International Journal of Service and Knowledge Management
International Institute of Applied Informatics
2023, Vol. 7, No. 2, IJSKM807

Semi-Automatic Category Estimation and Data Augmentation for Opinion Extraction of Product Components

Shogo Anda [*] , Masato Kikuchi [*], Tadachika Ozono [*]

Abstract

When customers purchase a product online, they use reviews to gather information about that product to help them make a purchase decision. Aspect-based Sentiment Analysis is a task that analyzes the review content from various perspectives, including the product itself, its components, and its retail outlets. We focus on comparing the characteristics of each component in a product with those of other products at the time of purchase. We define a task called component-based sentiment analysis (CBSA), which analyzes the review content from the perspective of only each component in the product. The CBSA task consists of opinion target extraction and polarity analysis. We approach that task with a classifier. We describe a semi-automatic category determination method for creating classification labels for CBSA and a data augmentation method to improve its classification performance. In experiments, we show that our category determination method can generate categories that cover 95% of the existing categories on e-commerce sites and that our data augmentation method improves the *macro-*$F1$*-measure* for uncommon opinions by 10%.

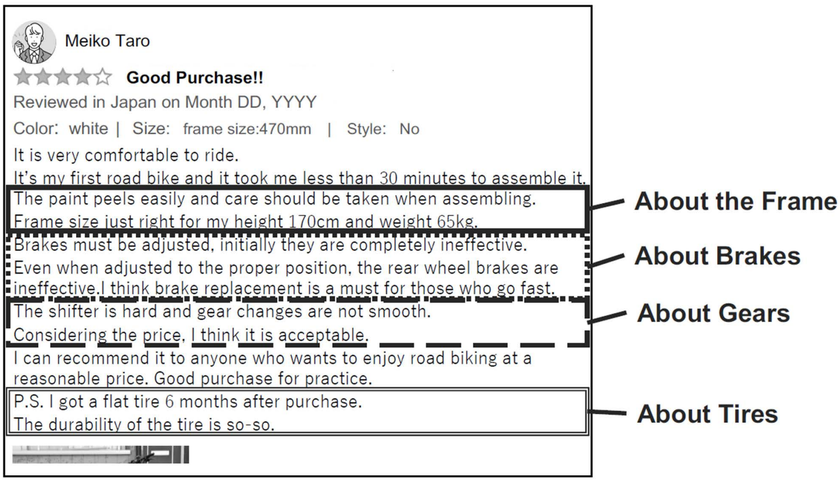
Keywords: information extraction, data augmentation, aspect-based sentiment analysis.

1 Introduction

Consumers usually get information about a product from product reviews on e-commerce sites when they are considering purchasing a product online. However, manually collecting information from an enormous number of reviews requires time and effort. Figure 1 shows an example of a bicycle review. This review describes the size of the frame, the usability of the brakes and gears, and the durability of the tires. Gathering information about a product's durability of tires or the design of its frame from a product page where many reviews have been posted requires time and effort.

We aim to assist in the collection of component-by-component information on products from product reviews. We use classifiers based on supervised learning to analyze the components mentioned and their attributes from the text contained in the reviews and to extract comments by component and by aspect from the reviews. The task of analyzing the objects mentioned in a sentence and their polarity is called aspect-based sentiment analysis (ABSA) and has been widely studied [1][2]. In this study, we focus on components

[*] Nagoya Institute of Technology, Aichi, Japan



Meiko Taro

Good Purchase!!

Reviewed in Japan on Month DD, YYYY

Color: white | Size: frame size:470mm | Style: No

It is very comfortable to ride.
It's my first road bike and it took me less than 30 minutes to assemble it.
The paint peels easily and care should be taken when assembling.
Frame size just right for my height 170cm and weight 65kg.
Brakes must be adjusted, initially they are completely ineffective.
Even when adjusted to the proper position, the rear wheel brakes are ineffective.I think brake replacement is a must for those who go fast.
The shifter is hard and gear changes are not smooth.
Considering the price, I think it is acceptable.
I can recommend it to anyone who wants to enjoy road biking at a reasonable price. Good purchase for practice.
P.S. I got a flat tire 6 months after purchase.
The durability of the tire is so-so.

Figure 1: A review sample of the bicycle

of products and their tributes as the target of ABSA called component-based sentiment analysis (CBSA).

We have been developing a text classifier for CBSA. There are two tasks to achieve highly accurate text classification for CBSA. The first is that the required classification labels are unknown in advance. We cannot know in advance which components in a product are mentioned in a review of a specific genre and with what aspect, and therefore cannot define the components and aspects to be the classification labels when creating the classifier. Some product genres have component categories on the e-commerce site that are equal to the classification labels, but not all genres have them. We must create classification labels that seem to consider the content of the entire review of the specific genre. Second, Classifying low-frequency comments is difficult. The number of mentions per component or attribute in the review is imbalanced. Thus, the recall rate for low-frequency comments is reduced when they are used as training data. Classifiers for CBSA need to improve classification performance for low-frequency data.

We propose two methods to create a BERT-based classifier model for opinion extraction of CBSA. One is an extraction method based on pattern matching for classification label candidates, and the other is a data augmentation method using WordNet [3]. In the evaluation experiments, we conducted three experiments: comparing the labels created using our method with existing categories on the e-commerce site, evaluating data augmentation using WordNet, and evaluating the performance of the classifier using the training data we had created. As a result, 95% of the existing labels on the e-commerce site were reproduced. We also show that data augmentation using WordNet increased classification performance for low-frequency comments by 10%.

The rest of this paper is organized as follows. Section 2 presents related work, and Section 3 describes text classification for CBSA. Section 4 proposes our methods. Section 5 describes the setup and results of three experiments to evaluate our method. Section 6 discusses our methods based on the experimental results, and Section 7 concludes our remarks.

2 Related Work

This section describes related work on ABSA, BERT, and WordNet. We are developing a CBSA system that contains a BERT-based text classifier with data augmentation using WordNet [3]. CBSA is ABSA for analyzing opinions of product components.

ABSA is a task that analyzes the polarity of a sentence evaluated from a particular aspect. ABSA is defined as a task in which the opinions under review have an aspect category consisting of specific entities and attributes, and the sentiment toward that category is analyzed [1]. For example, in the laptop PC domain, entities include LAPTOP, DISPLAY, KEYBOARD, SUPPORT, COMPANY, etc. Similarly, attributes include GENERAL, PRICE, QUALITY, etc. ABSA selects an appropriate aspect category from a predefined set of entities and attributes for each domain and analyzes the polarity for that aspect category. The text classification by component and attribute in this study is similar to the entity and attribute in Pontiki et al. However, since we cannot know in advance which components and aspects are mentioned in the review, the classification of components and attributes will be performed after those analyzes are made based on the entire review. In addition, the target of the analysis is limited to the components of the products within the entity. The entity in ABSA includes the components in the product, which are predefined and not easy to set up. We aim to target classification by component according to the specialty of the content of the review for each product type. For example, among bicycles, we believe that the level of detail of references to wheels and tires, rims, etc. will differ between city bikes and road bikes for skilled riders.

Liu et al. also defined ABSA as a task that analyzes the sentiment of an opinion target, where the opinion under review has an opinion target consisting of a specific entity and aspect [2]. Entities here refer to the product name and the services, companies, and individuals involved in the product purchase process, and aspects refer to the components of the product and the attributes of each entity. Liu et al. approached ABSA by extracting opinion targets from the sentences and classifying their polarity. In our study, we attempt to analyze the review content from the perspective of components and their attributes, unlike the entity and aspect definitions of Liu et al.

Automatic aspect category determination is still a challenging issue on ABSA. In ABSA, the text is assigned entity and attribute pairs from a default category based on its content. Category determination is the field that defines its entity and attribute categories. These studies detect or extract entities and attributes based on the review set [4][5]. Some studies focus on extracting opinion target expression (OTE) from the review set for their realization [6]. An OTE is a term referring to an entity or attribute in a sentence. Other methods of aspect category detection or extraction include the use of related words for each product genre using knowledge databases such as WordNet and ConceptNet [7][8], and studies that use abstract words of words in the collected review set [9][10]. There are other methods for analyzing review content, other than determining aspect categories and opinion targets, using clustering and topic modeling. In the method using clustering, the review sentences are clustered by distributed representation, and the clusters are named by a generative language model based on the clustered review sentences to analyze the review content [11]. The method using topic modeling identifies aspect and attribute words in the domain without requiring labeled data [12]. We focused on the pattern of opinion occurrence in the review, extracted OTEs, and manually created labels based on these OTEs in an attempt to create appropriate component and their attribute classification labels.

We use a classifier based on BERT, an encoder model that learns sentences from both the beginning and the end of the sentence using a model called Transformer. Due to its supposed ability to consider context and whole sentences, it has also been applied to review text classification tasks [13]. There are also models for various languages as open-source, and classification studies of reviews written in many languages [14][15]. We build a classification model based on BERT that extracts components and attributes in sentences containing opinions on product components. In addition, this paper uses the model published at Huggingface that is available in Japanese[1].

We use WordNet, a lexical database. All registered words belong to one or more synonyms collected for each meaning. Multiple synonyms form a hypo-hyper relationship with each other. By inputting a word, users can obtain the synonym to which the word belongs, as well as the words belonging to the synonym, the hyponym, and the hypernym. Using this database, we can obtain similar terms for words, higher-level terms, and lower-level terms. For example, if the input word is "bicycle", this word belongs to two synonyms: one as a noun and the other as a verb. The noun synonym is explained as "a wheeled vehicle that has two wheels and is moved by foot pedals" and the verb synonym is explained as "ride a bicycle." Synonyms are bike, wheel, and cycle. Data augmentation by word replacement using WordNet is known to be a simple and useful method [16]. In this study, based on words in a certain sentence, similar words are obtained and replaced using WordNet to create similar sentences and augment the training data.

This paper shows our system and its further evaluation to clarify the effectiveness of our data augmentation with automatic data quality checking. Unfortunately, our previous article has some limitations in the experiments [17]. Therefore, we report new experimental evaluations of our automatic data quality-checking method. Our experimental results demonstrate the necessity of the method to improve classification performance.

3 Component-based Sentiment Analysis

This section shows our opinion extraction of product components for CBSA and a text classification for the task. Then, we discuss two challenges in the text classification task; classification label creation and imbalanced data.

CBSA is a particular ABSA specializing in opinion analysis for product components. CBSA consists of opinion target extraction and polarity analysis. The opinion target extraction is a task to extract categories of product components and categories of their attributes from review comments. For example, "The tire went flat right away" is an opinion-target expression in a review comment. In this example, "Tire" is a product component category of bicycles, and "Durability" is an attribute category for the "Tire" category. This task determines from its text content that the text is an opinion of the "Tire" and "Durability" categories, and sentiment analysis is a task that analyzes whether the opinion is positive or negative about the durability of tires. Other categories of road bike components could include frames and gears, while attribute categories could include functionality and size. CBSA is a task that analyzes the polarity of review comments in view of their component category and attribute category pairs. This paper focuses on opinion target extraction that contains two issues denoted below.

First, we cannot know what components and attributes should be used as classification labels for each target genre when creating classifiers and training data in advance. When creating a classifier, we need to define the classification labels to be used for its target. The

[1] https://huggingface.co/cl-tohoku/bert-base-japanese

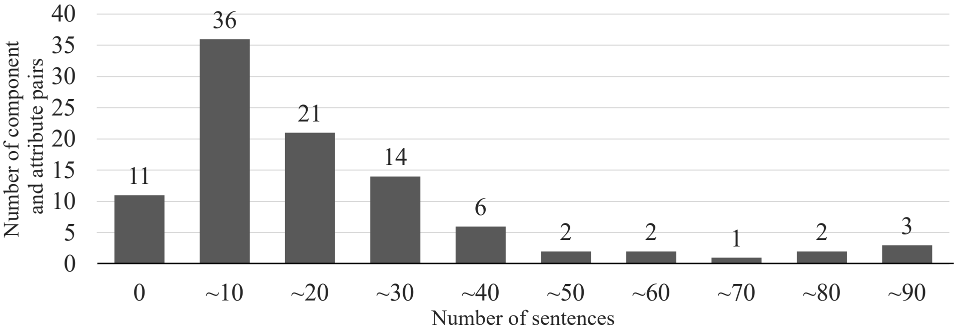


Figure 2: Histogram of the number of sentences in pairs of components and attributes

classification labels for CBSA should define the components and their attributes that are mentioned throughout the review in the target product genre. However, we do not know in advance what components will be mentioned within the review and in what attributes. For example, if a classification of bicycle reviews is attempted, it is easy to consider tires, frames, and other components, as well as attributes such as durability and appearance, as targets for classification, but we cannot know in advance whether there will be references to baskets and valves. In addition, when referring to tires, depending on the type of bicycle and the reviewer's understanding of bicycles, it may vary whether the comment should be considered for the wheel as a whole or subdivided into rims and spokes. Therefore, for text classification for CBSA, Therefore, for CBSA, we must determine which components and aspects should be subject to classification at what level of detail based on the content of the review of a product genre.

Second, training data created from the text under review are imbalanced, resulting in poor classification performance for minor comments. Training classifiers using training data created with component and attribute labels degrade classification performance for low-frequency labels [18]. In the bicycle example, the classification performance of frequently mentioned comments such as tires and frames is relatively high, but the classification performance of infrequently mentioned comments such as cranks and valves decreases. Table 1 shows a breakdown of the component and attribute labels assigned to the 1,000 road bike review sentences on amazon.co.jp. The intersection of each label is called the pair of the corresponding component and attribute. For example, 70 sentences belong to the "Tire" and "Durability" pair. The table shows a large difference in the number of sentences for each pair of component and attribute labels. Figure 2 shows a histogram of the number of sentences in each pair. The horizontal axis is the number of sentences belonging to a pair of components and attributes, and the vertical axis is the number of these pairs. The figure shows that the most common number of sentences is 10 or fewer, whereas some pairs have more than 70 sentences. This imbalance affects classifier training and reduces recall rates for fewer pairs in the training data.

In classification for CBSA, a lower recall rate for infrequent comments means being unable to get comments that are difficult for humans to detect. Therefore, to improve the accuracy of the classification for CBSA, it must improve classification performance for low-frequency components and attributes in the training data.

Table 1: Breakdown of 1,000 road bike review sentences

	Durability	Functionality	Preference	Installation	Weight	Size	Appearance	Total
Tire	70	5	22	53	3	23	22	170
Valve	8	0	19	2	0	2	2	33
Rim	15	1	10	19	0	5	15	51
Spoke	19	0	0	5	0	0	2	21
Handle	33	11	20	73	8	11	14	137
Brake	27	21	30	85	4	1	8	147
Bell	9	1	11	2	0	0	6	25
Gear	24	37	86	41	6	2	3	163
Pedal	38	5	23	68	7	2	10	117
Crank	25	4	7	20	1	1	0	42
Chain	38	12	7	35	1	3	13	90
Light	23	1	17	15	0	1	9	82
Saddle	17	6	85	45	3	13	13	151
Frame	22	6	24	23	18	14	55	130
Total	283	82	297	322	45	69	145	

4 Category Determination and Data Augmentation

We propose two methods and a classifier model to realize a text classifier for CBSA with high classification performance. One is a category determination method using pattern matching to create classification labels based on review content, the other is a data augmentation method using WordNet to generate similar sentences, and a classifier based on BERT.

4.1 Category Determination

In opinion target extraction in CBSA, the text is assigned components and their attributes according to its content. Therefore, a classifier that assigns components and attributes is necessary. However, we cannot know in advance what components and attributes are mentioned for a given product type. Therefore, we need to define categories, like ABSA. The category determination for CBSA supports the creation of classification labels for the opinion target extraction by determining the component and attribute categories that are mentioned in the product type concerned.

We propose a semi-automatic category determination method to create classification labels that consider the components and attributes mentioned throughout reviews within specific product types. Our method collects reviews of specific product types and creates component and attribute labels from them. Thus, we can create labels for products that do not have existing component and attribute categories on the e-commerce site. This method focuses on patterns that appear in the review, automatically extracts candidate component labels and candidate attribute labels, and then manually create component labels and attribute labels from these candidates.

This method consists of four steps. As step 1, we collect product reviews for specific

Table 2: Examples of candidate component labels extracted from the road bike reviews

Component names	Number of occurrences
assembly (improper)	250
bicycle (improper)	184
brake	153
tire	145
bike	84
saddle	75
pedal	75
price (improper)	69
bicycle shop (improper)	55
handle	54

product types. In step 2, we define the patterns to be extracted according to the sentence structure of the target language. From Japanese review comments, Kobayashi et al. [19] extracted opinions using a pattern “A の B が C” (“B of A is C” in English). Since this complete pattern does not occur frequently in product review comments, we divide this pattern into two patterns, “A の B” (“B of A”) and “B が C” (“B is C”), to obtain more review comments. The first pattern “A の B” extracts a component label “A” and an aspect label “B”. The second pattern “B が C" extracts a component label “B” and an aspect label “C”. These extracted labels are candidates of component and aspect labels. In step 3, we extract candidate component labels and candidate attribute labels from the reviews collected in step 1 using the patterns determined in step 2. In step 4, we manually create component labels and attribute labels from the label candidates we have extracted. These four steps extract label candidates from the review and create labels.

Tables 2, 3, and 4 show examples of component and attribute labels in the road bike domain created using this method. We used 30,000 review sentences extracted from product pages in the Sports/Outdoor/Cycling/Bikes/Road Bikes category within amazon[2]. In this example, we used MeCab, a Japanese morphological analyzer, and termextract, a Python module, for pattern extraction. Tables 2 and 3 show some of the candidate component labels and candidate attribute labels. Candidate component labels include those that do not indicate components, such as “assembly” and “bicycle shop,” while candidate attribute labels include synonyms and a variety of expressions. Thus, they are manually deleted and merged to create the component and attribute labels in Table 4. In this case, in the bicycle domain, 14 types of component labels are created such as tire, valve, etc., and 7 types of attribute labels are created such as durability, functionality, etc. This procedure is used to collect existing reviews on the e-commerce site and create component and attribute labels for specific product types.

4.2 Data Augmentation

We propose a data augmentation method using similar sentence generation for high-performance classification and a classifier model based on BERT. In this method, we

[2] https://www.amazon.co.jp

Table 3: Examples of candidate attribute labels co-occurring with the created component labels

Component labels	Attribute words
Brake	効く (effect)，弱い (weak)，甘い (weak)
Tire	パンク (got flat)，細い (narrow)，歪む (distorted)
Saddle	硬い (hard)，堅い (hard)
Pedal	外れる (loose)，ぐらつく (unsteady)，重い (heavy)
Handle	曲がる (misalignment)

Table 4: Component labels and attribute labels in the road bike domain

	Labels
Component labels (14 labels)	Tire, Valve, Rim, Spoke, Handle, Brake, Bell, Gear, Pedal, Crank, Chain, Light, Saddle, Frame
Attribute labels (7 labels)	Durability, Functionality, Preference, Installation, Weight, Size, Appearance

generate similar sentences of low-frequency comments in the training data and add them to the training data to improve the classification performance of minor comments.

Figure 3 shows a flow diagram of the data augmentation method. The upper left square in the figure refers to any pair in Table 1. We describe the procedure according to the flow. Initially, we divide the training data by component and by attribute, transforming it as shown in Table 1. Next, we set the minimum pair size (MPS) of the data augmentation. Similar sentences are generated and added to pairs per component and per attribute that do not meet the MPS up to the value of the MPS. After the MPS is set, similar sentences are generated and added to the training data.

We explain the procedure for generating similar sentences. A single sentence s is extracted from the pair p and a similar sentence s' of that sentence is generated. First, one word w is randomly selected from the s. In this case, all words are considered, with no restrictions on part of speech, etc. Next, synonyms of the selected word w are obtained from WordNet. In this method, we used words that belong to the same synonyms in WordNet. Some words may belong to more than one synonym. In such cases, one of the obtained synonyms is randomly selected (this word is called w'). The word w in the input sentence and the created w' are replaced to create sentence s'. The created s' is then quality-checked.

In the quality check, the expression around the replaced word w' is checked whether the expression is used in the review of that product type. The replaced word w' in s' and the words before and after it were used. Those three consecutive words were denoted as t. Quality checking is performed by checking whether t appears in the pre-collected set of unlabeled review sentences. If sentence s' passes the quality check, s' is added to pair p. If the number of sentences in p is less than MPS even after generating similar sentences from all sentences in pair p, t is changed to w' and the before or after a word or $t = w'$ during the quality check, and the similar sentences are generated and added again. When adding to the training data, if a sentence that already exists in the training data is generated, a similar sentence is generated again. This process is performed for all components and attribute pairs

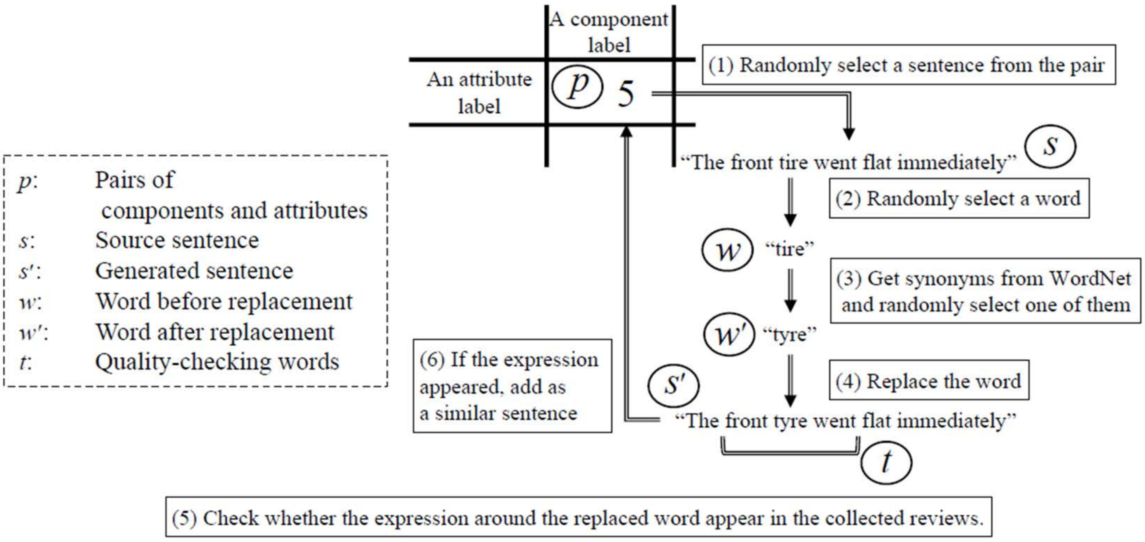

A component label
An attribute label
p
5
(1) Randomly select a sentence from the pair
"The front tire went flat immediately"
s
(2) Randomly select a word
w
"tire"
(3) Get synonyms from WordNet and randomly select one of them
w'
"tyre"
(4) Replace the word
s'
"The front tyre went flat immediately"
t
(6) If the expression appeared, add as a similar sentence
(5) Check whether the expression around the replaced word appear in the collected reviews.
p: Pairs of components and attributes
s: Source sentence
s': Generated sentence
w: Word before replacement
w': Word after replacement
t: Quality-checking words

Figure 3: Data augmentation using WordNet and quality-checking

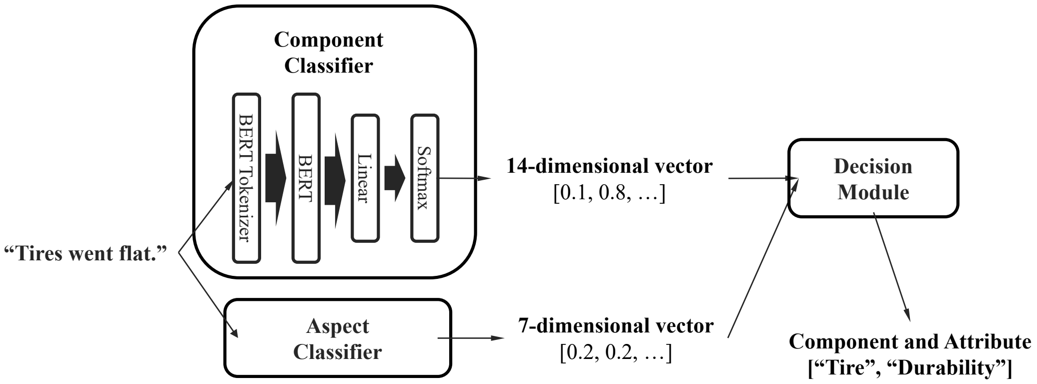


Figure 4: Two classifiers for determining categories of product components and their attributes from product reviews

in the training data and augments the data so that all sentences belonging to all components and attribute pairs in the training data have an MPS or higher.

As another approach to achieve high performance in text classification for CBSA, we propose a classifier model consisting of two classifiers based on BERT. The model takes a sentence as input and classifies predefined components and attributes for that sentence.

This model consists of a component classifier and an attribute classifier. The model structure is shown in Figure 4. This figure shows an example of a road bike review text as input. Sentences received as input are passed to the component and attribute classifiers. The component and attribute classifiers output a vector of dimensions for each classification label according to the input sentence. In the example in the figure, there are 14 and 7 types of component labels and attribute labels in road bikes, respectively, so the dimension of the vector of outputs of each classifier matches them. The vectors from each classifier are passed to the decision module, which determines the component and attribute labels to classify the input sentences based on these vectors. In this case, each vector is classified with a certain threshold value.

Each classifier consists of four layers. BERTTokenizer tokenizes the input sentence and enters it into BERT. At that time, the [CLS] token is added at the beginning of the sentence to realize BERT's unique classification that considers the entire sentence. When using BERT for classification tasks, the output based on this [CLS] token is forwarded to the next layer. Linear and Softmax compress and normalize the dimensions of BERT's output to output a vector with the same number of dimensions as the number of labels in each. This classifier model and the training data created using the methods described in subsection 4.1 are used to achieve text classification for CBSA.

5 Evaluation

We evaluated the two proposed methods and the proposed model in three experiments. Experiment A evaluated the category determination method, and Experiments B and C evaluated the data augmentation method and the classifier model.

5.1 Experiment A: Category Determination

We evaluated the usefulness of the labels created using our category determination method. We created component and attribute labels using our method and calculated the ratio of a match between these labels and the existing categories on the e-commerce site. We compared the component labels to the component categories on amazon.co.jp, and the attribute labels to the evaluation categories on kakaku.com.

We collected 30,000 product review sentences from Amazon in the road bike genre and created component and attribute labels using our label creation method. We calculated the percentage match between the two types of labels we created and the existing component and attribute categories on the e-commerce site, respectively. We calculated the percentage match between the component labels with the bicycle parts category on Amazon and the attribute labels with the bicycle evaluation category on Kakaku.com, since there is no static category on Amazon to compare with the attribute labels.

We used the string match and the semantic match for the match decision. The string match refers to a string match between the label created and the category on the e-commerce site. Attribute labels are created by manually merging labels based on label candidates. Therefore, unlike component labels, the string match is not applied because each string match itself has no meaning in attribute labels. The semantic match refers to the match between the created label and category when considering meaning. For example, in the component labels, chains and chainrings, brakes and brake wires, etc. are semantic matches. For attribute labels, design, appearance, weight, lightness, etc. are semantic matches. We calculated the percentage of semantic and sentiment matches in the categories on the e-commerce site for both component and attribute labels.

Figure 5 shows the results of this experiment. For component labels, our method created labels that matched 95% of the categories on amazon, with the string match at 45% and the semantic match at 50%. The only category that did not match the component label was "Stand." For attribute labels, the semantic match was 71%, creating a label that matched all but the "Driving Performance" and "Parts" categories on Kakaku.com. Of the categories that did not match, "Driving Performance" is an evaluation of the overall driving performance of the bicycle, and "Parts" is an evaluation of the parts on the bicycle.

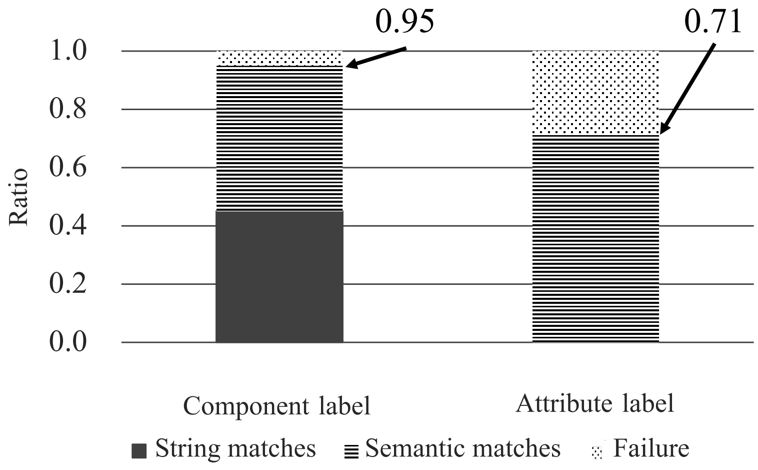


Figure 5: Ratio of categories on the e-commerce site to labels by our method

Table 5: Dataset Details

Type	Size	Component labels	Attribute labels	Pairs
Road bike	500	14	7	50
Laptop PC	500	10	7	47
Tent	500	10	6	38

5.2 Experiment B: Quality Checking and Data Augmentation

We evaluated the quality checks in our data augmentation method. We compared classification performance on training data with and without quality checks during data augmentation.

We collected 30,000 review sentences from amazon.co.jp about road bikes, laptop computers, and tents, and manually labeled 500 randomly selected sentences. Details of the datasets used in Experiments B and C are shown in Table 5. In this experiment, we used the road bike dataset from Table 5. Among the pairs of each component label and attribute label in the dataset, we randomly collected one sentence from each pair of 5 or more sentences and used them as evaluation data. Then, randomly 50 sentences were collected from the whole set and used as validation data, and the remaining 400 sentences were used as training data.

We compared four cases: *no aug*, *ours*, *no check*, and $no\ filter$. The *no aug* is no data augmentation, *ours* is augmentation with our method, and the *no check* is our method without quality checking, where the generated sentences are added up to the MPS. The $no\ filter$ is based on our method, in which the generated sentences that did not pass the check were added to the training data. The MPS for data augmentation was set to 15. We trained and calculated the classification performance using these four cases. The threshold in the decision module was adjusted to have the largest *macro-*$F1$*-measure* using the validation data. Every case was trained for 20 epochs, and the model with the least loss with the validation data was chosen.

We used *macro-*$F1$*-measure* for the classification performance. The $F1$*-measure* was

Table 6: Training data size after data augmentation for the four cases

	no aug	*no check*	*no filter*	*ours*
size	400	667	3838	667

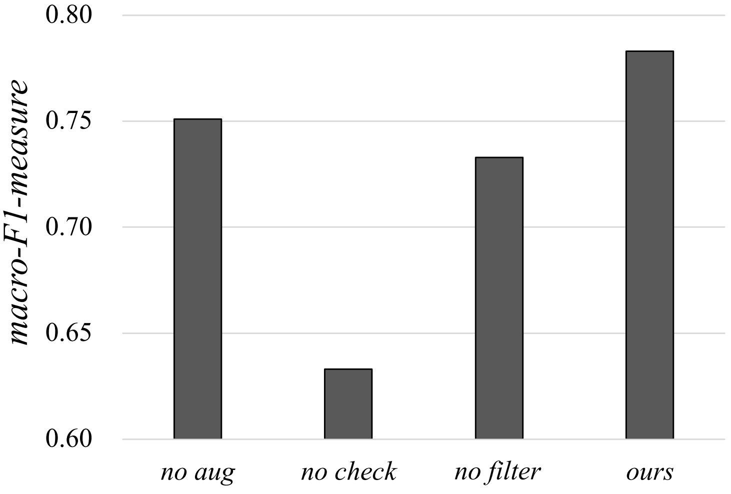


Figure 6: The *macro-F*1*-measure*s for four cases

calculated by

$$F1\text{-}measure_x = 2 \cdot \frac{Precision_x \cdot Recall_x}{Precision_x + Recall_x}.$$

The $Precision_x$ and the $Recall_x$ were calculated by the number of sentences with label x as the predicted label and with label x as the correct label. The *macro-F*1*-measure* was the average of the $F1$*-measure* of all labels and is calculated by

$$macro\text{-}F1\text{-}measure = \frac{1}{|Labels|} \sum_{l \in Labels} F1\text{-}measure_l.$$

The *Labels* denoted the set of labels and $|Labels|$ is the number of labels.

The training data size after augmentation for the four cases is shown in Table 6. For *no check* and *ours*, sentences were added up to MPS for each pair, so the data size after augmentation is 667. In $no\ filter$, all sentences generated during augmentation in ours were added, so the data size is 3838. The results of calculating the classification performance for each of the four cases are shown in Figure 6. Among the three augmentation methods, *ours* calculated the highest classification performance, and *no check* and $no\ filter$ show a lower classification performance than before the augmentation. The classification performance of *no check* is significantly lower than the others.

5.3 Experiment C: MPS and Data Augmentation

We evaluated the relationship between the number of similar sentences added by our similar sentence generation-based data augmentation method and classification performance. We

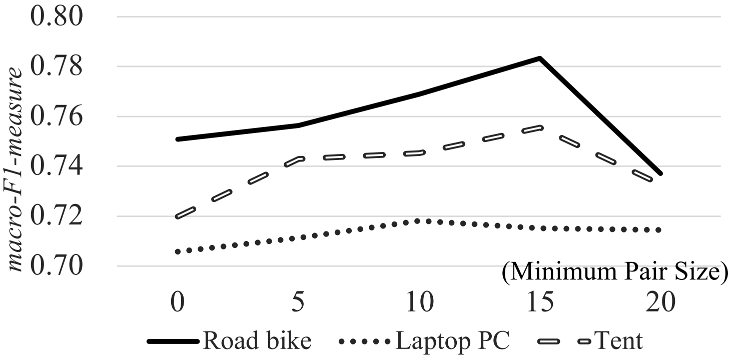


Figure 7: Effect of MPS during our data augmentation on classification performance

varied the MPS defined in our data augmentation method and observed the classification performance. The MPS is the minimum pair size set during our data augmentation. All pairs of components and attributes in the training data are added with similar sentences to have sentences above the MPS.

We used the dataset described in Experiment B (Table 5). The datasets were conducted in three domains: road bike, laptop PC, and tent. We collected one or two sentences from each pair in each dataset with five or more sentences to make a total of 50 sentences, which were used as evaluation data. This was to reduce the imbalance of each component and attribute pair within the evaluation data. Then, we collected 50 sentences randomly and used them as validation data and the rest as training data. For each training data, we defined several MPSs and applied data augmentation using our method. The MPS was set to 0,5,...,20 based on the size of the training data and the size of the pairs in the training data. We applied three types of data augmentation to the training data of each MPS and calculated the classification performance of each model. All cases were trained for 20 epochs and the threshold in the decision module was set to have the highest *macro-*$F1$*-measure* using the validation data. The model with the least loss in the validation data among the 20 models was adopted and the classification performance was calculated using the evaluation data. Classification performance was calculated by the same $F1$*-measure* and *macro-*$F1$*-measure* as in Experiment B.

Figure 7 shows the classification performance of the model with data augmentation using the road bike, laptop PC, and tent MPSs. For road bikes, classification performance increased from 0.75 to over 0.78 before data augmentation for the MPS=15 model. For tents, it also increased from 0.72 to over 0.75. However, for road bikes, there is no noticeable increase. Figure 8 shows the $F1$*-measure*s for each label for the road bikes that performed the highest classification performance. The figure shows that the classification performance is on average better for the component labels than for the attribute labels. The figure also shows an increase in classification performance for low-frequency data such as Spoke and Bell compared to the MPS=0 case. Figure 9 shows the classification performance calculated for the case of MPS=15 for road bikes, dividing pairs subject to data augmentation (i.e., pairs of components and attributes that contain only less than MPS in the training data) as minor comments and others as major comments. The figure shows that the classification performance for major comments only decreased by 0.02, while it increased by 0.10 for minor comments.

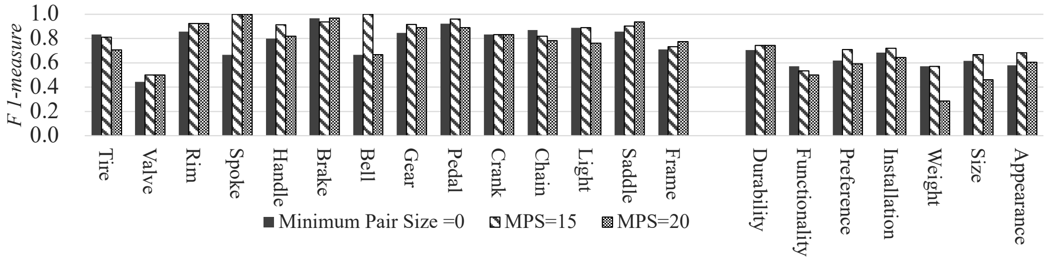


Figure 8: $F1$-*measure*s for each component label and attribute label on road bikes

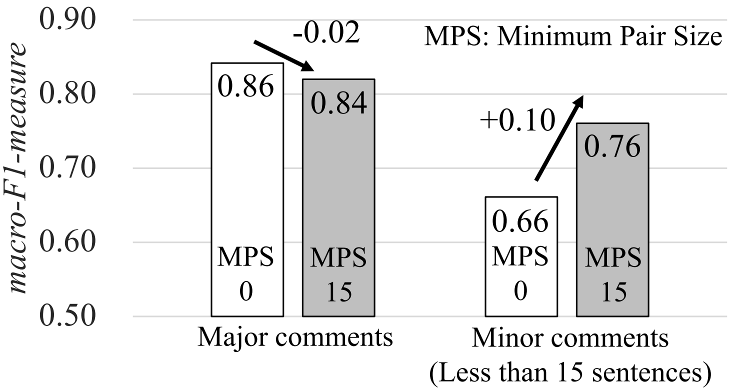


Figure 9: Changes in *macro-*$F1$-*measure* for minor comments and major comments

6 Discussion

In this section, we discuss the usefulness of our proposed method from three evaluation experiments and discuss the possibility of extending our method to multiple languages.

The results of Experiment A show a mismatch between the "Stand" category in the component labels and the "Driving performance" and "Part" categories in the attribute labels. Among the evaluation categories, "Driving performance" is not considered important when collecting component-by-component evaluations, which is the purpose of this study, because these are evaluations of the bicycle as a whole. In addition, "Parts" is an overall evaluation of each component on the bicycle, and therefore is not appropriate as an attribute of each component in this study. We show that our method is effective for the component and attribute categories on Amazon and Kakaku.com as the correct answers. In future work, we need to verify the effectiveness of our method for product types that do not have those categories on e-commerce sites.

The results of Experiment B show that a quality check within our data augmentation method is useful. When the quality check is not performed (*no check*), low-quality sentences are added to the low-frequency data, resulting in significant degradation of classification performance. The classification performance was also lower for $no\ filter$ than for $ours$. The $no\ filter$ has all the sentences included in $ours$, but the classification performance was negatively affected by the other low-quality sentences. This suggests that in this study, the quality of training data is more important than its quantity. In the present method, the quality of the generated sentences is considered to have been ensured by quality checking.

Experiment C shows that our method improved the classification performance in the road bike and tent domains. These experimental results suggest that this method should be improved for product domains with fine-grained aspect categories, such as laptop PCs. Also, the results of the classification performance calculated separately for minor and major comments show that the overall classification performance was improved by a significant improvement in the classification performance for minor comments. In this experiment, MPS=15 was optimal for the road bike and tent domains for about 500 training data. Thus, the optimal MPS for each domain can be derived experimentally.

We believe that our method can be applied to other languages other than Japanese, such as English and Chinese. Japanese text data were used in the examples and evaluation experiments. From them, we show that our method is effective for Japanese reviews. Among the three proposed methods, the data augmentation method and the classifier model rely on the language models of WordNet and BERT, respectively, and can be easily adapted to other languages where those models exist. In addition, the category determination method used patterns based on language structure to extract candidate labels. Therefore, by adapting the patterns to other languages, we expect that the proposed method as a whole can be adapted to other languages.

7 Conclusion

We explained CBSA for analyzing the review contents specifically for the components in a product. CBSA consists of opinion target extraction and polarity analysis, and a classifier is used for opinion target extraction. A category determination method to create classification labels for that classifier and a data augmentation method to improve the classification performance was described. In experiments, we showed that the labels created using our category determination method covered 95% of the categories on the e-commerce site, and that our data augmentation improved the *macro-$F1$-measure* for minor comments by 10%. In future work, further improvement is needed for product domains with fine-grained aspect categories, such as laptop PCs.

Acknowledgment

This work was supported in part by JSPS KAKENHI Grant Numbers JP19K12266, JP22K18006.

References

[1] M. Pontiki, D. Galanis, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, M. AL-Smadi, M. Al-Ayyoub, Y. Zhao, B. Qin, O. De Clercq, V. Hoste, M. Apidianaki, X. Tannier, N. Loukachevitch, E. Kotelnikov, N. Bel, S. M. Jiménez-Zafra, and G. Eryiğit, "SemEval-2016 Task 5: Aspect Based Sentiment Analysis," *SemEval 2016*, pp. 19–30, 2016.

[2] B. Liu, "Sentiment Analysis: Mining Opinions, Sentiments, and Emotions," *Studies in Natural Language Processing*, 2020.

[3] H. Isahara, F. Bond, K. Uchimoto, M. Utiyama, and K. Kanzaki, "Development of the Japanese WordNet," *In Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC 2008)*, pp. 2420–2423, 2008.

[4] S. Ramezani, R. Rahimi, and J. Allan, "Aspect Category Detection in Product Reviews using Contextual Representation," *In Proceedings of the 2020 ACM SIGIR Workshop on eCommerce (SIGIR eCom'20)*, 2020.

[5] P. Jeyanthi, R. Subhashini, and B. Bhamare, "Aspect Category Extraction for Sentiment Analysis using Multivariate Filter Method of Feature Selection," *International Journal of Recent Technology and Engineering*, vol. 8, 2021.

[6] Z. Chen and T. Qian, "Enhancing Aspect Term Extraction with Soft Prototypes," *In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*, pp. 2107–2117, 2020.

[7] R. Speer, J. Chin, and C. Havasi, "ConceptNet 5.5: An Open Multilingual Graph of General Knowledge," *In Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI 2017)*, pp. 4444–4451, 2017.

[8] Z. Zheng, Y. Cai, and L. Li, "Enhance Weakly-Supervised Aspect Detection with External Knowledge (Student Abstract)," *In Proceedings of the 36th AAAI Conference on Artificial Intelligence (AAAI 2022)*, vol. 36, no. 11, pp. 13 119–13 120, 2022.

[9] Y. Zeng, G. Wang, H. Ren, and Y. Cai, "Enhance Cross-Domain Aspect-Based Sentiment Analysis by Incorporating Commonsense Relational Structure (Student Abstract)," *In Proceedings of the 36th AAAI Conference on Artificial Intelligence (AAAI 2022)*, vol. 36, no. 11, pp. 13 105–13 106, 2022.

[10] J. Cao, R. Liu, H. Peng, L. Jiang, and X. Bai, "Aspect Is Not You Need: No-Aspect Differential Sentiment Framework for Aspect-Based Sentiment Analysis," *In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2022)*, pp. 1599–1609, 2022.

[11] P. Sircar, A. Chakrabarti, D. Gupta, and A. Majumdar, "Distantly Supervised Aspect Clustering and Naming for E-Commerce Reviews," *In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Track (NAACL-HLT 2022)*, pp. 94–102, 2022.

[12] C. Sindhu, D. Mukherjee, and Sonakshi, "A Joint Sentiment-Topic Model for Product Review Analysis of Electronic Goods," *In Proceedings of the 5th International Conference on Computing Methodologies and Communication (ICCMC 2021)*, pp. 574–578, 2021.

[13] M. Bilal and A. A. Almazroi, "Effectiveness of Fine-Tuned BERT Model in Classification of Helpful and Unhelpful Online Customer Reviews," *Electronic Commerce Research*, pp. 1–21, 2022.

[14] O. Montenegro, O. S. Pabón, and R. E. G. De Piñerez R., "A Deep Learning Approach for Negation Detection from Product Reviews Written in Spanish," *In Proceedings of the 47th Latin American Computing Conference (CLEI 2021)*, pp. 1–6, 2021.

[15] Y. Kawazoe, D. Shibata, E. Shinohara, E. Aramaki, and K. Ohe, “A Clinical Specific BERT Developed Using a Huge Japanese Clinical Text Corpus,” *Plos one*, vol. 16, no. 11, p. e0259763, 2021.

[16] J. Wei and K. Zou, “EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks,” *In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP 2019)*, pp. 6382–6388, 2019.

[17] S. Anda, M. Kikuchi, and T. Ozono, “Developing a Component Comment Extractor from Product Reviews on E-Commerce Sites,” *In Proceedings of the 12th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI 2022)*, pp. 83–88, 2022.

[18] V. López, A. Fernández, S. García, V. Palade, and F. Herrera, “An Insight into Classification with Imbalanced Data: Empirical Results and Current Trends on Using Data Intrinsic Characteristics,” *Information Sciences*, vol. 250, pp. 113–141, 2013.

[19] N. Kobayashi, K. Inui, and Y. Matsumoto, “Extracting Aspect-evaluation and Aspect-of Relations in Opinion Mining,” *In Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2007)*, pp. 1065–1074, 2007.