Effective construction and expansion of a sentiment corpus using an existing corpus and evaluative criteria estimation

Ryosuke Tadano **Kazutaka Shimada** **Tsutomu Endo**
Department of Artificial Intelligence, Kyushu Institute of Technology
{r_tadano, shimada, endo}@pluto.ai.kyutech.ac.jp

Abstract

In this paper, we propose a method for constructing and expanding a sentiment corpus effectively. We focus on association between a word and evaluative criteria. We use an existing corpus and analyze it to identify the association. We develop an annotation support tool for building a reliable corpus efficiently. The experiment result shows the effectiveness of the tool in terms of the annotation speed and agreement between annotators. Furthermore, for estimating the association of unknown words, we apply an existing domain estimation method to our task. The method estimates a domain of a word. We assume the domain to be an evaluative criterion in our corpus and estimate the evaluative criterion of a word. As a result, we verified that the method was effective in our task.

1 Introduction

As Web services such as Weblogs and BBSs have become widely used, people can easily post a review for products or services. Handling evaluative information (sentiment analysis) has become necessary. Building a sentiment corpus is an important task for the sentiment analysis.

Kaji and Kitsuregawa (2007) have proposed a method to automatically build a lexicon for sentiment analysis from the massive HTML documents. Takamura et al. (2005) have identified the polarity (p / n) of words with spin model. They are very effective because they are massive and independent of specific domains. However, these methods handled only the polarity of expressions.

Kobayashi et al. (2006) and Miyazaki et al. (2006) have built annotated corpora handling other information such as an aspect of the evaluation. We have also built an annotated corpus with evaluative criteria and the polarity. These methods were based on manual annotation. An advantage of manual annotation is that we can give more detailed information. However, it cannot build a massive corpus easily because it needs many costs. Another problem is reliability of the corpus. For building a reliable corpus, approaches using past annotated examples (Miyazaki et al., 2006) and generating gold standard by extracting agreement of the annotation (Ku et al., 2007) have been proposed. Snow et al. (2008) have shown the effectiveness of collecting non-expert annotations by using Amazon's Mechanical Turk system. Treating non-expert annotator is an important task.

As mentioned above, we built the annotated corpus as a preliminary experiment. However, the reliability of the corpus was not enough. We need to discuss a method for effective construction and expansion of a massive and reliable corpus using the existing corpus. In this paper, we develop an annotation support tool. We analyze our corpus to identify an association between evaluative criteria and a word. By applying features extracted from the existing corpus to the tool, we build a reliable corpus efficiently. Furthermore, we apply a domain estimation method proposed by Hashimoto and Kurohashi (2008) to our task and estimate the association between evaluative criteria and words.

2 Data

2.1 Corpus Annotation

In this paper, we treat game review documents which Shimada and Endo (2008) used for evaluative documents classification. The review documents were extracted manually from the Web site[1]. Seven evaluative criteria are given to each review, i.e., <Originality (o)>, <Graphics (g)>, <Music (m)>, <Addiction (a)>, <Satisfaction

[1] http://ndsmk2.net/

(s)>, <Comfort (c)>, and <Difficulty (d)>.

As a preliminary experiment, two annotators (A_1, A_2) annotated approximately 5,000 sentences in the review documents. Principles of the annotation process were as follows: (1) the tag set consists of 7 evaluative criteria mentioned above, (2) the polarity is Positive (P) or Negative (N), (3) the target is a sentence or a short phrase, and (4) the number of tags for an evaluative expression is one or more. Figure 1 shows an example of the actual annotation.

In this annotation, A_1 annotated 3,446 expressions and A_2 annotated 1,589 expressions. The reason why there was a difference of the number of annotated expressions is that the A_1 re-annotated them when the data was used for another sentiment analysis task that we studied.

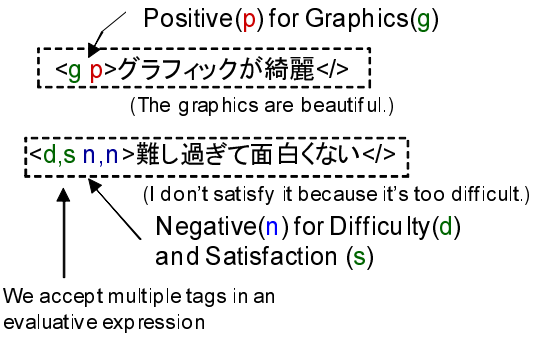

Figure 1: An example of the annotation.

2.2 Agreement of the existing corpus

A reliability measure for a corpus constructed is agreement between annotators. We calculated the agreement between A_1 and A_2. The rate of which both annotators detected the same expression was 42.7% and the rate of which annotators gave the same tag was 0.456 on κ value[2]. Ku et al. have reported that this value can be regarded as the moderate agreement in a scale of the κ value. However, it was not enough for the reliable corpus. In addition, there existed evaluative expressions which annotators failed to detect them.

3 Annotation support tool

In this section, we explain an annotation support tool to build more reliable corpus efficiently. The following three functions are implemented to our tool; (1) GUI support, (2) identification of association between evaluative criteria and words, and (3) presentation of annotated examples. Figure 2 shows the interface of the tool.

3.1 GUI support

Annotators used a text editor for building the corpus in section 2. In this case, there were problems of increase in the annotator's workload and decrease in efficiency of construction of the corpus. For solving these problems, we develop a GUI for simplification of annotation. It helps annotators work with a mouse. As a result, it leads to reduction of annotator's cost.

3.2 Identification of association between evaluative criteria and words

To reduce the cost of annotation, our tool indicates expressions which have possibility to be given annotation tag (see (a) in Figure 2). We firstly extract expressions associated strongly to evaluative criteria from the annotated corpus explained in section 2. For example, a word "BGM" tends to appear frequently in expressions given the evaluative criterion tag <Music>. Therefore, "BGM" and <Music> are regarded as associative and extracted as a pair. We define how strongly a word and evaluative criteria are associated by using the tf-idf algorithm. We reform the tf-idf algorithm by treating the number of annotated expressions and evaluative criteria. The process for identifying the association is as follows:

1. divide annotated expressions to morphemes by using a morphological analyzer[3],
2. for each word i and evaluative criterion j given to the expression, we count the frequency of i for j ($freq(i,j)$) and the number of words belonging to j (words(j)).
3. apply the tf-idf algorithm.

We define the tf-idf value as below.

$$tf_j^i = \frac{\log(freq(i,j)+1)}{\log(words(j))}$$

$$idf_i = \frac{Sum_{all}}{Sum_{include(i)}}$$

where Sum_{all} is the sum of annotated expressions and $Sum_{include(i)}$ is the sum of annotated expressions including i. Finally, the tf-idf value in our method is $tf_j^i \times idf_i$. Figure 3 shows the process of the identification.

[2] κ value is a statistical measure of inter-annotator agreement. It removes the agreement occurring by chance.

[3] We used Mecab. http://mecab.sourceforge.net

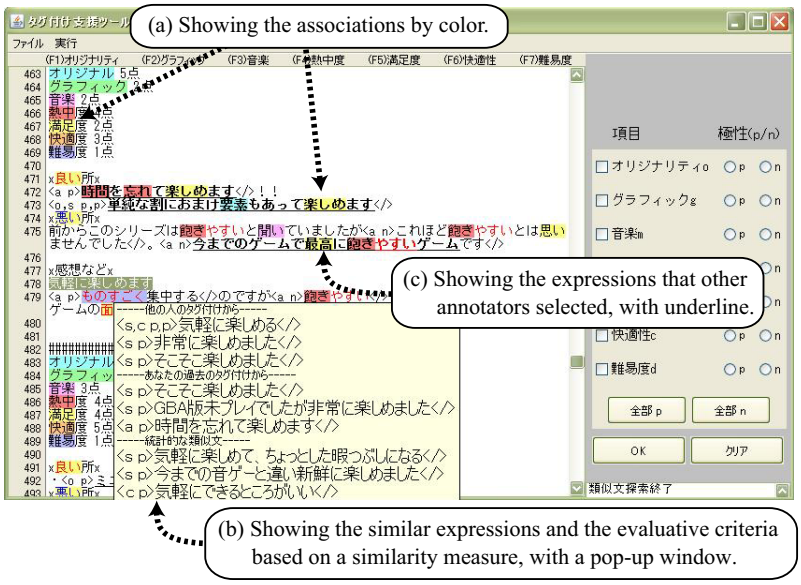

Figure 2: Our annotation support tool.

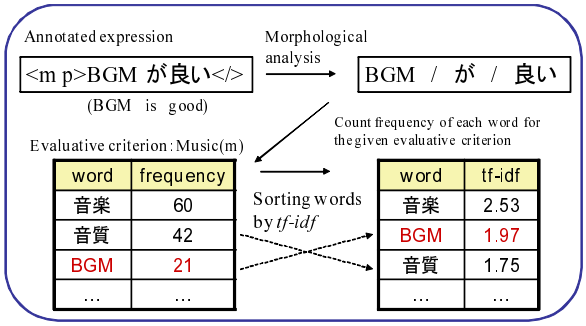

Figure 3: The process of identification.

3.3 Presentation of annotated examples

Miyazaki et al. (2006) have reported that presentation of annotated examples for annotators improved the agreement between annotators. Annotators can share their judgment for annotation by referring to the examples. We apply the method into our tool in the similar way. Our tool treats two types of annotated examples; (1) similar examples and (2) examples from the same documents. Similar examples show which tag is given to the similar expression. On the other hand, examples from the same document directly show how the other annotator annotated on the same document.

(1) Similar examples from annotated corpora

Our tool displays annotated examples which are similar to an expression (see (b) in Figure 2). These examples are extracted from annotated corpora. We use two types of similar examples. One is other annotator's annotation for different documents. It indicates a statistical annotation tendency. By using this tendency, annotators share their judgment. The other is to use own annotation. By referring own annotation, the annotator can maintain consistency in the annotation process. We define the degree of similarity between two expressions U_x and U_y as below.

$$SIM(U_x, U_y) = \frac{2M_{xy}}{M_x + M_y}$$

$$\begin{cases} M_{xy} : \text{the number of both expressions match of Bigram} \\ M_x, M_y : \text{the number of combinations of Bigram of } U_x, U_y \end{cases}$$

(2) Examples of other annotator's annotation from the same documents

These examples show (1) which expression other annotator detected and (2) which tag other annotator gave to the expression. Annotators can use them as a reference (see (c) in Figure 2). Thereby, high agreement of the annotation is expected[4].

3.4 Experiment

We evaluated the annotation support tool. We used test datasets (Ex.1 and Ex.2) which were different from the annotated documents in section 2. The numbers of sentences of the test datasets were 350 sentences for Ex.1 and 450 sentences for Ex.2.

[4]Note that annotators use this function as revision after their own annotation. A preliminary experiment showed that annotators failed to detect evaluative expressions if they refer to the examples from the beginning. This was because annotators were influenced too much by the presented example.

Table 1: Result of each annotator (Ex.1st).

Annotator \ Case		1	2	3	4
T_1	Num of DE	323	330	325	**342**
	Num of Tag	412	423	414	**426**
	Time(min)	73	43	64	115
T_2	Num of DE	382	**384**	365	374
	Num of Tag	409	431	433	**482**
	Time(min)	120	95	125	185

Table 2: Agreement of annotation (Ex.1st).

Case	1	2	3	4
Agreement of DE	0.828	0.824	0.808	**0.874**
Tag (κ value)	0.350	0.419	0.442	**0.573**

Table 3: Result of each annotator (Ex.2nd).

Annotator \ Case		1	2	3	4
T_1	Num of DE	—	**415**	395	**415**
	Num of Tag	—	**526**	508	**526**
	Time(min)	—	70	99	157
T_2	Num of DE	—	464	462	**480**
	Num of Tag	—	530	536	**598**
	Time(min)	—	118	148	215

Table 4: Agreement of annotation (Ex.2nd).

Case	1	2	3	4
Agreement of DE	——	0.840	0.815	**0.857**
Tag (κ value)	——	0.491	0.526	**0.687**

Two annotators (T_1, T_2) evaluated our tool in this experiment. For evaluating each function of the support tool, we experimented on the same documents with the following four cases. The process was the following order.

Case 1. without the support tool

Case 2. GUI (3.1) + association (3.2)

Case 3. **Case 2** + similar examples (3.3 (1))

Case 4. **Case 3** + other's examples (3.3 (2)), namely the revision of **Case 3**

We skipped **Case 1** in Ex.2. In **Case 2**, we adopted 270 pairs of a word and the evaluative criterion and 173 pairs of a word and the polarity which we identified in section 3.2. In **Case 3**, we adopted 1,109 examples whose annotation between A_1 and A_2 completely agreed.

Table 1 to Table 4 show the results of the first (Ex.1) and second (Ex.2) experiments. In Table 1 and Table 3, "Num of DE" is the number of detected expressions, "Num of Tag" is the number of annotated tags, and "Time" is time of annotation[5]. Table 2 and Table 4 show the agreement of detected expressions and the κ value for tags.

The results of Table 2 and Table 4 indicated that the agreement became higher as more functions were used. Therefore, we think that using the annotation support tool is effective in improving the agreement of annotation. We achieved the best agreement in **Case 4**. This result shows that presentation of the other annotator's examples from the same documents was effective as revision.

In **Case 3**, we could not achieve enough improvement of κ value. This was because our tool could not present appropriate expressions as similar examples. We assume that we need to prepare more examples for the presentation. Besides, we verified the number and agreement of detected expressions was small in **Case 3**. It happened in both Ex.1 and Ex.2. The support by the presentation of the annotated examples affected detection of evaluative expressions. However, even in that case, annotators hardly missed evaluative expressions which should be detected obviously. We consider that annotators detected evaluative expressions sufficiently as compared with the size of the datasets.

From the viewpoint of efficiency, annotators with the support tool (**Case 2**) could annotate faster than that without the tool (**Case 1**). Furthermore, our tool improved the agreement of the annotation. This result indicated that annotators could work efficiently and build a reliable corpus by using our tool. However, in **Case 3** and **Case 4**, annotators required more working time although the agreement improved. We need to discuss the usage of the functions in terms of working time of annotators.

The agreement of the annotation in Ex.2 was better than that in Ex.1. We think the reason that annotators could share their judgment of annotation by referring to other annotator's examples in **Case 4** of Ex.1. In other words, the revision (**Case 4**) affects future annotation of the annotators. The experience by **Case 4** might lead to the agreement of annotation in other annotation tasks for the annotators. This result denotes that the information-sharing with **Case 4** is effective in case that the annotators expand the corpus.

In this experiment, we evaluated with two annotators. However, it was not enough to evaluate our tool accurately. Therefore, we need to experiment

[5]The time of **Case 4** is the sum of **Case 3** and **Case 4** because it is the revision of **Case 3**.

with more annotators.

4 Evaluative criteria estimation

In section 3, we identified expressions associated to the evaluative criteria. They were extracted from the existing annotated corpus. However, these extracted expressions depend on the data. In other words, the method in section 3 cannot treat unknown expressions.

Hashimoto and Kurohashi (2008) have estimated a domain of a word by using the Web. The domain of the word means, for example, the word "Baseball" belongs to the domain <Sports> and the word "Parliament" belongs to the domain <Government>. We apply the method into our task. Here we assume the domains defined in the previous work to be the evaluative criteria in our task. By applying the method to our tool, we estimate the evaluative criterion of a word in the corpus. In the previous method, they computed a threshold for sorting the domains. We apply another approach for the determination of the threshold in the method.

4.1 Score calculation

First, we need keywords for the estimation process because it is based on a co-occurrence between a word that we want to estimate and keywords that are prepared manually. We prepared 20 keywords collected manually for each criterion. They represent contents of criteria, for example, a word "sound" is a keyword for <Music>.

Next, we compute A_c score between a word and a criterion[6]. An A_c score denotes the likelihood that a word belongs to a criterion. The A_c score is calculated by summing up the top five A_k scores of the criterion. An A_k score is a co-occurrence in the Web between a word and each keyword. The A_k score between a word (w) and a keyword (k) is computed as follows:

$$A_k(w,k) = \frac{n(ad-bc)^2}{(a+b)(c+d)(a+c)(b+d)}$$

where n represents the total number of Japanese Web pages. We supposed it to be 10 billion. a,b,c,d is given as below.

$$\begin{array}{rcll} a & = & hits(w\ \&\ k) & b = hits(w) - a \\ c & = & hits(k) - a & d = n - (a+b+c) \end{array}$$

where $hits(q)$ is the number of search engine hits when q is used as a query.

4.2 Determination of an evaluative criterion

We calculate A_c score as well as the previous work. If there is no threshold in this process, every word is associated with any evaluative criteria. Hashimoto and Kurohashi dealt with this problem by computing a threshold function for rejecting domains which have low A_d score. However, we apply a different approach in this paper because a huge amount of data is required for their method. In our method, we compute a f_d score by the following equation:

$$f_d = \frac{A_c - \mu}{\sigma}$$

where σ represents the standard deviation of a A_c score, μ represents the average of A_c scores, and $A_c - \mu$ denotes the deviation. We sort evaluative criteria on the basis of the score f_d. Finally, we select the 1st evaluative criterion for each word. If $f_d < K$, we do not accept any evaluative criteria for a word[7]. Here K is the threshold that is defined manually in our method.

4.3 Experiment

We evaluated the evaluative criteria estimation method. For each criterion, we prepared 20 words as a test data[8]. Here we subjectively classified each word in the test data into an evaluative criterion. We regard the classification results as the correct class of each word in the test data. We set the threshold value K to 1.4. This value was determined with a preliminary experiment.

The accuracy of the evaluative criteria estimation was 51.8%. The accuracy was not enough. However, the estimation results often contained evaluative criteria that were admissible as the correct criterion of the word although it was not matched with the subjective evaluative criterion that we selected. For example, we think that "be crazy about" in <Addiction> is admitted as <Satisfaction>. Table 5 shows examples of the estimation results of each evaluative criterion.

As the tendency of the estimation, many words were associated with <Satisfaction>. This was

[6] In the previous work, they called it A_d score. In this paper, we call it A_c because it is a score for each "criterion".

[7] Actually, if f_d of a word was less than K, we assign it to the <Not associable> tag.

[8] The test data did not include evaluative criterion words such "Graphics" and keywords mentioned in 4.1 because it is evident that they are estimated correctly.

Table 5: Examples of evaluative criteria estimation.

Prospective evaluative criterion: Addiction		Prospective evaluative criterion: Comfort	
target word	estimated criterion	target word	estimated criterion
上達 (improvement)	Addiction	キー (key)	Comfort
集め (collect)	Satisfaction	メニュー (menu)	Originality
夢中 (be crazy about)	Satisfaction	疲労 (fatigue)	Comfort
没頭 (absorption)	Satisfaction	邪魔 (obstacle)	Satisfaction

Prospective evaluative criterion: Difficulty		Prospective evaluative criterion: Graphics	
target word	estimated criterion	target word	estimated criterion
判定 (judgment)	Difficulty	鮮やか (vivid)	Graphics
アクション (action)	Originality	オープニング (opening)	Music
困難 (arduousness)	Difficulty	CG	Graphics
得意 (be good at)	Satisfaction	アニメ (animation)	Music

Prospective evaluative criterion: Music		Prospective evaluative criterion: Originality	
target word	estimated criterion	target word	estimated criterion
アレンジ (arrangement)	Music	機能 (function)	Comfort
ボイス (voice)	Music	ユニーク (unique)	Originality
うるさい (loud)	Satisfaction	個性的 (original)	Originality
リズム (rhythm)	Music	閃き (inspiration)	Satisfaction

Prospective evaluative criterion: Satisfaction		Estimated criterion: Not associable	
target word	estimated criterion	target word	
全体 (as a whole)	Satisfaction	繰り返し (repetition)	探索 (quest)
豊富 (abundant)	Originality	必要 (necessity)	ミッション (mission)
イマイチ (not enough)	Satisfaction	評価 (evaluation)	ソフト (software)
気に入る (favorite)	Satisfaction	自由 (freedom)	ムービー (movie)

because keywords associated with <Satisfaction> frequently co-occurred with any words. This result shows that it is important to assign appropriate keywords to each evaluative criterion.

Ranking the associations in each criterion is useful in practice. By using ranked associations, we can evaluate how strongly a word is associated with an evaluative criterion. One approach is to use the f_d score. However, we can not directly handle the f_d score for the ranking because the scale of the score in our current approach is different in each word. We need to discuss this problem.

5 Conclusion

In this paper, we developed the annotation support tool. For effective construction of a reliable corpus, we used the exiting corpus. The experiment result showed that our tool could improve the reliability of the corpus. Furthermore, we applied a domain estimation method into our task, evaluative criteria estimation. As a result, we verified that the method was effective in our task.

Future work includes (1) construction of a large-scale corpus and (2) improvement of evaluative criteria estimation.

References

C. Hashimoto and S. Kurohashi. 2008. Blog categorization exploiting domain dictionary and dynamically estimated domains of unknown words. In *Proceedings of ACL HLT 2008*, pages 69–72.

N. Kaji and M. Kitsuregawa. 2007. Building lexicon for sentiment analysis from massive html documents. In *Proceedings of EMNLP-CoNLL2007*, pages 1075–1083.

N. Kobayashi, K. Inui, and Y. Matsumoto. 2006. Designing the task of opinion extraction and structurization. In *IPSJ SIG Notes, NL171-18*, pages 111–118.

L.-W. Ku, Y.-S. Lo, and H.-H. Chen. 2007. Test collection selection and gold standard generation for a multiply-annotated opinion corpus. In *Proceedings of ACL 2007*, pages 89–92.

R. Miyazaki, N. Maeda, and T. Mori. 2006. Analysis of manual annotation of sentiment information in text and an annotation supporting tool. In *IPSJ SIG Notes, NL176-21*, pages 143–150.

K. Shimada and T. Endo. 2008. Seeing several stars: a rating inference task for a document containing several evaluation criteria. In *Proceedings of PAKDD 2008*, pages 1006–1014.

Rion Snow, Brendan O 'Connor, Daniel Jurafsky, and Andrew Y. Ng. 2008. Cheap and fast - but is it good? evaluating non-expert annotations for natural language tasks. In *Proceedings of EMNLP-08*.

H. Takamura, T. Inui, and M. Okumura. 2005. Extracting semantic orientations of words using spin model. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL2005)*, pages 133–140.