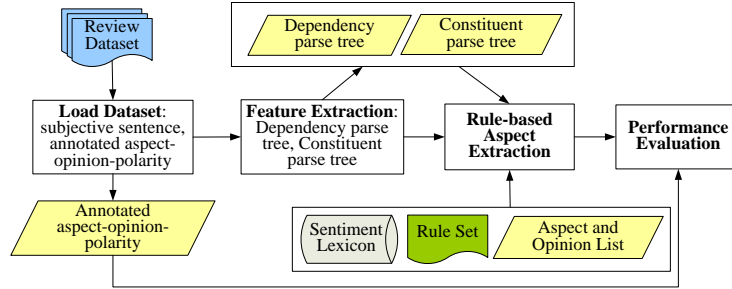


Figure 8 Aspect extraction lexicon experiment steps.

4.1 Experiments Results

In the experiment, we used three methods as baselines: Double Propagation (DP) [1], Aspectator (APT) [2], and our previous work (Dependency with Sequential Covering, DSC1) [4]. The part of the DP and APT methods that we used was the aspect extraction rule set. Tables 2 to 3 and Figure 9 show the results of the aspect extraction experiments for the baselines and our proposed method (DSC2).

Table 2 Experiment results for recall.

Dataset	DP	APT	DSC1	DSC2
Nokia 6610	0.14	0.281	0.66	<u>0.763</u>
Creative Labs Nomad Jukebox	0.173	0.272	0.58	<u>0.725</u>
Apex AD2600 Progressive-scan	0.062	0.103	0.4	<u>0.699</u>
ABSA16_Restaurants_Train_SB1	0.342	0.381	0.59	<u>0.771</u>
Average	0.1793	0.2593	0.56	<u>0.7395</u>

Table 3 Experiment Results for Precision

Dataset	DP	APT	DSC1	DSC2
Nokia 6610	0.532	0.281	<u>0.58</u>	0.531
Creative Labs Nomad Jukebox	0.353	0.245	<u>0.43</u>	0.348
Apex AD2600 Progressive-scan	0.382	0.133	<u>0.51</u>	0.442
ABSA16_Restaurants_Train_SB1	0.37	0.424	<u>0.54</u>	0.536
Average	0.4093	0.2708	<u>0.51</u>	0.4643

Based on the results of the comparison, it can be seen that our method outperformed all baselines for the four datasets for recall. As for precision, our method outperformed the first two baselines for 2 datasets: Apex and Restaurants, and for F-measure it was superior with all datasets except for the Creative dataset.

4.2 Result Analysis

This result means that the proposed method generally performed better than the other three methods, with an average increase in recall and F-measure of 0.41 and 0.22, respectively. In particular, when compared to DSC1, DSC2 was almost always superior, except for the precision score. This shows that DSC2's rule set coverage increased, which was caused by the handling of negative examples and the efficiency of rule pruning. Meanwhile, the low precision of DSC2 was caused by the rule set extracting many wrong aspects. However, these errors were still tolerable since they were smaller than the increase in recall value. In the following subsection we present an analysis of some extraction errors that occurred.

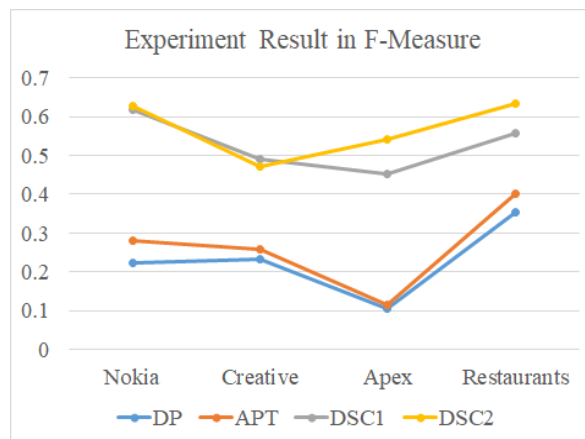


Figure 9 Experiment result in f-measure.

4.2.1 The Use of Dependency Rules on Complex Sentences

On complex sentences, which contain a subordinate clause that gives an explanation of the main clause, the application of simple dependency rules is less appropriate. The extracted aspect-opinion pair may be an explanation of the aspect word or opinion word in the main clause. For example, in the sentence in Figure 10(a) using rule R1 (Figure 5(a)), extract the aspect-opinion pair ('battery', 'low'). The pair is not proper for being an opinion-aspect because the

clause “when battery is low” is an explanation of the opinion expression “does not work well”.

4.2.2 Rule Conflict

A situation that was often found is rule conflict, which means that several rules can be used for the same sentence, for example, for the identical aspect term. However, each of them produced different opinion pairs, but not all of them were true. For example, in the sentence in Figure 10(b).

By using the R123 rule (Figure 10(g)), an aspect-opinion pair (‘price’, ‘great’) was obtained. However, with the R100 rule (Figure 10(f)), another pair (‘features’, ‘great’) was obtained. A mechanism for selecting rules and identifying additional conditions that are conditions for (or not applicable) rules is needed.

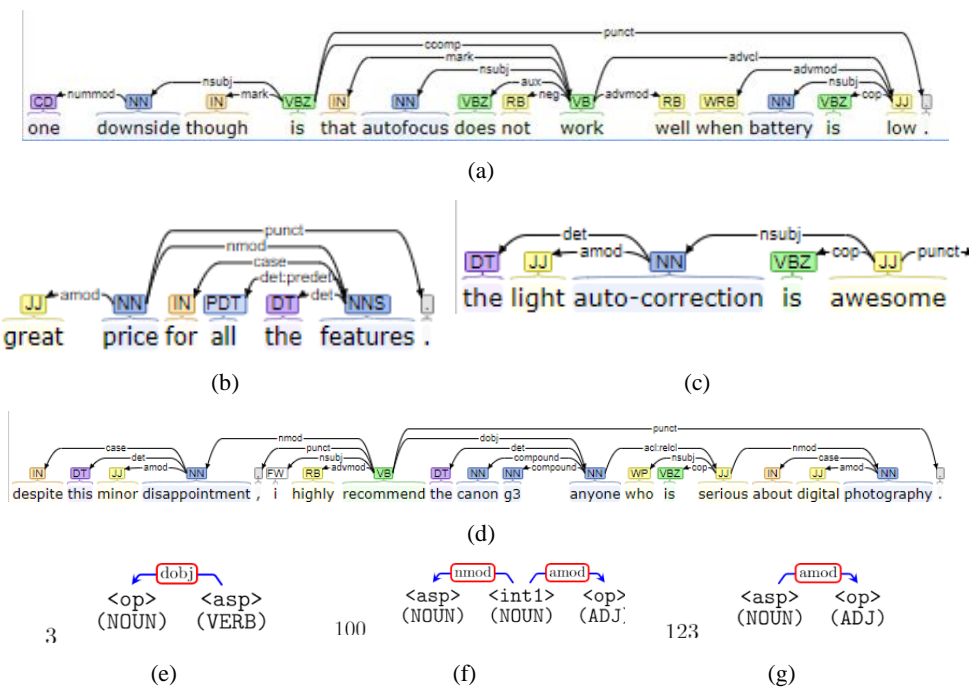


Figure 10 Review sentences and rules.

4.2.3 Aspect/opinion Term Ambiguity

We also found ambiguity in the aspect/opinion term, which can cause extraction errors. For example, in sentence 10(c), the word ‘light’ is classified as an adjective (JJ) and if we use the R123 rule (Figure 10(g)), a wrong aspect-opinion pair is generated (‘auto-correction’, ‘light’). The annotated pair of aspect-opinion is (‘light auto-correction’, ‘awesome’).

4.2.4 Dependency Parser Fallacy

Because this method is based on dependency rules, its performance is very dependent on the quality of the dependency parser. Errors of the dependency parser can result in extraction errors. One of the weaknesses of the existing dependency parser is incomplete sentence handling. For example, Figure 10(d) sentence. By using the R3 rule (Figure 10(e)), (‘canon g3 anyone’, ‘highly recommended’) is generated. The word ‘anyone’ is considered to have a ‘compound’ relation with ‘g3’ so that it is extracted as the aspect phrase ‘canon g3 anyone’.

5 Conclusion and Future Works

In this paper, we proposed an efficient aspect extraction rule learning method based on dependency pattern and the Sequential Covering algorithm. The purpose of this research was to produce a rule learning algorithm that has broad coverage and can be used for many data domains. The experimental results showed that our approach was superior to the baseline methods with an average increase in F-measure of 0.22 with a highest F-measure of 0.633. The resulting dependency rules were simple and easy to understand. This method also has the advantage of not requiring a seed word.

However, there were some errors in the aspect extraction results. These include rule conflicts, use of complex sentences, word ambiguity, and dependency parser errors. In a future work, we plan to make improvements in the rule pruning stage by running it automatically and using complete external domain knowledge.

Acknowledgement

This research was supported by a Doctoral Dissertation Research Grant from the Ministry of Research, Technology, and Higher Education Republic of Indonesia within the research project Utilization of Complex Feature and External Knowledge for Aspect-based Sentiment Analysis.

References

- [1] Qiu, G., Liu, B., Bu, J. & Chen, C., *Opinion Word Expansion and Target Extraction through Double Propagation*, Computational Linguistics, **37**(1), pp. 9-27, 2011.
- [2] Bancken, W., Alfarone, D. & Davis, J., *Automatically Detecting and Rating Product Aspects from Textual Customer Reviews*, Proceedings of Workshop on Interactions between Data Mining and Natural Language Processing (DMNLP), pp. 1-16, 2014.
- [3] Liu, Q., Gao, Z., Liu, B. & Zhang, Y., *Automated Rule Selection for Aspect Extraction in Opinion Mining*, IJCAI Int. Jt. Conf. Artif. Intell., pp. 1291-1297, 2015.
- [4] Ruskanda, F.Z., Widiantoro, D.H. & Purwarianti, A., *Sequential Covering Rule Learning for Language Rule-based Aspect Extraction*, Proceeding of 2019 International Conference on Advanced Computer Science and Information Systems (ICACSIS), pp. 229-234, 2019.
- [5] Jin, W. & Ho, H.H., *A Novel Lexicalized HMM-based Learning Framework for Web Opinion Mining*, Proceedings of the 26th Annual International Conference on Machine Learning – ICM, pp. 1-8, 2009.
- [6] Toh, Z. & Wang, W., *DLIREC: Aspect Term Extraction and Term Polarity Classification System*, Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), SemEval, pp. 235-240, 2014.
- [7] Jakob, N. & Gurevych, I., *Extracting Opinion Targets in a Single-and Cross-domain Setting with Conditional Random Fields*, Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 1035-1045, 2010.
- [8] Yang, B. & Cardie, C., *Joint Inference for Fine-grained Opinion Extraction*, Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pp. 1640-1649, 2013.
- [9] Samha, A.K., Li, Y. & Zhang, J., *Aspect-based Opinion Mining from Product Reviews Using Conditional Random Fields*, Data Mining and Analytics: Proceedings of the 13th Australasian Data Mining Conference, pp. 119-128, 2015.
- [10] Mitchell, M., Aguilar, J., Wilson, T. & Van Durme, B., *Open Domain Targeted Sentiment*, Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP 2013), pp. 1643-1654, 2013.
- [11] Wu, C., Wu, F., Wu, S., Yuan, Z. & Huang, Y., *A Hybrid Unsupervised Method for Aspect Term and Opinion Target Extraction*, Knowledge-Based Systems, **148**, pp. 66-73, 2018.

- [12] Zhang, L. & Liu, B., *Aspect and Entity Extraction for Opinion Mining*, in *Data Mining and Knowledge Discovery for Big Data*, Studies in Big Data, Berlin, Heidelberg: Springer, pp. 1-14, 2014.
- [13] Titov, I. & McDonald, R., *Modeling Online Reviews with Multi-grain Topic Models*, Proceedings of the 17th International Conference on World Wide Web, pp. 111-120, 2008.
- [14] Moghaddam, S. & Ester, M., *On the Design of LDA Models for Aspect-based Opinion Mining*, Proceedings of the 21st ACM International Conference on Information and Knowledge Management – CIKM '12, pp. 803-812, 2012.
- [15] Chen, Z., Mukherjee, A. & Liu, B., *Aspect Extraction with Automated Prior Knowledge Learning*, 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014 – Proceedings of the Conference, **1**, pp. 347-358, 2014.
- [16] Mukherjee, A. & Liu, B., *Aspect Extraction through Semi-supervised Modeling*, ACL '12 Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers – Volume 1, July, pp. 339-348, 2012.
- [17] Wang, T., Cai, Y., Leung, H., Lau, R.Y.K., Li, Q. & Min, H., *Product Aspect Extraction Supervised with Online Domain Knowledge*, Knowledge-Based Systems, **71**, pp. 86-100, 2014.
- [18] Hu, M. & Liu, B., *Mining Opinion Features in Customer Reviews*, American Association for Artificial Intelligence, pp. 755-760, 2004.
- [19] Akhtar, M.S., Gupta, D., Ekbal, A. & Bhattacharyya, P., *Feature Selection and Ensemble Construction: A Two-step Method for Aspect Based Sentiment Analysis*, Knowledge-Based Systems, **125**, pp. 116-135, 2017.
- [20] Wu, Y., Zhang, Q., Huang, X.X. & Wu, L., *Phrase Dependency Parsing for Opinion Mining*, Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, **3**, pp. 1533-1541, 2009.
- [21] Nguyen, T.H. & Shirai, K., *PhraseRNN: Phrase Recursive Neural Network for Aspect-based Sentiment Analysis*, Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 2509-2514, 2015.
- [22] Popescu, A.M. & Etzioni, O., *Extracting Product Features and Opinions from Reviews*, Proceedings of the Conference on Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP), pp. 339-346, 2005.
- [23] Blair-Goldensohn, S., Hannan, K., McDonald, R., Neylon, T., Reis, G.a. & Reynar, J., *Building a Sentiment Summarizer for Local Service Reviews*,

- WWW Workshop on NLP in the Information Explosion Era, pp. 339-348, 2008.
- [24] Luo, H., Li, T., Liu, B., Wang, B. & Unger, H., *Improving Aspect Term Extraction with Bidirectional Dependency Tree Representation*, IEEE/ACM Transactions on Audio Speech and Language Processing, **27**(7), pp. 1201-1212, 2019.
- [25] Liu, Q., Gao, Z., Liu, B. & Zhang, Y., *Automated Rule Selection for Opinion Target Extraction*, Knowledge-Based Systems, **104**(C), pp. 74-88, 2016.
- [26] Jindal, N. & Liu, B., *Mining Comparative Sentences and Relations*, AAAI'06 Proceedings of the 21st National Conference on Artificial Intelligence, pp. 1331-1336, 2006.
- [27] Chao, C. & Jiang, W., *Study on the Subjective and Objective Text Classification and Pretreatment of Chinese Network Text*, 2012 4th International Conference on Intelligent Human-Machine Systems and Cybernetics, **2**, pp. 25-29, 2012.
- [28] Mitchell, T.M., *Learning Set of Rules*, in *Machine Learning*, McGraw-Hill, pp. 274-306, 1997.
- [29] Furnkranz, J., Gamberger, D., & Lavrac, N., *Foundations of Rule Learning*, Springer, 2012.
- [30] Michelson, M. & Macskassy, S., *An Efficient Sequential Covering Algorithm for Explaining Subsets of Data*, Proceedings of International Conference on Artificial Intelligence ICAI, pp. 261-267, 2010.
- [31] Pontiki, M., Galanis, D., Papageorgiou, H., Androutsopoulos, I., Manandhar, S., AL-Smadi, M., Al-Ayyoub, M., Zhao, Y., Bing Qin, B., De Clercq, O., Hoste, V., Apidianaki, M., Tannier, X., Loukachevitch, N., Kotelnikov, E., Bel, N., Jiménez-Zafra, S.M. & Eryigit, G., *SemEval-2016 Task 5: Aspect Based Sentiment Analysis*, Proceedings of SemEval-2016, pp. 19-30, 2016.
- [32] Chen, D. & Manning, C.D., *A Fast and Accurate Dependency Parser using Neural Networks*, Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP) , pp. 740-750, 2014.
- [33] Powers, D.M.W., *Evaluation : From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation*, Journal of Machine Learning Technologies, **2**(1), pp. 37-63, 2007.
- [34] Li, Q., Jin, Z., Wang, C. & Zeng, D.D., *Mining Opinion Summarizations using Convolutional Neural Networks in Chinese Microblogging Systems*, Knowledge-Based Systems, **107**, pp. 289-300, 2016.