Domain-specific Sentiment Analysis using Contextual Feature Generation

Yoonjung Choi, Youngho Kim, Sung-Hyon Myaeng
Korea Advanced Institute of Science and Technology
355 Gwahak-ro Yuseong-gu Daejeon, South Korea
+82-423506210
{choiyj35, bruceyhkim, myaeng}@kaist.ac.kr

ABSTRACT
This paper presents a novel framework for sentiment analysis, which exploits sentiment topic information for generating context-driven features. Since the domain-specific nature of sentiment classification led the task more problematic, considering more contextual-information such as topic or domain is essential. In our system, we first automatically extract sentiment clues in different domains by our observation. We identified that a sentiment clue is often syntactically related to a sentiment topic in a sentence, which is defined as a primary subject of sentiment expression, such as event, company, and person. We bootstrap from a small set of seed clues and generate new clues by utilizing linguistic dependencies and collocation information between sentiment clues and sentiment topics. Next, we learn a domain-specific sentiment classifier for each domain with the newly aggregated clues. We ran experiments to see how the bootstrapping algorithm to converge and aggregate new clues and verified that the extracted domain-context features are more effective than generally-used features in sentiment analysis by running them on the same sentiment classifier.

Categories and Subject Descriptors
H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *Information filtering*. I.2.7 [**Artificial Intelligence**]: Natural Language Processing – *Text analysis.*

General Terms
Algorithms, Experimentation.

Keywords
Sentiment analysis, sentiment classification.

1. INTRODUCTION
People are eager to know what others are thinking or feeling about subject matters such as products, politicians, and social issues, as witnessed by a rapid growth in online reviews and news groups (e.g., Amazon, New York Times). Sentiment Analysis (SA), the task of extracting positive or negative aspects of free texts, is quite useful for individuals, governments, and companies. Reflecting the importance, many researchers have analyzed sentiment-bearing texts such as product reviews and news articles [9, 12].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
TSA'09, November 6, 2009, Hong Kong, China.
Copyright 2009 ACM 978-1-60558-805-6/09/11…$10.00.

Although successes exist, there are still rooms for research. The previous research shows that sentiment clues such as "great" and "hate" play an important role in SA. By spotting such clue words in an expression, its polarity can be determined. However, as Stoyanov and Cardie addressed [15], a sentiment expression is characterized by its source, its polarity, and its topic or target, and the problem of determining the semantic orientation (i.e., sentiment polarity) is very context-sensitive. As such, Wilson et al [14] studied the polarity change of phrase-level clue (e.g., *little* threat) by identifying *polarity shifter* words such as "*little*", "*lack*", and "*abate*" (e.g., *abate* the damage). However, their approach to recognize the contextual polarity is limited in: 1) the necessity of sufficient training data (they employed a fully supervised approach), and 2) the observation of lexical-level context (a few words preceding or following of sentiment expression). Different from this, if we consider semantic-level context information, we could obtain a more accurate classifier. Specifically, the topic information largely influences the determination of semantic orientation. As an example, "NASDAQ up is accelerated" is judged as positive news about "NASDAQ" (sentiment topic). However, it is not easy to identify sentiment clues in the sentence, and the previous keyword spotting approach would fail to classify it as positive. Though we can consider "up" and "accelerate" as positive sentiment clues (since those words are frequently used in the context of positive sentences), the sentence like "Global temperature up is accelerated" would be erroneously classified; "up" and "accelerated" are negative clues for "Global temperature" (sentiment topic) in this sentence while those are positive features for "NASDAQ" in the previous example. So, the sentiment classification with only sentiment clues is flawed unless considering the sentiment topic. In addition, as Pang and Lee [4] discussed, a joint topic-sentiment analysis is quite desirable because it is assumable that the document set used in SA can be created by issuing a topic query on Web-search (i.e., the documents for SA would contain many sentiment expressions related to the query topic). Thus, we may assure that topic-specific features would enhance the existing sentiment classifier.

In order to alleviate the problems in SA, in this paper, we propose a novel framework for SA regarding domain-specific features induced by topic-context information. The classification system consists of two parts: contextual feature generation (extracting sentiment clues whose polarities are contextually induced) and domain-specific sentiment classifier construction. Firstly, to develop context features, we associate sentiment clues with contextual information such as sentiment topics and domains. For example, "unpredictable" (which is a positive clue in a movie domain, while it conveys a negative meaning for car functionality

[13]) should be treated as a positive clue only if it is used in a movie domain or used together with a specific topic such as "plot" as in "the plot of Harry Potter is unpredictable". In that, we generate a domain-specific lexicon using topic-context information, which would be more effective in discerning the polarity of a sentiment expression than a general purpose lexicon used in previous approaches. To extract domain-context sentiment clues, we employ a bootstrapping technique using a small set of seed clues in each domain. The task is to classify a candidate word into either a positive, negative, or neutral category in a progressive manner using a bootstrapping method. The method exploits a syntactic dependency between a sentiment topic (e.g. "U.S. beef"), which is likely to be domain-specific, and a clue candidate (e.g. "criticize") in a sentence like "A civil company criticized importing U.S. beef." Once a sentiment topic for a known sentiment clue is identified from a small set of training sentences, a new clue can be detected when it co-occurs with the sentiment topic frequently in the same domain. In other words, we determine the contextual polarity of a new clue if it is associated with an old clue via a sentiment topic.

After a set of new domain-specific, contextually-driven clues is extracted, a new SA classifier is learned using the expanded lexicon to generate a new set of polarity-determined expressions (sentences) to be used for the next iteration of clue extraction. This process stops when no more clues are added. We expect that the domain-specific features would be more discriminative than those collected for general purposes. In addition, since the features utilized in the classifier are induced by topic-context information, the classifier inherently implies topic information.

2. RELATED WORK

In this paper, we incorporate two tasks (i.e., sentiment clue generation and sentiment classification) into the system of domain-context sentiment analysis. As related studies, we describe related works about sentiment lexicon generation and sentiment classification.

With a growing interest in SA, many researchers put some efforts for the task of sentiment lexicon generation. [21] utilized conjunctions (e.g., "and" conjoins two adjectives which are equally polarized) as was also done by [16] recently. [13] used co-occurrences between seed clues and new clues whereas [1] exploited WordNet by capturing lexical relations (i.e., antonym and synonym) of clues and their glosses (e.g., the glosses of "intrepid" and "good" would share the same keywords). However, they all share the same limitation of not considering topic or domain-specificity, without which the number of sentiment clues that can be obtained is smaller or they are inaccurate

Many of previous studies on sentiment classification dealt with document-level classification; for example, classifying a review into positive or negative [9, 12, 22]. In contrast to document-level sentiment analysis, our task is classifying a sentence into positive, negative, or neutral. Previously, some research has dealt with the same problem. Yu and Hatzivassiloglou [23] determined the sentence-level polarity by summing up a polarity score for each word in the sentence. Kim and Hovy [24] counted the prior polarities of sentiment clues in the sentence. However, their approaches are somewhat naïve because they just averaged the polarity scores of prior gathered clues. Although Wilson et al [14] used statistical machine learning to disambiguate phrase-level contextual polarity, the contextual classifier is weak by requiring a large amount of training data. Also, none of published works has considered semantic-level context information (i.e., sentiment topic) for sentiment analysis.

3. PROBLEM FORMULATION

We begin with some definitions of the key elements in the proposed system for domain-context sentiment analysis using domain-specific lexicon generation.

A *query topic* is a question a user creates to retrieve relevant documents and may have multiple fields like title and description. Given a query topic $qt = \langle t_1, t_2, \ldots t_i \rangle$ represented as a vector of terms, a set of documents $D_{qt} = \{doc_1, \ldots, doc_n\}$ is retrieved by the system.

A *domain* is defined operationally to be a collection of query topics with a related theme such as "Asian economic crisis" and "Japan's collapse of bubble economy." This is the target for which a lexicon is constructed. While a set of documents retrieved by a single query topic alone can be used for building a more specific lexicon, it may present a problem of data sparseness, not to mention its usefulness in SA. More formally, a domain *dm* consists of a set of similar query topics, i.e., $dm = \{qt_1, \ldots, qt_n\}$ where qt_i is similar to qt_j for all i and j. A document collection for *dm* is defined as the union of document sets corresponding to *qt*'s in *dm*, i.e., $D_{dm} = \{D_{qt_1} \cup \ldots \cup D_{qt_n}\}$. A domain-specific lexicon is constructed from this collection.

A *sentiment clue* is a word that causes positive or negative sentiment in a text (a sentence in this paper). For example, the polarity of "People support Obama for his deterministic attitude" is determined to be positive because of the sentiment clue "support". We assume that each sentiment expression includes at least one sentiment clue that determines its polarity.

A *sentiment topic* is a real-world entity (e.g., person, company) or an abstract entity (e.g., event, policy), which is the primary subject of sentiment expression as intended by the source of sentiment (i.e., sentiment holder) [15]. The unit of a sentiment topic is assumed to be a noun phrase (e.g., Yasukuni Shrine) excluding a pronoun alone in the current work, and each sentiment expression contains zero or more sentiment topics since it may contain a pronoun that refers to a sentiment topic in a previous sentence. A sentiment topic is assumed to be strongly related to the sentiment clue in the same expression.

Given the definitions, we develop domain-specific sentiment analysis system which can 1) identify sentiment clues whose polarity values are contextually-driven and 2) classify a sentence into positive, negative, or neutral. In our system, we incorporate two core functionalities into a bootstrapping algorithm (which will be explained in Section 4). While some domain-independent clues like "happy" can be associated with a sentiment topic in a particular domain, domain-specific clues are assumed to be found only within a domain context. While a sentiment clue can be found in multiple domains, possibly with different polarity values associated with it, we focused on identifying sentiment clues in a particular domain using a domain-specific corpus.

4. DOMAIN-SPECIFIC SENTIMENT ANALYSIS

In a domain-specific sentiment analysis, the first step is to prepare a domain corpus, which contains relevant documents for a set of queries in the domain. We then employ a bootstrapping technique

to extract context features and develop a domain-specific sentiment classifier, which are the core of this paper.

4.1 Domain Corpus Preparation

A news article collection is a rich resource for SA, containing both positive and negative sentences. The recent Opinion Analysis track of NTCIR Workshop [10, 11] has used news data for the SA task, which includes English news sentences whose polarity values are annotated as positive, neutral, or negative [19, 20]. It consists of 45 query topics and 606 relevant articles associated with them. Since we need a sufficient number of documents for each domain, we manually grouped several similar topics as well as the associated documents into four domains to generate four domain corpora (see Section 5.1 for more details).

4.2 Bootstrapping algorithm

The bootstrapping algorithm consists of four steps. After the preprocessing step (Step 1), steps 2, 3, and 4 are repeated until no new clues are introduced. Each step is described in detail in the following sub-sections.

Step 1: Preprocessing. The algorithm starts with a small set of training examples, each of which consists of a sentence and its polarity, and generate a set of seed clues.

Step 2: Identifying Sentiment Topics. Sentiment topics are extracted from the sentences in the training examples by utilizing the clue set.

Step 3: Generating Sentiment Clues. Sentiment clues that are contextually related to the sentiment topics are identified and added to the current clue set.

Step 4: Learning Sentiment Classifier. A domain-specific sentiment classifier is learned based on the updated clues and the training examples. The classifier is applied to the domain corpus to generate new training examples. If new training examples are found, Steps 2, 3 and 4 are repeated

4.2.1 Preprocessing

The preprocessing module mainly aims at extracting seed clues in a given domain. We started with a publicly available resource for SA, SentiWordNet [2], which is widely used and publicly available for research purpose.

Table 1. SentiWordNet: "unpredictable"

POS	pos.	neg.	word	sense
ADJ	0.0	0.625	unpredictable#a	#1
ADJ	0.0	0.0	unpredictable#a	#2
ADJ	0.0	0.25	unpredictable#a	#3

Each sense in a synset is POS-tagged and weighted with a number in [0, 1] as negative or positive, as shown in Table 1. There are 45,864 senses in this lexicon.

In order to generate seed clues with their polarity, we need to map the sentiment clue candidates in the training examples to SentiWordNet. We first randomly select a small number of seed sentences from D_{qt}, whose polarity values are known, to generate training examples. A verb, adjective, or noun unigram in the seed sentences becomes a seed clue if its highest (or lowest) sentiment weight in the synset is greater (or less) than 0.7 (or -0.7). If the weights of a word are greater than 0.7 and less than -0.7 at the same time, it is considered ambiguous and excluded from the seed clue set. For later processing, verbs and nouns are normalized to their base forms in WordNet[1].

4.2.2 Sentiment Topic Identification

Sentiment topics are extracted automatically from the training examples by using the current clue set (seed clues for the first iteration). The extraction algorithm is based on our observation that a sentiment topic is strongly connected to sentiment clues when a sentiment topic and a clue have a syntactic dependency in a sentiment-revealing sentence and when they co-occur in the domain corpus. For example, a sentiment topic "Yasukuni Shrine" often co-occurs with "resent" in negative sentiment sentences.

A list of sentiment topic candidates are first generated using the dependency relations involving the seed clues, which are found by a dependency parser. For the current work, we use Stanford Statistical Parser [6]. An example for an output of the parser is shown in Fig. 1. In this example, "the opinion", "Professor Smith", and "the opinion of Professor Smith" are the candidate sentiment topics because each of them is dependent on the verb "criticize" (a sentiment clue). While "the opinion" and "the opinion of Professor Smith" are directly dependent on the clue, "Professor Smith" is indirectly dependent on it (i.e., "criticize"➔ "opinion" ➔ "of" ➔ "Smith"). For an adjective clue, we reverse the dependency relation because an adjective would be dependent on a noun phrase. As a result, we can create a candidate list from each training example, which includes noun phrases dependent on an adjective clue, a governing verb, or a noun clue.

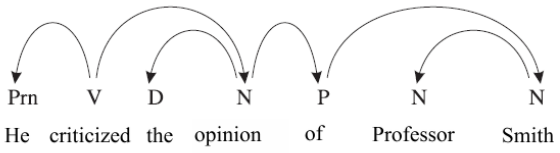

Figure 1. Dependency Parsing Example

The next step is to calculate the score of each sentiment topic candidate based on co-occurrence information and pick the highest ranked candidate as the sentiment topic. We first compute the contextual similarity between a noun phrase (i.e., candidate), *NP*, and the current clue set, *C*, based on the co-occurrence information as follows:

$$csim(NP, C) = \frac{\sum_{i=1}^{n} np_i \times \forall c_i}{\sqrt{\sum_{i=1}^{n} np_i^2 \times \sum_{i=1}^{n} \forall c_i^2}}$$

where np_i represents the occurrence (0 or 1) of a *NP* in the *i*-th sentence from the document set of a query topic (D_{qt}), and $\forall c_i$ represents the binary occurrence of any clue from *C* in *i*-th sentence of D_{qt}. The more frequently an *NP* co-occurs with any clue in *C*, the higher its score.

Similarly, we compute the contextual similarity between a candidate *NP* and the query topic that contributed to the generation of the domain corpus. Since sentiment topics are supposed to be related to the domain, the contextual relationship between a candidate *NP* and the query should provide useful

[1] Lexical database http://wordnet.princeton.edu/

information in judging its appropriateness. Using the same formula but with individual query words *W*, we compute *csim* (*NP*, *W*) as follows.

$$csim(NP,W) = \frac{\sum_{i=1}^{n} np_i \times w_i}{\sqrt{\sum_{i=1}^{n} np_i^2 \times \sum_{i=1}^{n} w_i^2}}$$

where np_i and w_i represent the occurrence (0 or 1) of a *NP* and a *W* in the *i*-th sentence from D_{qt}. From this formula, an *NP* that co-occurs with query words gets a high score.

Combining the two contextual similarity values, we score each sentiment topic candidate as follows:

$$scr(NP) = \lambda \cdot csim(NP,C) + (1-\lambda) \sum_{\forall W \in qt} \mu_W \cdot csim(NP,W)$$

where μ_W is the weight of a keyword in the query, and λ is empirically set to 0.5 (see Section 4.1). Besides, a query topic consists of several query words (e.g., "I would like to know about El Nino."). In order to reduce the negative effect from less important query words (e.g., "I", "would", "like"), we adopt a term weighting scheme. For query term weights, we employ a TF/ISF scoring in D_{qt}. where TF and ISF stand for term frequency and inverse sentence frequency, respectively. As in [5], inverse document frequency (IDF) is strongly correlated to ISF. As a result of this step, we can extract sentiment topics in training examples.

4.2.3 Sentiment Clue Generation

This step includes two tasks: gathering new sentiment clue candidates and determining the polarity and accepting/rejecting each candidate as a new clue. We assume that in the same domain, the polarity of a clue is fixed and that different sentiment clues have the same polarity if they are connected to a common sentiment topic in D_{qt}.

Since the amount of data may not be sufficient within D_{qt}, (some *qt* contains only several relevant documents), however, we apply our generation scheme to the domain collection D_{dm} and generate a set of sentiment clue candidates for each sentiment topic in D_{dm}. A new sentiment clue candidate is generated when 1) it is an adjective that has a dependency on a sentiment topic or 2) it is a verb or noun that governs a sentiment topic.

For a sentiment topic, *st*, we first generate a set of linked words: $LW(st) = \{w | st \sim w\}$where $st \sim w$ represents that *st* and *w* (a noun, verb, or an adjective) have the syntactic dependency mentioned above. Since the candidates may or may not exist in the current clue set, we divide them into two subsets:

1) $KW(st)$ for the linked words whose polarity values are known already since they are already in the current seed set
2) $UW(st)$ for the linked words whose polarity values are unknown because they are new sentiment clue candidates

Using these two sets, we eliminate some of the sentiment topics if they satisfy any of the following conditions.

1) $KW(st) = \emptyset$ or $UW(st) = \emptyset$
2) The difference between the numbers of positive and negative clues in $KW(st)$ is not sufficiently high.

The difference is measured as follows:

$$diff(st) = \frac{\left||POS(st)| - |NEG(st)|\right|}{|KW(st)|}$$

where $POS(st)$ and $NEG(st)$ represent the sets of positive and negative clues in $KW(st)$, respectively. A sentiment topic *st* is removed if $diff(st)$ is lower than an empirically set threshold 0.3.

Given a set of clue candidates generated from the surviving sentiment topics, the next step is to determine whether individual clues should be accepted or rejected. For each sentiment topic, *st*, we first calculate the sentiment score of each candidate, c_{new} in $UW(st)$ as follows:

$$senti\_scr(c_{new}) = \frac{\sum_{i=1}^{|KW(st)|} \pi^i \cdot senti\_scr(c_{old}^i)}{|KW(st)|}$$

where c_{old}^i is *i*-th known clue in $KW(st)$ and π^i is the weight of c_{old}^i towards c_{new}. We assume that the sentiment score of c_{new} can be computed as the weighted average of all the connected known clues (i.e.,$\forall c_{old}$) and the weight π^i should reflect the degree to which the old and new clues occurs together in the domain corpus, which is computed as:

$$\pi^i = \frac{p(c_{old}^i, c_{new} | D_{dm})}{p(c_{new} | D_{dm})} = \frac{freq(c_{old}^i, c_{new} | D_{dm})}{freq(c_{new} | D_{dm})}$$

where $freq(c_{old}^i, c_{new}|D_{dm})$ is the co-occurrence frequency of words c_{old}^i and c_{new} in D_{dm}.

After scoring, we examine further to determine whether the candidate clue is contextually acceptable for a particular polarity. A candidate is deemed to have the correct polarity if it frequently co-occurs with the clues of the same polarity in the sentences containing the candidate. We compute the probabilities of a candidate co-occurring with positive clues and negative clues, respectively, by using a unigram language model and compare them. The *inspect* score for c_{new} is calculated using the language model $\theta_{c_{new}}$ as follows:

$$inspect(c_{new}) = \frac{p(C_{pos} | \theta_{c_{new}})}{p(C_{neg} | \theta_{c_{new}})}$$

where C_{pos} and C_{neg} are the sets of positive and negative clues in $\theta_{c_{new}}$, respectively. The log-likelihood of the score is obtained with:

$$\left| \ln \sum_{i=1}^{|C_{pos}|} freq(c_{pos}^i | \theta_{c_{new}}) - \ln \sum_{j=1}^{|C_{neg}|} freq(c_{neg}^j | \theta_{c_{new}}) \right|$$

where $freq(c_{pos}^i|\theta_{c_{new}})$ represents the frequency of *i*-th positive old clue in the set of sentences containing c_{new}. The absolute value of the likelihood ratio approaches to 0 if c_{new} is ambiguously co-occurring with any of positive and negative clues, and c_{new} in this case is unacceptable. If the value is much higher than 0, c_{new} is acceptable to be added into the current *C* as a new clue. As a result of this step, we use the sentiment score of each new clue (i.e., $senti\_scr(c_{new})$) and update the current clue set.

4.2.4 Learning a Domain-specific Classifier

As in the bootstrapping algorithm, we need to accrue new clues based on the sentiment topics iteratively. After the first iteration

using the initial training examples and the clue set, we need additional training examples from the domain corpus so that new sentiment topics are identified and new clues extracted. To do this, we need to learn a sentiment classifier using the current clue set and generate additional positive and negative sentences.

We employ a k-means clustering algorithm [7] to identify positive or negative sentences from the domain corpus. It is not appropriate to use a supervised machine learning based classifier because there are not a sufficient number of training examples for the expanded clue set. We first generate the feature space consisting of currently known clues and assign weights to them by their sentiment scores. Each sentence in D_{dm} can be expressed as a feature vector. After constructing three centroids for positive, negative, and neutral sentiments with the training examples, we apply the k-means clustering algorithm to the remaining sentences whose polarity values have not been determined yet in the corpus. Since the initial centroids are constructed with a relatively small set of polarity-known examples, the process is repeated until it reaches a stable state. The clustering procedure is as follows:

Step 1. Based on the current training examples, three initial means (i.e., centroids) are computed as follows:

$$\vec{c}_j = \frac{1}{|C_j|} \sum_{\vec{S} \in C_j} \frac{\vec{S}}{\|\vec{S}\|}$$

where $\vec{S}$ is a training example which belongs to C_j,

$$C_j = \{POS, NEG, NEU\}$$

Step 2. Three clusters are created by associating new examples to the nearest mean.

Step 3. A new mean is computed for each cluster created from Step 2. Steps 2 and 3 are repeated until the clustering reaches a stable state.

When the clustering is done, all the examples in D_{dm} are classified into the positive, negative, or neutral category. We pick the top five examples (i.e., the five nearest to the centroid), except the training examples, for each cluster and update the training example set for the next iteration of the bootstrapping algorithm.

5. EXPERIMENT

In a series of experiments, we verify the effectiveness of domain-specific clues and classifiers generated by our bootstrapping algorithm. Before the evaluation, we optimize the proposed model for sentiment topic identification, which is an integral part of the bootstrapping process, and then evaluate the bootstrapping algorithm in the main experiment. To how good the domain-specific classifier was built, we tested a sentiment classifier whose features are from the clue set. In the first set of experiments, we measured sentiment classification effectiveness at each iteration step of the bootstrapping process to see how much can be gained by introducing a new set of clues incrementally. In the next experiment, we compared the contextually aggregated features with general purpose clues by using supervised learning framework. More specifically, we implemented SVM based sentiment classifier featured by our domain-specific clues and SentiWordNet words, and tested on the same corpus.

5.1 Experimental Set-up

We developed a domain corpus by utilizing the collections from NTCIR-6 Opinion Analysis Pilot Task (Seki et al., 2007) and NTCIR-7 Multilingual Opinion Analysis Task (Seki et al., 2008). Overall, NTCIR-6 and NTCIR-7 test collections contain 30 topics (490 related documents) and 17 topics (167 relevant news articles), respectively. The result of the domain corpus consists of 45 query topics and 606 relevant news documents (12,840 sentences). The query topics were manually grouped to form four domains as shown in Table 2.

To build a gold standard for sentence-level sentiment classification, each sentence was tagged as positive, negative, or neutral. We used the annotation results provided by the NTCIR test collection (Seki et al., 2007, 2008) for which three annotators were hired, and resolved the conflicts, if any, with majority voting. Table 3 presents the gold-standards for NTCIR-6 & 7 collections, and Table 3 describes detailed information about the gold standard. Since many topics in the NTCIR collection are relevant to BIZ and INTL domains, the numbers of the sentences in both domains are much greater than those in the other two.

Table 2. Query Topic Grouping

Domain	Query Topic Examples	# Topics
Business (BIZ)	How Asian are countries dealing with Asian economic crisis?	9
International event (INTL)	Find reports on G8 Okinawa Summit.	12
Environment (ENV)	What is the Green House Effect?	7
Politics (POL)	Find articles about sexual harassment of Clinton and Lewinski.	6

Table 3. Gold-standard

Domain	Positive	Negative	Neutral
BIZ	222 (7.0%)	481 (15.2%)	2,472 (77.9%)
INTL	195 (6.3%)	634 (20.5%)	2,267 (73.2%)
ENV	63 (5.2%)	234 (19.3%)	913 (75.5%)
POL	44 (3.7%)	188 (15.6%)	971 (80.7%)

In addition, we used NTCIR-7 collection to develop the gold-standard for sentiment topic identification. Three annotators select sentiment topics for each polarized sentence (i.e., positively or negatively polarized sentence, and the polarity determination is based on the three annotators, as we specified above), and the confliction was resolved by the majority voting. So, like participants in NTCIR-7 Multilingual Opinion Analysis Task (Seki et al., 2008), our system extracts sentiment topics for each polarized sentence, and the polarity judgment is primarily informed by the above gold-standard in this task. Also, by using the model introduced in Section 3.2.2, we extract the NPs whose scores are over threshold (empirically set as 0.75) as sentiment topics of each polarized sentence. Besides, in this experiment, we should resolve string matching problem because sentiment topic is a word-sequence (i.e., phrase) and strict mapping between the system result and the human annotation is somewhat unreasonable, e.g., "the president, Bush" is the same as "Bush" or "president" in the same context. Thus, we leniently accept the system result as correct if 1) the system result and the annotation answer are exactly matched; 2) the head word of the system result and that of the answer are precisely matched. Based on this criterion, 1,376 sentiment topics are annotated in 867 polarized sentences from

NTCIR-7 corpus, and we measured the precision and recall performances on sentiment topic extraction.

5.1.1 Sentiment Topic Identification Experiment

We ran a preliminary experiment for sentiment topic identification model, introduced in Section 4.2.2, on a partial data set of the NTCIR-7 test collection [20]. Since the identification model needs domain-specific clues and the bootstrapping algorithm should be run on a domain corpus, our experiment was limited to a single domain (INTL). The partial data set contains 468 polarized sentences (i.e., positive and negative), and 653 target phrases are annotated within those sentences (target annotation task was only proposed in NTCIR-7). We assume that the target phrase would be a sentiment topic of each positive or negative sentence, and we conduct a preliminary experiment to optimize the weight value (i.e., λ) in sentiment topic identification. Since the value of λ is [0,1], we initially ran the bootstrapping algorithm with λ=0.1, and we gradually increase the value by 0.1 to 0.9.

As a result, we obtained λ=0.1 as an optimal value where the identification model achieved the highest accuracy of 0.778 in F-Measure (0.714 in Precision and 0.853 in Recall), which means that two different contextual similarities are equally important; in the model, we combine the similarities between sentiment topics and sentiment clues, and query topic words (see Section 4.2.2).

5.2 Experimental Results

For comparisons, we measured precision and recall performance of the sentiment classifier under different conditions. Since the performance is biased by the size of initial seeds, we used three different cases, 15, 30, and 45 examples, together with the seed clues obtained from SentiWordNet.

5.2.1 Convergence Rates

Before discussing the effects of extracting sentiment clues on sentiment classification for different numbers of iterations, we first show the number of sentiment clues generated as we repeat the bootstrapping process until it reaches a plateau.

Figure 2 depicts the curves for twelve different cases covering the three different cases of the numbers of seed clues and the four domains. It shows that the number of iterations to reach a plateau varies depending on the sizes of the domain corpora and the sizes of the seed sets. For example, BIZ@15 (the case of using 15 seed clues for the BIZ domain) and INTL@15 require 18 iterations whereas ENV@15 and POL@15 reached the plateaus after 14 iterations. This result comes from the fact that BIZ and INTL include more data with which more sentiment topics and clues can be generated with additional iterations. A similar trend is observed with the different seed sizes. Every 45-seed case obtains more clues in much shorter time than other cases with a smaller seed size. Starting with a larger number of seeds, a larger number of sentiment topics and clues are generated for the entire corpus. The differences in the slops indicate that in order to generate a maximal number of clues for a domain, it is more important to have a large size corpus than start with a large size of a seed set.

5.2.2 Bootstrapping Performance

For fair comparisons, we used the same numbers of seed clues and training examples for the three different sentiment classes across all the 12 cases (four domains and three seed numbers). The numbers in the y-axis of Fig. 3 and 4 are the F-measure values for classification performances on both positive and negative sentiment cases (separated in Fig. 3 for BIZ and INTL domains and Fig. 4 for ENV and POL domains for readability).

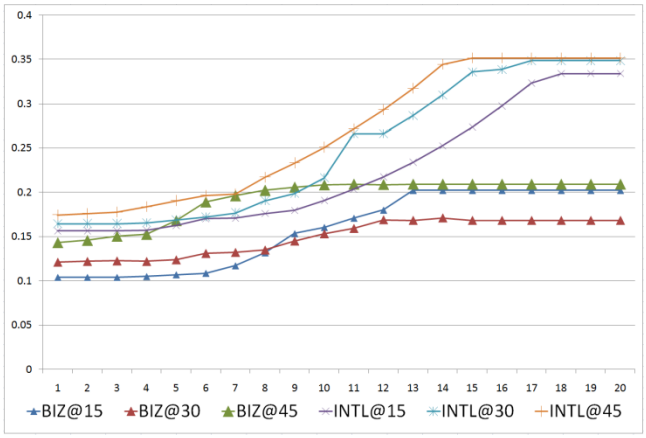

Figure 2. F-Measure of Polarity Classification in BIZ and INTL domains

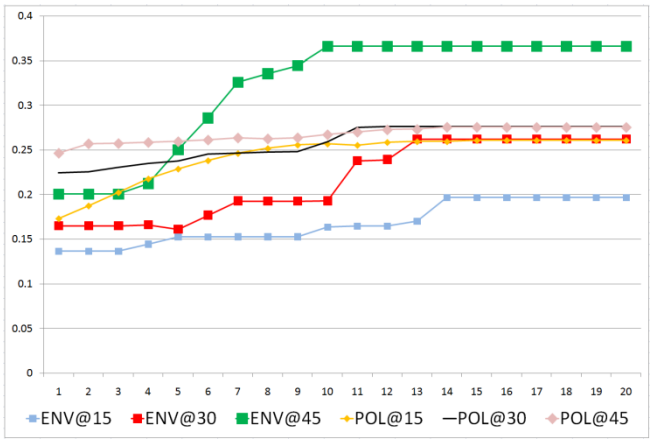

Figure 3. F-Measure of Polarity Classification in ENV and POL domains

By comparing the convergence and classification results (i.e., Fig. 2 and Figures 3 and 4), we can see that the performance improvements generally follow the increments of the new clues after iterations regardless of the domains and the seed set sizes although the rates all differ. For example, when the number of clues reaches the plateau, the improvement in effectiveness also stops. For each domain, the larger the number of seeds is, the higher the overall performance. While this confirms that the number of training examples affects the quality of the learning process, the performance after sufficient number of iterations, even with a relatively smaller number of seeds, can be comparable with those of larger numbers of seeds with the proposed bootstrapping algorithm (see the case for POL).

We can observe that there is no performance drop on the way to the maximum number of iterations in every case. This is strong evidence that the additional clues generated by the bootstrapping algorithm do not hurt sentiment classification because the sentiment topic identification and new clue selection were done rather conservatively.

An interesting but somewhat unexpected result is that the size of the domain corpus did not affect the performance increases. This

also indicates that the rapid increases in the number of new clues did not affect the effectiveness at the same speed. While the performance increases in the cases of the BIZ domain with a large collection are very low, the same for the ENV domain with a small collection is much stiffer. We conjecture that SA in the BIZ domain is more difficult than in other domains. Similarly, The ENV domain is easier for SA, especially with a sufficient number of clues.

5.2.3 Comparison with general Clues in Supervised Learning

In order to examine our hypothesys that contextually-driven features considering sentiment topic would enhance the existing keyword spotting based system in SA, we compare the performances of sentiment classifiers featured by context features and generally used clues in SA. Specifically, in this experiment, we used SVM-based sentiment classifier in our domain-corpus. We employed 5-fold cross validation on each domain and trained SVM classifier featured as SentiWordNet (SentiWN) and the generated domain-specific clues (OUR). Thus, we verify the effectiveness of the domain-specific clues by comparing the precision and recall performances of two cases for positive and negative sentences.

Table 4. Comparison against SentiWordNet clues in SVM classifier

Dm	System	Precision	Recall	F-Measure
BIZ	SentiWN	0.712	0.551	0.621
	OUR	0.845 (*+18.7*%)	0.717 (*+30.1*%)	0.776 **(+24.9%)**
INTL	SentiWN	0.735	0.635	0.681
	OUR	0.798 (*+8.6*%)	0.874 (*+37.6*%)	0.834 **(+22.4%)**
ENV	SentiWN	0.651	0.377	0.477
	OUR	0.663 (*+1.9*%)	0.590 (*+56.5*%)	0.624 **(+30.8%)**
POL	SentiWN	0.739	0.299	0.426
	OUR	0.617 (*-16.5*%)	0.646 (*+116.*1%)	0.631 **(+48.3%)**

As Table 4 shows, overall performances with domain-specific clues are superior to those without the domain clues. Particularly, recall performances on all domains are significantly improved (In POL, the enhancement is over-doubled). Besides, the experiments on ENV and POL show that the case of relatively low performance with general clues could be highly probable to enhance by the bootstrapping performance. However, the performances in less-data domains (i.e., ENV and POL) are much lower than those of sufficient domains (i.e., BIZ and INTL) in both SentiWN and OUR cases.

5.2.4 Discussion

Our qualitative analysis shows that our system can extract many reasonable domain-specific clues and sentiment topics based on a small set of seed clues as in Table 5.

Table 5. A Sample Result for Sentiment Clues and Topics

	Business Domain
Seed Clues	crisis, problem, depression, panic, good, hope
Sentiment Topics	Thai, Asia, East Asia, Risk Management, International Monetary Fund, IMF, GDP, Asian Development Bank, World Bank, AOL, Stock, Korea, U.S.
Expanded Clues	inflation, precipitation, tight, consultant, vie, persist, deficit, cry, shrink, sharp, useful, corporate, difficult, unscathed, slow, keen, fight, rise, lost, rapid, little, threat, blasé, confident, weaken

Initially, a small set of seed clues such as “crisis”, “panic”, and “depression” were extracted by SentiWordNet. Then sentiment topics were identified like “Thai”, “Asia”, and “World Bank” which are linked to the seed clues. At the final step, domain-specific clues such as “precipitation”, “vie”, and “persist” were extracted as negative clues. More specifically, “persist” can convey a negative sentiment in its context, since it came from “The financial crisis on Asian countries would persist.” (“persist” is a negative clue as long as it co-occurs with “Asian countries” in the Business domain).

Some erroneous results were found, too. We observe that there are more errors in positive than negative clues. This is due to the fact that the sentiment topics in negative sentences are more reliable and reusable in identifying negative clues. Newspaper articles tend to report on more sensational news containing negative stories. In addition, sentiment topics in positive examples are more diverse in their forms, and the sheer number of positive topics is smaller with more polarity conflicts identified in positive sentences. As a result, the number of overt positive sentences with short and clear clue words is small. Besides, the clues bearing positive-sentiment are not always close to the sentiment topic, making harder to identify them.

A common source of errors for both of the classes is the unit of sentiment clues. Since we viewed a clue as a unigram, our system could not capture phrase-level clues. Nevertheless, the ability to identify the head word of a clue phrase was helpful because such head words often occur frequently in the right sentiment contexts.

6. CONCLUSION

In this paper, we proposed a domain-specific sentiment analysis system utilizing context features in news texts. The system includes two main modules: context feature generation and domain-specific sentiment classifier learning. In order to generate domain-context classifier using context features, we utilize the bootstrapping method exploiting syntactic dependency and co-occurrence between a sentiment topic and a contextual clue. The proposed method is semi-supervised, and the evaluation shows that our method is quite successful in extracting contextual clues in news domains and hence in enhancing sentiment classification performance. As a future work, we plan to expand the domain corpora by adding more examples into insufficient-data domains. As discussed in Section 5.2.4, we also need to expand the unit of clues from a word to a phrase.

7. ACKNOWLEDGMENTS

This work was financially supported by Microsoft Research Asia, the grant from the strategic technology development problem 2008-F-047-02 of the MKE, and the MKE(The Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) support program supervised by the NIPA(National It Industry Promotion Agency" (NIPA-2009-(C1090-0903-0008)).

8. REFERENCES

[1] A. Esuli and F. Sebastiani. Determining the semantic orientation of terms through gloss classification. In *Proceedings of 14th ACM International Conference on Information and Knowledge Management (CIKM)*, 2005, pages 617-624.

[2] A. Esuli and F. Sebastiani. SentiWordNet: a publicly available lexical resource for opinion mining. In *Proceedings of 5th International Conference on Language Resources and Evaluation (LREC)*, 2006, pages 417-422.

[3] A. Aue and M. Gamon. Customizing sentiment classifiers to new domains: a case study. In *Proceedings of 4th Recent Advances in NLP (RANLP)*, 2005.

[4] B. Pang and L. Lee. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval 2(1-2)*, 2008, pages 1-135.

[5] C. Blake. A comparison of document, sentence, and term event spaces. In *Proceedings of 44th Annual Meeting of Association for Computational Linguistics (ACL)*, 2006, pages 601-608.

[6] D. Klein and C. Manning. Accurate unlexicalized parsing. In *Proceedings of 41st Annual Meeting of Association for Computational Linguistics (ACL)*, 2003, pages 423-430.

[7] G. Frahling and C. Sohler. A fast k-means implementation using corsets. In *Proceedings of 22nd Annual Symposium on Computational Geometry*, 2006, pages 135-143.

[8] H. Kanayama and T. Nasukawa. Fully automatic lexicon expansion for domain-oriented sentiment analysis. In *Proceedings of the conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2006, pages 355-363.

[9] K. Dave, S. Lawrence, and D. Pennock. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of 12th International Conference on World Wide Web (WWW)*, 2003, pages 519-528.

[10] N. Kando. Overview of the Sixth NTCIR Workshop. In *Proceedings of 6th NTCIR Evaluation Workshop*, 2007, pages 1-9.

[11] N. Kando. Overview of the Seventh NTCIR Workshop. In *Proceedings of 7th NTCIR Evaluation Workshop*, 2008, pages 1-9.

[12] P. Turney. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the Association for Computational Linguistics (ACL)*, 2002, pages 417-424.

[13] P. Turney and M. Littman. 2003. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems*, 21(4), 2003, pages 315-346.

[14] T. Wilson, J. Wiebe, and P. Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT/EMNLP)*, 2005, pages 347-354

[15] V. Stoyanov and C. Cardie. Annotating topics of opinions. In *Proceedings of 6th International Conference on Language Resources and Evaluation (LREC)*, 2008, pages 3213-3217.

[16] X. Ding and B. Liu. The utility of linguistic rules in opinion mining. In *Proceedings of 30th annual ACM conference on Research and Development in Information Retrieval (SIGIR)*, 2007, pages 811-812.

[17] Y. Kim and S. Myaeng. Opinion analysis based on lexical clues and their expansion. In *Proceedings of 6th NTCIR Evaluation Workshop*, 2007.

[18] Y. Ko. and J. Seo. Learning with unlabeled data for text categorization using bootstrapping and feature projection techniques. In *Proceedings of Proceedings of 42nd Annual Meeting of Association for Computational Linguistics (ACL)*, 2004, pages 255-262.

[19] Y. Seki, D. Evans, L. Ku, H. Chen, N. Kando, and C. Lin. Overview of opinion analysis pilot task at NTCIR-6. In *Proceedings of 6th NTCIR Evaluation Workshop*, 2007.

[20] Y. Seki, D. Evans, L. Ku, L. Sun, H. Chen, and N. Kando. Overview of Multilingual Opinion Analysis Task at NTCIR-7. In *Proceedings of 7th NTCIR Evaluation Workshop*, 2008, pages 185-203.

[21] V. Hatzivassiloglous and K. R. McKeown. Predicting the semantic orientation of adjectives. In *Proceedings of 35th Annual Meeting of Association for Computational Linguistics (ACL)*, 1997, pages 174-181.

[22] P. Beineke, T. Hastie, and S. Vaithyanathan. The sentiment factor: Improving review classification via human provided information. In *Proceedings of 42nd Annual Meeting of Association for Computational Linguistics (ACL)*, 2004, pages 263-270.

[23] H. Yu and V. Hatzivassiloglou. 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)*, 2003, pages 129-136.

[24] S. Kim and E. Hovy. 2004. Determining the sentiment of opinions. In *Proceedings of Conference on Computational Linguistics (COLING)*.