# Generating Domain-Specific Clues using News Corpus for Sentiment Classification

**Youngho Kim[§], Yoonjung Choi[+], Sung-Hyon Myaeng[*]**

[§]Department of Computer Science, University of Massachusetts Amherst, 140 Governors Drive Amherst, MA, US
[+*]KAIST, 335 Gwahak ro Yuseong gu, Daejeon, South Korea

**[§]yhkim@cs.umass.edu, [+]choiyj35@kaist.ac.kr, [*]myaeng@kaist.ac.kr**

## Abstract

The domain dependent nature of sentiment analysis calls for domain specific sentiment clues. This paper addresses the problem of automatically generating such domain specific sentiment clues and proposes an effective solution. The main idea is to bootstrap from a small seed set and generate new clues by using syntactic dependency and collocation information between sentiment clues and sentence level topics that are defined to be a primary subject of a sentiment expression (e.g., event, company, and person). Our experiments show that the automatically extracted clues are effective for sentiment classification.

## 1. Introduction

Sentiment Analysis (SA) is concerned with identifying positive or negative opinions, emotions, and attitudes expressed in free texts (Pang and Lee 2008). The popularity of this field has been largely driven by intriguing applications in areas like product reviews and political polls (Turney 2002). In a general form, a sentiment expression is characterized by its polarity (i.e. positive or negative), often with its topic (or target) and source (or sentiment holder).

Some past studies proposed to expand a set of sentiment clues (e.g., "great" and "hate") which play an important role in SA (Turney 2002; Esuli and Sebastiani 2006). However, they focus on a general purpose sentiment lexicon without addressing domain specificity and it's been shown that a sentiment classifier trained for one domain does not perform well in others (Aue and Gamon 2005). A clue or a feature learned and used by a classifier may not be found in a new domain, or key clues in a new domain may not exist in an old domain. Even if a clue exists in both domains, it may bear different polarity values in two different domains. For example, "unpredictable" is positive in a movie plot review but negative for a car's functionality.

To alleviate this problem, sentiment clues need to be associated with contextual information such as topics, expressions, and domains. For example, "unpredictable" should be treated as a negative 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". This calls for development of a domain-specific lexicon using contextual information, which would be more effective in discerning the polarity of a sentiment expression than a general-purpose lexicon.

This paper presents a novel method for extracting domain-specific sentiment clues from a news corpus, which employs 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 is novel in that it exploits a syntactic dependency between a 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", provided that "criticize" is not already in the clue lexicon. Once a topic for a known sentiment clue is identified, it helps a new clue be detected when they co-occur in the same domain. In other words, we discover a new clue if it is associated with an old clue through 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 sentences, which are in turn used for the next iteration of clue extraction: identifying new topics and new clue candidates. This process stops when no more clues are added. We expect the domain-specific features would be more discriminative than those collected for general purposes.

Riloff et al. (2003) introduced the bootstrapping process for populating a general purpose lexicon, but they focused on learning subjective nouns (not specifically determine positive or negative polarity). Our previous work (Choi et al. 2009) describes a domain-specific sentiment classifier where the contextual polarity of a sentiment clue is derived by the same method described in this paper. This paper summarizes the method and focuses on the effectiveness of domain-specific sentiment clues by showing a comparison against the state-of-the-art approach as well as a further analysis of the convergence rate of the method.

## 2. Sentiment Clue Generation Method

We use a domain corpus, which contains relevant documents to a domain (details in Section 3). The bootstrapping algorithm consists of four steps. After step 1, steps 2, 3, and 4 are repeated until no new clues are introduced.

**Step 1 (Preprocessing).** This step aims at extracting seed clues in a given domain by using SentiWordNet (SentiWN) (Esulit and Sebastiani 2006). We first randomly select a small number of seed sentences from a domain, whose polarity values are known (training examples). A verb, adjective, or noun unigram in the seed sentences becomes a seed clue when it is found in SentiWN with a sufficiently high weight. For later processing, verbs and nouns are normalized to their base forms in WordNet[1].

**Step 2 (Topic Identification).** Topics in a training example are extracted by using the current clue set (seed clues from Step 1). We observe that a topic in a sentiment sentence is strongly connected to a sentiment clue when they have a syntactic dependency in a sentiment-revealing sentence and when they co-occur in the domain corpus. A list of topic candidates is identified by using the dependency relations involving the seed clues, which are found by a dependency parser[2]. The score of each topic candidate is calculated by its co-occurrence with the sentiment clues, and we pick the top ranked candidate. For measuring co-occurrence, we compute the similarity between a noun phrase (candidate), *NP*, and the clue set, *C*.

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

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

Similarly, we compute the similarity between *NP* and the query that contributed to the generation of the domain corpus (details in Section 3) since appropriate topics must be related to the domain. Using the same formula but with an individual query word *W*, we compute *sim (NP, W)* as follows.

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

where $np_i$ and $w_i$ represent the occurrence (0 or 1) of a *NP* and a *W* in the *i*-th sentence in *D*. From this formula, an *NP* that co-occurs with query words gets a high score. Combining the two similarity values, we score each topic candidate as follows:

$$scr(NP)=\lambda \cdot sim\left(NP,C\right)+(1-\lambda)\sum_{W\in query} sim\left(NP,W\right)$$

where λ is empirically set to 0.5. To ensure that a candidate is sentiment-related, we test whether there is a polarity-possessing clue having a syntactic dependency with the topic in the corpus.

**Step 3 (Sentiment Clue Generation).** This step includes two tasks: gathering new sentiment clue candidates and accepting/rejecting each candidate as a new clue after determining its polarity. A new sentiment clue candidate is generated when it is an adjective that has a dependency on a topic, or a verb or noun that governs a topic. For each clue candidate $c_{new}$, we calculate its sentiment score:

$$senti\ scr(c_{new})=\frac{\sum_{i=1}^{|C_{old}|} \pi^i \cdot senti\ scr\left(c_{old}^i\right)}{\left|C_{old}\right|}$$

where $c^i_{old}$ is *i*-th known clue for the topic through a dependency relation and $\pi^i$ is the weight of $c^i_{old}$ 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 $\pi^i$ should reflect the degree to which the old and new clues occur together in the domain, which is computed as:

$$\pi^i=\frac{p\left(c_{old}^i,c_{new}\right)}{p\left(c_{new}\right)}=\frac{freq\left(c_{old}^i,c_{new}\right)}{freq\left(c_{new}\right)}$$

where $freq(c^i_{old},\ c_{new})$ is the co-occurrence frequency of words $c^i_{old}$ and $c_{new}$ in *D*.

After scoring, we determine whether the candidate clue is acceptable for the calculated polarity. A candidate is deemed to have the correct polarity if there are sufficient number of sentences containing the candidate and other clues of the same polarity. We compute the probabilities of a candidate co-occurring with positive clues and negative clues, respectively, by using a language model (unigram term distribution) and compare them. The *inspect* score for a new clue *c* to have a fixed polarity value is calculated using the language model *θ*:

$$inspect(c)=\frac{p\left(C_{pos}\middle|\theta_c\right)}{p\left(C_{neg}\middle|\theta_c\right)}$$

where $C_{pos}$ and $C_{neg}$ are the sets of positive and negative clues in $\theta_c$, respectively. The language model probability $p(\cdot|\theta_c)$ is estimated by counting the frequency of sentences containing positive (or negative) words and $c_{,}$ together. If the difference ratio is much higher than 0, *c* is acceptable to be added into the current clue set as a new clue.

**Step 4 (Generating Additional Training Examples).** After generating new clues with the initial seed examples, we generate additional training examples (positive and negative sentences) by means of an unsupervised clustering learning algorithm using the current clues. Each sentence in the domain documents is represented as a vector consisting of current clues and their weights (sentiment scores). After constructing three initial centroids for positive, negative, and neutral clusters with the training examples, the sentences are clustered with a k-means clustering algorithm. The resulting sentences are

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

[2] Stanford Parser (http://nlp.stanford.edu/software/)

used as training examples for the next iteration of the clue generation method.

## 3. Experiment

We ran experiments to show effectiveness of contextual clues generated by our bootstrapping algorithm. Instead of making subjective judgments of their quality, we opted for a practical method - building and testing a sentiment classifier whose features are constructed from the clue set.

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). In this test collection, there are 45 queries and 12,840 relevant sentences tagged as positive, negative, or neutral. We grouped relevant queries for four domains (Business, International Event, Environment, and Politics). The number of sentences in each domain is described in Table 1.

**Table 1. Sentence Statistics for Four Domains**

| 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%) |

For comparisons, we calculated F-measure of the sentiment classifier under different initial clues. We tested with three different seed cases (15, 30, 45 sentences).

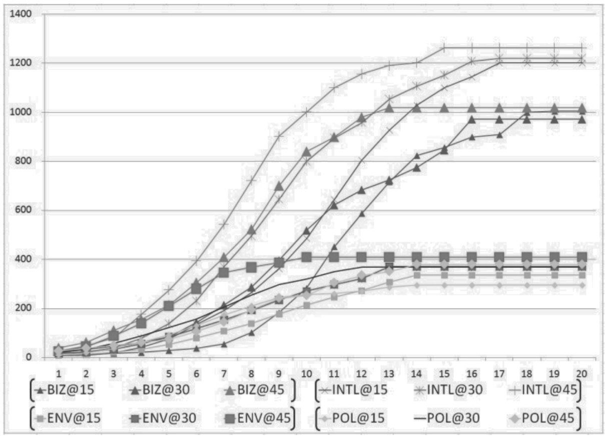

**Figure 1. Number of Iterations for Convergence**

**Convergence Rates.** We show the number of sentiment clues generated as we repeat the bootstrapping process until it reaches a plateau. Figure 1 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 (using 15 seed clues for BIZ) 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. 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, it is more important to have a large size corpus than start with a large seed set.

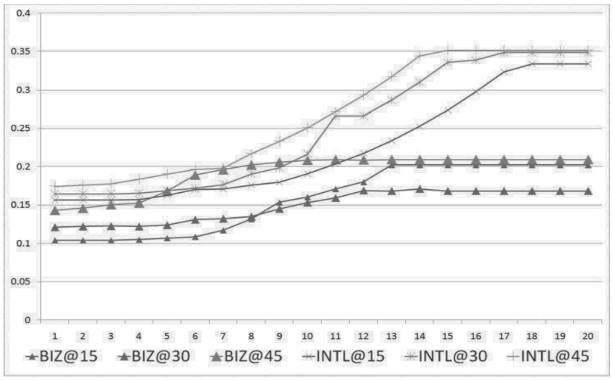

**Figure 2. Polarity Classification in BIZ and INTL**

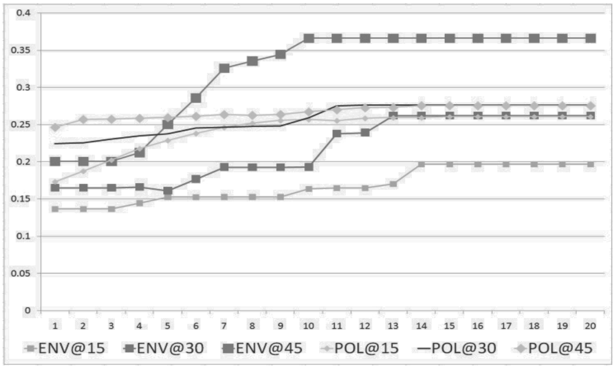

**Figure 3. Polarity Classification in ENV and POL**

**Bootstrapping Performance.** For fair comparisons, we used the same numbers of seed clues and training examples for the sentiment classes across all the 12 cases (4 domains and 3 seed numbers). The numbers in the y-axis of Fig. 2 and 3 are the F-measure values for classification performances on both positive and negative sentiment cases (BIZ and INTL in Fig. 3 and ENV and POL Fig. 4 for readability). By comparing Fig. 2 and Fig. 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 each domain, the larger the number of seeds is, the higher the overall performance. 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.

**Comparison with another Method.** Our proposed method was compared against a previous state-of-the-art approach for context-dependent clue generation (Ding and Liu 2007). The algorithm is based on three linguistic rules:

1) Intra-sentence conjunction (e.g. With "it's *great* and has a *long* battery life." *long* is obtained as positive).
2) Pseudo intra-sentence conjunction (e.g., "it has a *long* battery life, which is *great*.")
3) Inter-sentence conjunction (e.g. "It's *amazing*. The battery life is *long*.")

We implemented the approach using the identical corpora and seed clues. The seed clues were expanded by using the "DING system" to a new set, which is then used for our k-means classifier to measure the classification performance. For a fair comparison, we only measured the performances on positive and negative sentences. The results are shown in Table 2.

**Table 2. Comparison against DING's approach (F-Measure)**

| System | BIZ | INTL | ENV | POL |
|---|---|---|---|---|
| DING | 0.174 | 0.236 | 0.249 | 0.277 |
| OURS | 0.212<br>(+21.8%) | 0.356<br>(+50.8%) | 0.368<br>(+47.8%) | 0.282<br>(+1.8%) |

For the three out of four domains, our approach is significantly superior to the approach in the DING system. The differences in the improvements across the four domains coincide with the results in Fig. 3, in that the performance increases in the BIZ and POL domains are smaller against the DING system than the other domains as is the behavior in the increases over iterations.

**Comparison with general Clues in Supervised Learning.** To examine our hypothesis that contextually-driven features considering topics would enhance the existing keyword based system in SA, we compared the performances of the Support Vector Machine (SVM) sentiment classifiers: one using the context clues (OURS) and the other using SentiWordNet (SentiWN).

**Table 3. Comparison in SVM (F-measure)**

| System | BIZ | INTL | ENV | POL |
|---|---|---|---|---|
| SentiWN | 0.621 | 0.681 | 0.477 | 0.426 |
| OURS | 0.776<br>(+24.9%) | 0.834<br>(+22.4%) | 0.624<br>(+30.8%) | 0.631<br>(+48.3%) |

As Table 3 shows, the performance with the clues in OURS is superior to that with SentiWN clues across all the domains. Particularly, recall performances on all domains are significantly improved. Besides, the results on ENV and POL show that the bootstrapping method is particularly helpful for the case with relatively low performance. At the same time, the performances in the ENV and POL domains are much lower than the others because of the lack of training data.

## 4. Conclusion

In this paper, we proposed a semi-supervised approach to the problem of domain-specific sentiment clue generation in news articles. The evaluation shows that our method is quite successful in extracting contextual clues in news domains and hence in enhancing sentiment classification performance. For further work, we plan to expand the domain corpora by adding more examples into the domains.

## Acknowledgements

This work was financially supported by Microsoft Research Asia, the grant from the strategic technology development program 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-2010-C1090-1011-0008).

## References

Aue, A. and Gamon, M. 2005. Customizing sentiment classifiers to new domains: a case study. In *Proc. of RANLP '05*.

Choi, Y., Kim, Y., and Myaeng, S.-H. 2009. Domain-specific Sentiment Analysis using Contextual Feature Generation. In *Proc. of TSA 2009.*

Ding, X. and Liu, B. 2007. The utility of linguistic rules in opinion mining. In *Proc. of SIGIR '07*.

Esuli, A. and Sebastiani, F. 2006. SentiWordNet: a publicly available lexical resource for opinion mining. In *Proc. of LREC '06*.

Kim, Y. and Myaeng, S.-H. 2007. Opinion analysis based on lexical clues and their expansion. In *Proceedings of NTCIR 6*.

Pang, B. and Lee, L. 2008. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2), Now Publishers, MA.

Seki, Y., Evans, D., Ku, L., Chen, H., Kando, N., and Lin, C. 2007. Overview of opinion analysis pilot task at NTCIR-6. In *Proc. of NTCIR 6*.

Seki, Y., Evans, D., Ku, L., Sun, L., Chen, H., and Kando, N. 2008. Overview of Multilingual Opinion Analysis Task at NTCIR-7. In *Proc. of NTCIR 7*.

Turney, P. 2002. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In *Proc.s of ACL '02*.

Riloff, E., Wiebe, J., and Wilson, T. 2003. Learning subjective nouns using extraction pattern bootstrapping. In *Proc. of CoNLL '03*.