Generalizing Syntactic Structures for Product Attribute Candidate Extraction

Yanyan Zhao, Bing Qin, Shen Hu, Ting Liu
Harbin Institute of Technology, Harbin, China
{yyzhao,bqin,shu,tliu}@ir.hit.edu.cn

Abstract

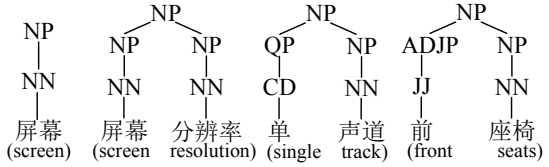
Noun phrases (NP) in a product review are always considered as the product attribute candidates in previous work. However, this method limits the recall of the product attribute extraction. We therefore propose a novel approach by generalizing syntactic structures of the product attributes with two strategies: intuitive heuristics and syntactic structure similarity. Experiments show that the proposed approach is effective.

1 Introduction

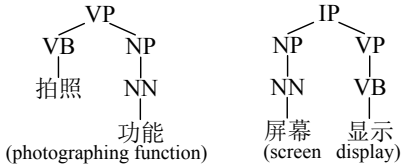
Product attribute extraction is a fundamental task of sentiment analysis. It aims to extract the product attributes from a product review, such as "picture quality" in the sentence "The picture quality of Canon is perfect." This task is usually performed in two steps: product attribute candidate extraction and candidate classification.

Almost all the previous work pays more attention to the second step, fewer researchers make in-depth research on the first step. They simply choose the NPs in a product review as the product attribute candidates (Hu and Liu, 2004; Popescu and Etzioni, 2005; Yi et al., 2003). However, this method limits the recall of the product attribute extraction for two reasons. First, there exist other structures of the product attributes except NPs. Second, the syntactic parsing is not perfect, especially for the Non-English languages, such as Chinese. Experiments on three Chinese datasets[1] show that nearly 15% product attributes are lost, when only using NPs as the candidates. Obviously, if using the candidate classification techniques on these NP candidates, it would lead to poor performance (especially for recall) for the final product attribute extraction.

Based on the above discussion, it can be observed that product attribute candidate extraction is well worth studying. In this paper, we propose an approach by generalizing the *syntactic structures* of the product attributes to solve this problem. Figure 1 lists some syntactic structure samples from an annotated corpus, including the special forms of NPs in Figure 1(a) and other syntactic structures, such as VP or IP in Figure 1(b). We can find that the syntactic structures can not only cover more phrase types besides NP, but also describe the detailed forms of the product attributes.



(a) syntactic structure samples of NP



(b) syntactic structure samples of other phrases

Figure 1: Syntactic structure samples of the product attributes (acquired by an automatic phrase parser).

In order to exploit more and useful syntactic structures, two generalization strategies: intuitive heuristics and syntactic structure similarity are used. Experiments on three Chinese domain-specific datasets show that our approach can significantly improve the recall of the product attribute candidate extraction, and furthermore, improve the performance of the final product attribute extraction.

[1] It refers to the training data in Section 3.1.

Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL, pages 377–380,
Los Angeles, California, June 2010.

2 Approach

The standard syntactic structures of the product attributes can be collected from a training set[2]. Then a simple method of *exact matching* can be used to select the product attribute candidates from the test set. In particular, for a syntactic structure[3] T in the test set, if T exactly matches with one of the standard syntactic structures, then its corresponding string can be treated as a product attribute candidate.

However, this method fails to handle similar syntactic structures, such as the two structures in Figure 2. Besides, this method treats the syntactic structure as a whole during exact matching, without considering any structural information. Therefore, it is difficult to describe the syntactic structure information explicitly. All of these prevent this method from generalizing unseen data well.

To overcome the above problems, two generalization strategies are proposed in this paper. One is to generalize the syntactic structures with two intuitive heuristics. The other is to deeply mine the syntactic structure by decomposing it into several substructures. Both strategies will be introduced in the following subsections.

2.1 Intuitive Heuristics

Two intuitive heuristics are adopted to generalize the syntactic structures.

Heu1: For the near-synonymic grammar tags in syntactic structures, we can generalize them by a normalized one. Such as the red boxes in Figure 2, the POSs "NNS" and "NN" show the same syntactic meaning, we can generalize "NNS" with "NN". The near-synonymic grammar tags are listed in Table 1.

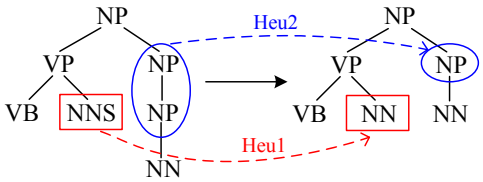


Figure 2: Generalizing a syntactic structure with two intuitive heuristics.

Heu2: For the sequence of identical grammar tags in syntactic structures, we can replace them with one. The reason is that the sequential grammar tags always describe the same syntactic function as one grammar tag. Such as the blue circles in Figure 2.

Replaced by	Near-synonymic grammar tags
JJ	JJR, JJS
NN	NNS, NNP, NNPS, CD, NR
RB	RBR, RBS
VB	VBD, VBG, VBN, VBP, VBZ, VV
S	SBAR, SBARQ, SINU, SQ

Table 1: The near-synonymic grammar tags.

2.2 Syntactic Structure Similarity

The heuristic generalization strategy is too restrictive to give a good coverage. Moreover, after this kind of generalization, the syntactic structure is used as a whole in exact matching all the same. Thus, as an alternative to the exact matching, tree kernel based methods can be used to implicitly explore the substructures of the syntactic structure in a high-dimensional space. This kind of methods can directly calculate the similarity between two substructure vectors using a kernel function. Tree kernel based methods are effective in modeling structured features, which are widely used in many natural language processing tasks, such as syntactic parsing (Collins and Duffy, 2001) and semantic role labeling (Che et al., 2008) and so on.

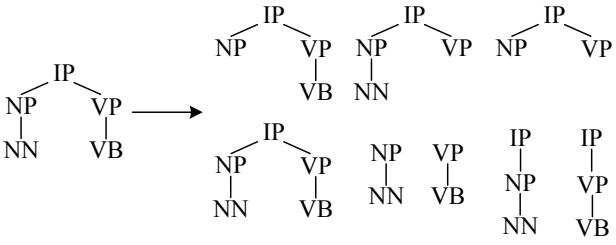


Figure 3: Substructures from a syntactic structure.

In this paper, the syntactic structure for a product attribute can be decomposed into several substructures, such as in Figure 3. Correspondingly, the syntactic structure T can be represented by a vector of integer counts of each substructure type:

$$\begin{aligned}\Phi(T) &= (\phi_1(T), \phi_2(T), ..., \phi_n(T)) \\ &= (\#\text{ of substructures of type 1}, \\ &= \#\text{ of substructures of type 2}, \\ &\quad ..., \\ &= \#\text{ of substructures of type } n)\end{aligned}$$

[2]We use Dan Bikel's phrase parser for syntactic parsing.

[3]We simply select the syntactic structures of the strings under three words or four words with "的"("of" in English).

After syntactic structure decomposition, we can count the number of the common substructures as the similarity between two syntactic structures. The commonly used convolution tree kernel is applied in this paper. Its kernel function is defined as follows:

$$K(T_1, T_2) = \langle \Phi(T_1), \Phi(T_2) \rangle = \sum_i (\phi_i(T_1) \cdot \phi_i(T_2))$$

Based on these, for a syntactic structure *T* in the test set, we can compute the similarity between *T* and all the standard syntactic structures by the above kernel function. A similarity threshold th_{sim}[4] is set to determine whether the string from *T* is a correct product attribute candidate.

3 Experiments

3.1 Datasets and Evaluation Metrics

Three domain-specific datasets are used in the experiments, which is from an official Chinese Opinion Analysis Evaluation 2008 (COAE2008) (Zhao et al., 2008). Table 2 shows the statistics of the three datasets, each of which is divided into training, development and test data in a proportion of 2:1:1.

Domain	# of sentences	# of standard product attributes
Camera	1,780	1,894
Car	2,166	2,504
Phone	2,196	2,293

Table 2: The datasets for three product domains.

Two evaluation metrics, recall and noise ratio, are designed to evaluate the performance of the product attribute candidate extraction. Recall refers to the proportion of correctly identified attribute candidates in all standard product attributes. Noise ratio refers to the proportion of incorrectly identified attribute candidates in all candidates.

3.2 Comparative methods

We choose the method, which considers NPs as the product attribute candidates, as the baseline (shown as **NPs_based**).

Besides, in order to assess the two generalization strategies' effectiveness, four experiments are designed as follows:

[4]In the experiments, th_{sim} is set to 0.7, which is tuned on the development set.

SynStru_based: It refers to the syntactic structure exact matching method, which is implemented without the two proposed generation strategies.

SynStru_*h*: It refers to the strategy only using the first generalization.

SynStru_*kernel*: It refers to the strategy only using the second generalization.

SynStru_*h*+*kernel*: It refers to the strategy using both two generalizations, i.e., it refers to our approach in this paper.

3.3 Results

Table 3 lists the comparative performances on the test data between our approach and the comparative methods for product attribute candidate extraction.

Domain	Method	Recall	Noise ratio
Camera	NPs_based	81.20%	63.64%
	SynStru_based	84.80%	67.67%
	SynStru_*h*	92.08%	74.74%
	SynStru_*kernel*	92.51%	75.92%
	SynStru_*h*+*kernel*	92.72%	76.25%
Car	NPs_based	85.25%	69.35%
	SynStru_based	86.31%	72.66%
	SynStru_*h*	93.78%	78.01%
	SynStru_*kernel*	94.56%	79.50%
	SynStru_*h*+*kernel*	94.71%	80.44%
Phone	NPs_based	84.11%	63.76%
	SynStru_based	86.26%	67.09%
	SynStru_*h*	93.13%	73.62%
	SynStru_*kernel*	93.47%	75.11%
	SynStru_*h*+*kernel*	93.63%	75.35%

Table 3: Comparisons between our approach and the comparative methods for product attribute candidate extraction.

Analyzing the recalls in Table 3, we can find that:

1. The performance of SynStru_based method is better than NPs_based method for each domain. This can illustrate that syntactic structures can cover more forms of the product attributes. However, the recall of SynStru_based method is not high, either.

2. The two generalization strategies, SynStru_*h* and SynStru_*kernel* can both significantly improve the performance for each domain, comparing to the SynStru_based method. This can illustrate that our two generalization strategies are helpful.

3. Our approach SynStru_*h*+*kernel* achieves the best performance. This can illustrate that the two generalization strategies are complementary to each

other. And further, mining and generalizing the syntactic structures is effective for candidate extraction.

However, the noise ratio for each domain is increasing when employing our approach. That's because, more kinds of syntactic structures are considered, more noise is added. However, we can easily remove the noise in the candidate classification step. Thus in the next section, we will assess our candidate extraction approach by applying it to the product attribute extraction task.

4 Application in Product Attribute Extraction

For the extracted product attribute candidates, we train a maximum entropy (ME) based binary classifier to find the correct product attributes. Several commonly used features are listed in Table 4.

Feature	Description
lexical	the words of the product attribute(PA)
	the POS for each word of the PA
	three words before the PA
	three words after the PA
	the words' number of the PA
syntactic	the syntactic structure of the PA
binary (Y/N)	Is there a stop word in the PA?
	Is there a polarity word in the PA?
	Is there an English word or number in the PA?

Table 4: The feature set for product attribute extraction.

Table 5 shows the product attribute extraction performances on the test data. We can find that the performance (F1) of our approach is better than NPs_based method for each domain. We discuss the results as follows:

1. Comparing to the NPs_based method, the recall of our approach increases a lot for each domain. This demonstrates that generalized syntactic structures can cover more forms of product attributes.

2. Comparing to the NPs_based method, the precision of our approach also increases for each domain. That's because syntactic structures are more specialized than the phrase forms (such as NP, VP) in the previous work, which can filter some noises from the phrase(NP) candidates.

5 Conclusion

This paper describes a simple but effective way to extract the product attribute candidates from product reviews. The proposed approach is based on deep analysis into syntactic structures of the product attributes, via intuitive heuristics and syntactic structure decomposition. Experimental results indicate that our approach is promising. In future, we will try more syntactic structure generalization strategies.

Domain	Method	R (%)	P (%)	F1 (%)
Camera	NPs_based	59.62	68.38	63.70
	Our approach	62.96	73.32	67.74
Car	NPs_based	59.94	64.87	62.31
	Our approach	67.34	65.90	66.61
Phone	NPs_based	58.53	71.14	64.22
	Our approach	67.84	76.13	71.74

Table 5: Comparisons between our approach and the NPs_based method for product attribute extraction.

Acknowledgments

This work was supported by National Natural Science Foundation of China (NSFC) via grant 60803093, 60975055, and the "863" National High- Tech Research and Development of China via grant 2008AA01Z144.

References

Wanxiang Che, Min Zhang, AiTi Aw, Chew Lim Tan, Ting Liu, and Sheng Li. 2008. Using a hybrid convolution tree kernel for semantic role labeling. *ACM Trans. Asian Lang. Inf. Process.*, 7(4).

Michael Collins and Nigel Duffy. 2001. Convolution kernels for natural language. In *NIPS*, pages 625–632.

Minqing Hu and Bing Liu. 2004. Mining opinion features in customer reviews. In *Proceedings of AAAI-2004*, pages 755–760.

Ana-Maria Popescu and Oren Etzioni. 2005. Extracting product features and opinions from reviews. In *hltemnlp2005*, pages 339–346.

Jeonghee Yi, Tetsuya Nasukawa, Razvan Bunescu, and Wayne Niblack. 2003. Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques. In *Proceedings of the IEEE International Conference on Data Mining*.

Jun Zhao, Hongbo Xu, Xuanjing Huang, Songbo Tan, Kang Liu, and Qi Zhang. 2008. Overview of chinese opinion analysis evaluation 2008.