Collecting Evaluative Expressions for Opinion Extraction

Nozomi Kobayashi Kentaro Inui Yuji Matsumoto
Nara Institute of Science and Technology
Takayama, Ikoma, Nara, 630-0192, Nara
{nozomi-k,inui,matsu}@is.aist-nara.ac.jp
Kenji Tateishi Toshikazu Fukushima
NEC Internet System Lab.
Takayama, Ikoma, Nara, 630-0101, Nara
{k-tateishi@bq,t-fukushima@cj}.jp.nec.com

Abstract

Automatic extraction of human opinions from Web documents has been receiving increasing interest. To automate the process of opinion extraction, having a collection of evaluative expressions such as "*the seats are comfortable*" would be useful. However, it can be costly to manually create an exhaustive list of such expressions for many domains, because they tend to be domain-dependent. Motivated by this, we have been exploring ways to accelerate the process of collecting evaluative expressions by applying a text mining technique. This paper proposes a semi-automatic method that uses particular cooccurrence patterns of evaluated subjects, focused attributes and value expressions.

1 Introduction

There are explosively increasing number of Web documents that include human opinions, indicating dissatisfaction with products, complaints about services, and so on. Automatic extraction of such opinions has been receiving interest from the NLP and text mining communities (Dave et al., 2003; Murano and Sato, 2003; Morinaga et al., 2002).

The following is an excerpt from a message board on a car review site.

The seats are very comfortable and supportive.
But the back seat room is tight . . .

This example suggests that the core of an opinion typically consists of three elements: an evaluated subject, focused attribute and value. One can extract the following triplets from above sentences:

⟨*Product_X, seats, comfortable*⟩
⟨*Product_X, seats, supportive*⟩
⟨*Product_X, back seat room, tight*⟩

Once opinions are obtained in the form as above, one can, for example, statistically analyze them and summarize the results as radar-charts in a fully automatic manner. In fact, our group has developed a prototype system that generates radar-charts of opinions extracted from review sites (Tateishi et al., 2004).

Motivated by this, we are aiming at the development of an automatic method to extract opinions, each of which is specified by a triplet ⟨evaluated subject, focused attribute, value⟩ from Web documents.

One approach to this goal is to use a list of expressions which possibly describe either evaluated subject, focused attribute or value (referred to subject expressions, attribute expressions, and value expressions, hereafter). Presumably, given a target domain, it is not difficult to obtain a list of expressions of subjects (product names, service names, etc.). However, it can be considerably expensive to manually create an exhaustive list of attribute and value expressions for many domains, because they tend to be domain-dependent. For example, "*gas mileage*" is an attribute expression in the car domain, but is not in the computer domain. The purpose of this paper is to explore how to reduce the cost of creating a list of evaluative expressions: attribute expressions and value expressions. We propose a semi-automatic method that uses particular cooccurrence patterns of subjects, attributes and values. We then report experimental results and show its efficiency compared to manual collection of those expressions.

2 Related work

Tarney (2002) and Pang et al. (2002) propose a method to classify reviews into *recommended*

or *not recommended*. While their work focuses on document-wise classification, some other researchers approach sentence-wise classification. Yu and Hatzivassiloglou (2003), for example, address the task of discriminating opinion sentences from factual sentences and classifying opinion sentences into positive or negative. Hatzivassiloglou and Wiebe (2000) discuss the usefulness of gradable adjectives in determining the subjectivity of a sentence. These research aim at the determination of the specific orientation (positive / negative) for sentence or document. In contrast, we aim not only at classifying opinions as positive or negative, but also at extracting the grounds why the opinion determined to be positive or negative. We will realize extracting the grounds by extraction of triplets.

There have also been several techniques developed for acquiring subjective words. For example, Hatzivassiloglou and McKeown (1997) propose a method to identify the semantic orientation (positive / negative) of adjectives. Riloff et al. (2003) apply bootstrapping algorithms to obtain subjective nouns. These work intend to collect the words that are useful for determining subjectivity. As mentioned above, in order to extract triplets from opinions, we will collect expressions with the help of specific patterns that relates some of the elements in the triplets ⟨evaluated subject, focused attribute, value⟩.

3 Attribute and value

Let us first discuss what sorts of expressions we should collect as attribute and value expressions for the sake of opinion extraction.

Consider one example, "*the leather seat (of some Product_X) is comfortable*". This opinion can be considered to contain at least the following information:

- Subject: The subject of evaluation is *Product_X*.
- Attribute: The opinion focuses on a particular aspect of *Product_X*, "*the leather seat*".
- Value: The opinion says that the value of the attribute of *Product_X* is "*comfortable*".

To be more general, we consider an opinion as a chunk of information consisting of these three slots: ⟨*Subject*, *Attribute*, *Value*⟩. The attribute slot specifies which aspect of a subject is focused on. Attributes of a subject of evaluation include its qualitative and quantitative properties, its constituents, and services associated with it. The value slot specifies the quantity or quality of the corresponding aspect. The goal we pursue in this paper is to build a lexicon of linguistic expressions that can be used to realize attributes or values in the above sense.

Note that an opinion may also have a specific orientation (i.e. favorable or unfavorable). For example, "*I like the leather seats of Product_X*" expresses the writer's favorable orientation to the attribute "*the leather seats*". One may want to introduce the fourth slot and define an opinion as 4-tuple ⟨*Subject*, *Attribute*, *Value*, *Orientation*⟩. However, it is not necessarily worthwhile because the distinction between Value and Orientation is sometimes messy. We thus simply regard Orientation as a special type of Value.

4 Collecting expressions using cooccurrence patterns

Opinions can be linguistically realized in many ways. One of the typical forms would be:

⟨*Attribute*⟩ *of* ⟨*Subject*⟩ *is* ⟨*Value*⟩.

We use such typical patterns of textual fragments as clues for collecting attribute and value expressions. For example, applying the above coocurrence pattern to "*the leather seat of Product_X is comfortable*", we can learn that "*the leather seat*" may be an attribute expression and "*comfortable*" a value expression. If we have already known that "*comfortable*" is a value expression, we can reason that "*leather seat*" is more likely to be an attribute expression.

Figure 1 illustrates the process of collecting attribute/value expressions. The overall process consists of repeated cycles of *candidate generation* followed by *candidate selection*. In each cycle, the candidate generation step automatically produces an ordered list of candidates for either attribute or value expressions using coocurrence patterns and the current dictionaries of subject, attribute and value expressions. In the candidate selection step, a human judge selects correct attribute/value expressions from the list and add them to the dictionaries. Updates of the dictionaries may allow the candidate generation step to produce different candidates. Repeating this cycle makes both the attribute and value dictionaries richer gradually.

4.1 Candidate generation

The candidate generation step uses three kinds of resources: (a) a collection of Web documents, (b) a set of coocurrence patterns, and (c) the latest version of the subject dictionary, the attribute dictionary and the value dictionary.

Suppose that we have the following coocurrence pattern:

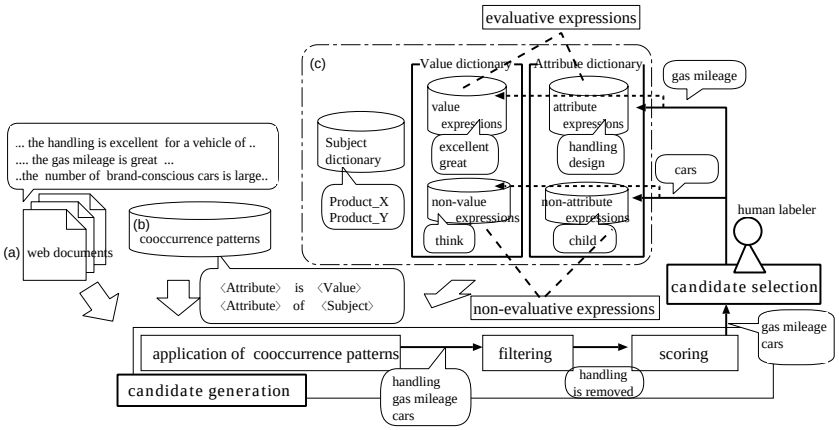

Figure 1: Semi-automatic process of collecting attribute/value expressions

$\underline{\langle Attribute \rangle}$ *is* $\langle Value \rangle$.

In this notation, we assume that $\langle Value \rangle$ corresponds to an already known value expression and the underlined slot $\underline{\langle Attribute \rangle}$ denotes an expression that can be taken as a candidate of an attribute expression. If our document collection includes sentences such as[1]:

... $\langle \underline{the\ handling} \rangle_a$ *is* $\langle excellent \rangle_v$ *and ...*

... $\langle \underline{the\ gas\ mileage} \rangle_a$ *is* $\langle great \rangle_v$ *...*

We can obtain "*the handling*" and "*the gas mileage*" as candidates for attribute expressions.

Here we must note that such cooccurrence patterns may also generate non-evaluative candidates as in the following case, from which a candidate expression "*cars*" is extracted:

... the $\langle \underline{cars} \rangle_a$ *is* $\langle large \rangle_v$ *so that ...*

To reduce the labor of manual checking of such non-evaluative expressions, we first filter out candidates that have already been registered either in the attribute and value dictionaries. For this purpose, each dictionary is designed to keep expressions that have been judged as non-evaluative expressions in an earlier cycle as well as evaluative expressions. In case of Figure 1, "*handling*" is filtered out because it is already registered as an attribute expression. In addition to this simple filtering, we also use a statistics-based scoring function to rank extracted candidates and provide the human judge with only a limited number of highly ranked candidates. The details of the scoring function we used in the experiments will be given in Section 5.1.

4.2 Candidate selection

In the candidate selection step, a human judge labels an arbitrary number of highly ranked candidates and register them into the dictionaries. In Figure 1, given two candidates "*gas mileage*" and "*cars*", the human labeler has judged the former as attributive expression and the latter as non-attributive expression.

5 Experiments

We conducted experiments with Japanese Web documents in two domains, cars and video games (simply game, hereafter), to empirically evaluate the effectiveness of our method compared to a manual collection method. In the experiments, we hired a person as the examiner who had no knowledge about the technical details of our method.

5.1 Semi-automatic collection

5.1.1 Resources

Document collections: We collected 15 thousand reviews (230 thousand sentences) from several review sites on the Web for the car domain and 9.7 thousand reviews (90 thousand sentences) for the game domain.

Subject dictionaries: For subject expressions, we collected 389 expressions for car domain (e.g. *"BMW"*, *"TOYOTA"*) and 660 expressions for the game domain (e.g. *"Dark Chronicle"*,*"Seaman"*).

Initial attribute dictionary: For the seed set of attribute expressions, we manually chose the following 7 expressions for both domains that considered to be used across different domains:

nedan (cost), *kakaku* (price), *sâbisu* (service), *seinou* (performance), *kinou* (function), *sapôto* (support), *dezain* (design).

Initial value dictionary: For the seed set of value expressions, we used an existing thesaurus

[1] $\langle \rangle_a$ denotes the word sequence corresponding to the attribute slot of the cooccurrence pattern. Likewise, we also use $\langle \rangle_v$ for the value slot and $\langle \rangle_s$ for the subject slot.

Table 1: cooccurrence patterns

Pat.1	⟨*Value*⟩-MOD ⟨*Subject*⟩	Pat.2 ⟨*Value*⟩-MOD ⟨*Attribute*⟩	Pat.3 ⟨*Value*⟩-MOD ⟨*Attribute*⟩
e.g.	⟨*shibutoi*⟩$_v$ ⟨*Product_1*⟩$_s$	e.g. ⟨*yasuppoi*⟩$_v$ ⟨*dezain*⟩$_a$	e.g. ⟨*subarashii*⟩$_v$ ⟨*handoringu*⟩$_a$
	stubborn Product_1	cheap design	great handling
	(. . .stubborn Product_1. . .)	(. . .cheap design. . .)	(. . .great handling. . .)
Pat.4	⟨*Subject*⟩-no ⟨*Attribute*⟩	Pat.5 ⟨*Attribute*⟩-{ga,wa,etc.} ⟨*Value*⟩	Pat.6 ⟨*Attribute*⟩-{ga,wa,etc.} ⟨*Value*⟩
e.g.	⟨*Product_3*⟩$_s$-no ⟨*dezain*⟩$_a$	e.g. ⟨*nennpi*⟩$_a$-ga ⟨*yoi*⟩$_v$	e.g. ⟨*interia*⟩$_a$-ga ⟨*yoi*⟩$_v$
	Product_3-of design	gas mileage-TOP great	interior-TOP nice
	(the design of Product_3)	(the gas mileage is great)	(the interior is nice)

Pat. 7	⟨*Subject*⟩-no ⟨*Attribute*⟩- {ga,wa,etc.} ⟨*Value*⟩
e.g.	⟨*Product1*⟩$_s$-no ⟨*interia*⟩$_a$-wa ⟨*kirei.*⟩$_v$
	Product1-of interior-TOP beautiful.
	(the interior of Product1 is beautiful.)
Pat. 8	⟨*Subject*⟩-no ⟨*Attribute*⟩-{ga,wa,etc.} ⟨*Value*⟩
e.g.	⟨*Product_2*⟩$_s$-no ⟨*enjine*⟩$_a$-ha ⟨*pawafuru*⟩$_v$
	Product_2-of engine-TOP powerful
	(the engine of Product_2 is powerful.)

and dictionaries to manually collect those that were considered domain-independent, obtaining 247 expressions, most of which were adjectives. The following are examples of them:

yoi (good), *kirei*(beautiful), *akarui* (bright), *kiniiru* (like / favorite), *takai* (high)

Cooccurrence patterns: We preliminarily tested various cooccurrence patterns against another set of documents collected from the domain of mobile computers. We then selected eight patterns as shown in Table 1 because they appeared relatively frequently and exhibited reasonable precisions. In addition to above patterns, we used another heuristic rule which indicate attribute and value expressions with suffixes. For example, we regard "antei-sei" (stability) as a candidate of attribute.

To reduce noise in the extraction, we specify the applicability condition of each pattern based on part-of-speech. For attributes, we extract unknown words, nouns (including compound nouns), except numerical expressions. For values, we extract only adjectives, verbs (including *sahen*-verbs), nominal adjectivals and noun phrases.

5.1.2 Scoring

With the above part-of-speech-based restrictions, Pat.4 to Pat.6 are still relatively underconstrained and tend to generate many non-evaluative expressions. We thus introduce a scoring function that accounts two scoring factors. One factor is the frequency of extracted expressions; namely, candidates with a high frequency in the target document collection have the preference. The other factor is the reliability of clues used for extraction. Suppose that we want to estimate the reliability of an instantiated cooccurrence pattern, say, "⟨*Attribute*⟩ *is low*". If this pattern has produced not only correct candidates such as "*cost*" and "*seat position*" but also many non-evaluative candidates such as "*body height*", we can learn from those results that the pattern is not so reliable, presumably less reliable than, say, "⟨*Attribute*⟩ *is comfortable*" which has produced very few non-evaluative candidates. Based on this consideration, we estimate the reliability of an instantiated pattern by the log-likelihood ratio between candidates and evaluative expressions.

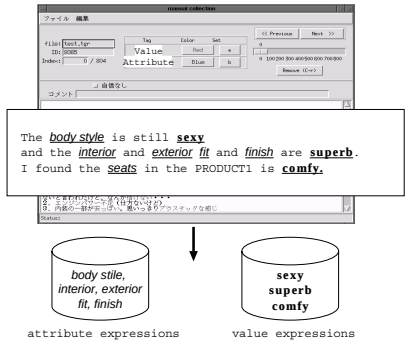

Figure 2: Interface of the manual collection tool

5.2 Manual collection

We conducted experiments comparing a manual method to our semi-automatic method. We had a human examiner tag attribute and value expressions using a tagging tool. Figure 2 shows an example of this process with the expressions that are tagged underlined. The examiner tagged expressions in 105 reviews (about 5,000 sentences) from the car domain and 280 reviews (about 2,000 sentences) from the game domain. Those reviews were taken from the same document collections that we used with our semi-automatic method. It is important to note that while the same person was responsible for both manual collection of evaluative expressions and judgment of our semi-automatic method, we avoided possible conflicts of interest by evaluating our method before manually collecting expressions.

6 Results and Discussion

6.1 Collection efficiency

Figures 3 and 4 show the plots of the number of collected expressions versus the required time.

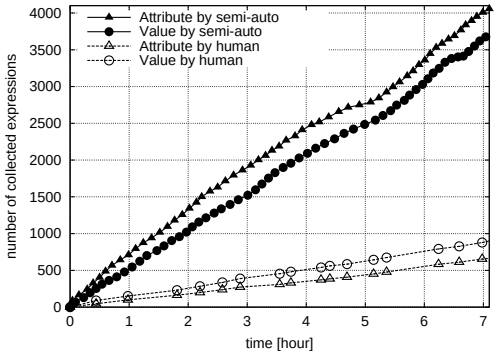

Figure 3: Number of collected expressions (car)

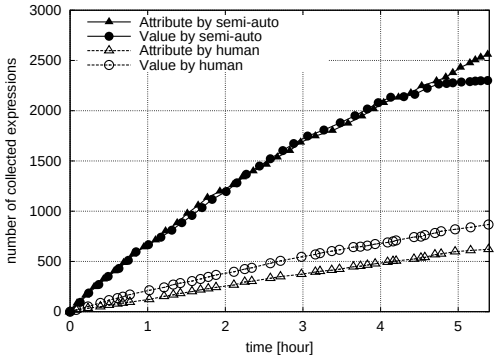

Figure 4: Number of collected expressions(game)

For the semi-automatic collection, we plot the cumulative number of expressions in each cycle of the collections process. For the manual collection, we plot the cumulative number of expressions collected from each 5 articles. The figures show that the semi-automatic method is significantly more efficient than the manual collection in collecting the same number of expressions. For example, the semi-automatic method takes only 0.6 hours to collect the first 500 attribute expressions while the manual extraction requires more than 5 hours. We also find that both domains exhibit quite similar tendencies. This indicates that our method is likely to work well in a wide range of domains. Recall that, preliminary to the experiments, we used documents in the mobile computer domain, which was considerably different from the car and game domains, to tune the cooccurrence patterns. This suggest that the same set of patterns will work well in other domains.

One problem observed from the results is that the number of extracted expressions does not exhibit convergence. We think that this is due to the lack of proper treatment of compound expressions. For example, the examiner chose "engine" and "response" as attribute expressions in the car domain; however she also registered "engine response" into the attribute dictionary as a different entry. We are seeking more sophisticated ways for dealing with compound expressions that accounts their internal semantic structure. One simple option is to regard expressions comprised only with already known expressions as granted. The plausibility of such treatment should be examined in future experiments.

6.2 Coverage

It is also important to see to what extent the set of semi-automatically collected expressions cover the set of manually collected expressions. We investigated the overlap between the human extracted and semi-automatically collected expressions, finding that the semi-automatic collection covered 45% of manually collected expressions in the car domain and 35% in the game domain. Table 2 shows examples, where "common" indicates expressions collected commonly in both ways, and "semi-auto" and "manual" are expressions collected only by each method.

There are several reasons why the coverage of the semi-automatic method is not very high. One is that the current semi-automatic method does not generate candidates spanning beyond base-phrase (bunsetsu) boundaries. The human examiner does not suffer from this restriction and can collect complex phrases such as "*me kara uroko* (see the light)" and "*kaizen-no yochi ari* (there's room for improvement)" as well. This accounts for 30% of the total number of uncovered value expressions in the car and game domain.

In the attribute expressions, however, "*A no B*" ("B of A" in English) accounts for large share of uncovered attribute expressions (20% of the total number in the car domain and 27% for the game domain). "*A no B*" sometimes expresses a hierarchical relation between attributes, which are not recognized in our current setting. Consider "*engine no oto* (sound of the engine)" for example. This expression consists of "*engine*" and "*oto* (sound)", which are both attribute expressions. However, we should regard "*oto* (sound)" as an attribute of "*engine*". As this example shows, we should take into account hierarchical relations between attributes, deciding the range of attribute expressions.

6.3 Utility of cooccurrence patterns

Table 3 shows the usefulness of the patterns, where "number" indicates the number of expressions extracted by the patterns, and "correct/incorrect" indicates the number of value/non-value and attribute/non-attribute expressions.

Overall, the patterns that extract value expressions outperform the patterns that extract at-

Table 2: Examples of collected expression

		common	semi-auto	manual
car	Attribute	sutairu(style) bodî karâ (body color)	shajû (weight) seijaku sei(quietness)	SOHC enjin(engine type) akarui sikichou (light in color tone)
	Value	good(good) jyouhin (elegant)	arai (gross) miwakuteki (alluring)	tekitô (appropriate) 100%
game	Attribute	oto (sound) ivento CG(event CG)	batoru supîdo(battle speed) kihon sousa (basic operation)	purei jikan (play time) chitekina senryaku (intelligent strategy)
	Value	yasuppoi (cheap) tsumaranai (boring)	soudai (magnificent) komikaru (comical)	sutoresu ga tamaru (stressful) ki ga nukenai (exciting)

Table 3: Performance of cooccurrence patterns

		car			game		
		accuracy	number	correct/incorrect	accuracy	number	correct/incorrect
Value	pat.1,2	0.81	1362	1108/ 254	0.79	901	709/ 192
	pat.5	0.69	4917	3398/1519	0.82	2581	2119/ 462
	pat.8	0.67	239	159/ 80	0.93	15	14/ 1
Attribute	pat.3	0.42	895	372/ 523	0.24	894	214/ 680
	pat.4	0.46	726	331/ 395	0.63	40	25/ 15
	pat.6	0.76	5225	3965/1260	0.66	3975	2631/1344
	pat.7	0.58	273	159/ 114	0.56	23	13/ 10

tribute. We speculate several reasons: one is that value expressions also cooccur with named entities (e.g. product names, company names, and so on) or general expressions such as "*mono* (thing)". Another reason is that it is difficult to decide the scope of attribute expressions. As mentioned above, there is a hierarchical relation between attributes. For example, while "*character*" is an attribute, "character" may have its own attribute such as "*kao* (face)" or "*ugoki* (motion)". Presumably, whether this "attribute of attribute" is included in the attribute expression depends on the person making the judgment.

7 Conclusion

In this paper, we proposed a semi-automatic method to extract evaluative expressions based on particular cooccurrence patterns of evaluated subject, focused attribute and value. We reported the experimental results, which showed that our semi-automatic method was able to collect attribute and value expressions much more efficiently than manual collection and that the cooccurrence patterns we used in the experiments worked well across different domains. Our next step will be directed to the extraction of triplets ⟨*Subject, Attribute, Value*⟩ from the Web documents.

References

Kushal Dave, Steve Lawrence, and David M. Pennock. 2003. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th International World Wide Web Conference*.

Vasileios Hatzivassiloglou and Kathleen R. McKeown. 1997. Predicting the semantic orientation of adjectives. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and the 8th Conference of the European Chapter of the Association for Computational Linguistics* (*ACL-EACL*), pages 174–181.

Vasileios Hatzivassiloglou and Janyce M. Wiebe. 2000. Effects of adjective orientation and gradability on sentence subjectivity. In *Proceedings of the 18th International Conference on Computational Linguistics* (*COLING*), pages 299–305.

Satoshi Morinaga, Kenji Yamanishi, Kenji Tateishi, and Toshikazu Fukushima. 2002. Mining product reputations on the web. In *Proceedings of the Eighth ACM SIGKDD Internatinal Conference on Knowledge Discover and Data Mining*, pages 341–349.

Seiji Murano and Satoshi Sato. 2003. Automatic extraction of subjective evaluative sentences using syntactic patterns. In *Proceedings of the 9th Annual Meeting of the Association for NLP*, pages 67–70. (in Japanese).

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (*EMNLP*), pages 79–86.

Ellen Riloff, Janyce Wiebe, and Theresa Wilson. 2003. Learning subjective nouns using extraction pattern bootstrapping. In *Proceedings of the 7th Conference on Natural Language Learning* (*CoNLL*), pages 25–32.

Peter D. Tarney. 2002. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics* (*ACL*), pages 417–424.

Kenji Tateishi, Toshikazu Fukushima, Nozomi Kobayashi, Masayuki Wade, Tetsuro Takahashi, Takashi Inui, Atsushi Fujita, Kentaro Inui, and Yuji Matsumoto. 2004. Web opinion extraction and summarization based on product's viewpoint. In *Proceedings of the 10th Annual Meeting of the Association for NLP*. (in Japanese).

Hong Yu and Vasileios Hatzivassiloglou. 2003. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (*EMNLP*), pages 129–136.