Detecting Nasty Comments from BBS Posts

Tatsuya Ishisaka and Kazuhide Yamamoto

Nagaoka University of Technology
1603-1, Kamitomioka, Nagaoka, Niigata 940-2188 Japan
{ishisaka,yamamoto}@jnlp.org

Abstract. We propose a method to detect Japanese nasty comments from posts on bulletin board systems (BBS). Nasty comments can cause many social problem, because they express potentially harmful words and phrases. There are methods to recognize harmful words, but they are insufficient. Therefore, we present a method for detecting such comments on a BBS with many posts using an n-gram model. In addition, we compared our method with a support vector machine (SVM) that is based on nasty words. As a result, we detected nasty comments that are different to those by the SVM. We also observe higher detection accuracy by combining two methods.

Keywords: sentence detection, nasty comment, n-gram, SVM, BBS

1 Introduction

We focus on Japanese nasty comments that are seen on the Web, particularly, on bulletin board systems (BBS). BBSs are used mainly for information-sharing, consultation, and discussion; however, unfortunately we also see some nasty comments posted on them. Recently, young people, such as primary and secondary students, have been posting such comments. Furthermore, there are many cases where these comments drain the victim emotionally for a long time, and can negatively affect the victim's social life. In a worst-case scenario, the victim commits suicide. These comments are increasing every year all over the world, and have become a social problem, which has been classified as cyber-violence or cyber bullying. This is an effect of the insufficient regulation of the Internet.

The nasty comments must be managed automatically. There are companies in Japan that patrols Web pages manually to find nasty comments. However, manual patrolling is very expensive, so the process to detect nasty comments on BBSs should be automated as much as possible.

Research on detecting nasty comments is similar to research on classifying harmful contents and detecting spam blogs. The POESIA project (Hidalgo *et al.*, 2002)(Hidalgo *et al.*, 2003), which was funded by the European Commission, created a filter for harmful content. This filter classifies whether Web text is pornographic, and is adapted for the English, Italian, and Spanish languages. In other cases, the NET PROTECT project (Grilheres *et al.*, 2004) developed a text classifier for harmful information. Harmful information in Web texts, including pornography, bomb-making, drugs, and violence, is classified based on machine learning using an support vector machine (SVM) (Lee *et al.*, 2007). In addition, Kolari et al. discussed how SVM models based on local and link-based features can be used to detect spam blogs (Kolari *et al.*, 2006).

These studies achieved a high level of accuracy, and several companies are now introducing filtering services against harmful sites; therefore, we can expect to find such harmful sites. However, this filtering is limited only to harmful words that deal with subjects such as pornography and drugs. In contrast, because there can be many patterns for nasty comments, they are difficult to detect. Since nasty comments can express harmful expressions not only in words but also in phrases, we also need to focus on nasty phrases. In addition, we must identify a meaning precisely, because the context and neighboring words can determine whether an expression should be classified as nasty. Following are two sample comments that include a nasty word:

(a) あの政治家は 死ね (aono seijika wa shine.) (That politician must die.)
(b) あの料理は バカ うまい (ano ryouri wa baka umai.) (That dish is very delicious.)

The “死ね (to die)” comment directs harm to another person, as shown in (a). A word such as “die” can be detected easily regardless of its neighboring words, because we need to only judge whether the comment includes “die”. A word such as “バカ (stupid)” make someone who is annoyed or impatient. However “バカ (stupid)” in (b) means “とても (very)” which is used just for emphasis. Therefore, detection is difficult because we need to consider the neighboring words.

Furthermore, a Japanese morpheme analyzer cannot segment the words of BBS posts correctly because they contain several coined words. Therefore, we cannot correctly detect them that are segmented, so nasty words cannot be registered sufficiently. Because of this, we use an n-gram to cope with context and with over-segmented words. The n-gram is used in some natural language processing tasks. Mori and Nagao(Mori and Nagao, 1996) presents a statistical method based on n-gram model for unknown word identification. The method estimates how likely the input string is to be a word. The method cannot cover low frequency unknown words.

In the following sections, we present how we detect nasty comments on a BBS with many posts using an n-gram model.

1.1 Definition of Nasty Expression

Definition of “nasty” can be vague. In this paper, “nasty” is defined as insults and slander words and phrases that are directed toward another person. In other words, slander that forms part of a story or that is used ironically, is not targeted. Therefore, nasty comment is defined as sentence containing these nasty. Concrete examples of nasty comments are shown in Table 1.

Table 1: Concrete Examples of Nasty Comments

Nasty Comments		
Japanese	alphabetized form	English translation
みんなまとめて逝け	minna matomete ike	Fuck off and die
死んでくれって思う	shinde kure te omou	I hope you die
バカな暇人野郎	baka na hima jin yarou	A stupid person of leisure
マジうざい	maji uzai	You are seriously annoying
キモイ！	kimoi!	scumbag!
ヲタは地獄に落ちろ	ota wa jigoku ni ochiro	Go to hell Otaku
死ね	shine	Fuck you

2 Method

Our method consists of the following four steps:

1. Building seeds dictionary of nasty words
2. Collecting nasty comments
3. Making an n-gram model
4. Detecting nasty comments

A schematic view of flow of proposed method is shown in Figure 1.

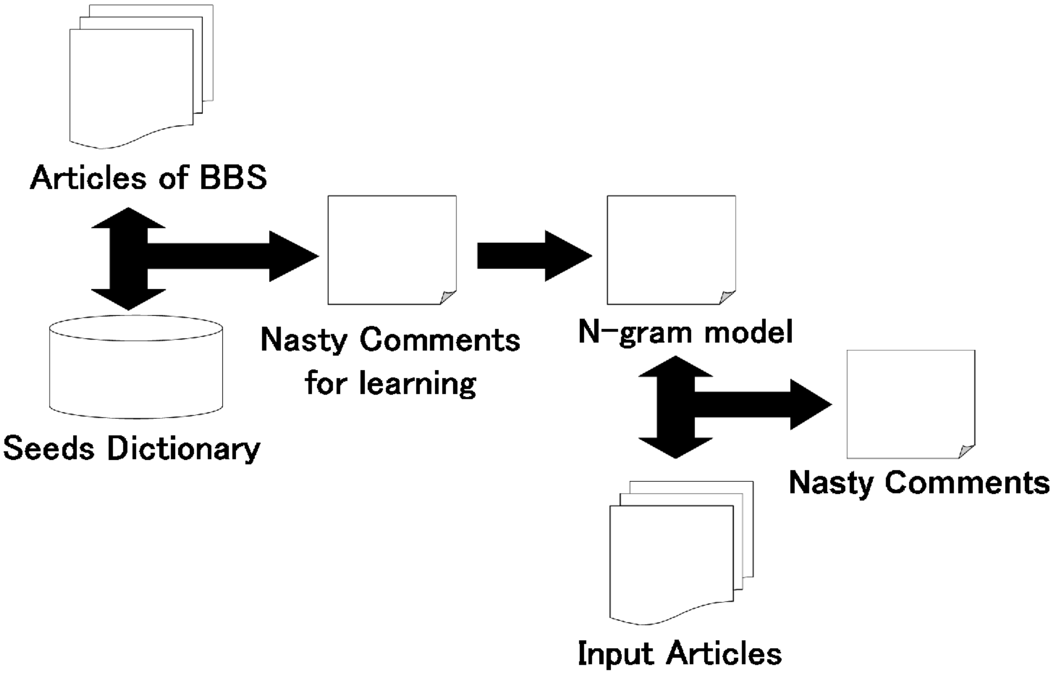

Figure 1: Flow of Proposed Method

2.1 Building seeds dictionary of nasty words

Many nasty comments are necessary to make n-gram model, but it is very expensive to extract them manually from many comments. Therefore, we first extracted nasty keywords manually from the BBS posts and then built a small dictionary. We registered 103 nasty keywords and collected nasty comments automatically using this dictionary.

2.2 Collecting Nasty Comments

We collected articles from a huge BBS called 2-channel[1] (2ch). Many Japanese use or read 2ch, and 2ch is famous in Japan for containing many nasty comments. We defined nasty comments as 2ch including the nasty keyword as a nasty comment. By using our dictionary of registered nasty keywords, we analyzed approximately two thousand articles and collected approximately 6,500 nasty comments every day. Our collected nasty comments are shown in Table 1. We judged that the detection was insufficient because there were only a few variations of nasty comments seen in our comment collection.

As a result of our continued collection of comments from 2ch using the small dictionary, we obtained approximately 200,000 nasty comments. In addition, we collected approximately a half million non-nasty comments from 2ch, because we wanted to analyze strings of words that connect with only nasty comments.

2.3 Making an n-gram Model

We also collected strings of words that connect with the nasty words. In this paper, we focus on word 1-gram to word 5-gram. However, the problem is that a nasty expression may not be a single word. We converted nasty expression which consists of multiple words into a single word (<NASTY>) to obtain more neighboring words.

[1] http://2ch.net/

Following is an example of change:

Before: あの バカ な マス ゴミ の せい で (ano baka na masu gomi no sei de)
After: あの <NASTY> の せい で (ano <NASTY>) no sei de

In this way, we can also obtain the surrounding words of the nasty expression.

We used SRILM (Stolcke, 2002) to create an n-gram model. The SRILM automatically performs back-off smoothing for low-frequency problems. We tokenized the comment into words and tagged the part-of-speech information using the Japanese morphological analyzer ChaSen[2]. All of the morphemes of the comment are modified to their original form to avoid dispersing the probability. We calculated the probability for connecting to nasty words. In addition, we made two-type word n-gram models using the forward n-gram and the backward n-gram for approximately 700,000 comments.

Furthermore, we extracted only word n-grams including the nasty expression from each language models, and merged those models into the nasty words model. Part of the model is shown in Table 2. The scores on the left are the conditional probabilities. <NASTY> is the part which was a nasty expression. <NASTY> is deleted when it is used. The models are regarded as 1- to 4-gram model when we use them. The model has approximately 53,000 patterns as shown in Table 2, and sentences containing these phrases are our extraction targets. We assumed that those with higher probability are nasty.

Table 2: Example of the Nasty Words Model

Nasty word n-gram Model	
0.94	<NASTY> だ な 日本 (<NASTY> da na nihon)
0.67	<NASTY> は さっさと 日本 から (<NASTY> wa sassato nihon kara)
0.62	<NASTY> は 何でも 他人 の (<NASTY> wa nandemo tanin no)
0.94	ない 化学 の 専門 <NASTY> (nai kagaku no senmon <NASTY>)
0.22	顔 見る と 大体 <NASTY> (kao miru to daitai <NASTY>)
0.41	相当 身勝手 だ 単なる <NASTY> (soutou migatte da tannaru <NASTY>)

2.4 Detecting Nasty Comments

We detect nasty comments with the nasty words model. In the following, we show an example of the process flow for detection. If an input sentence includes the phrase of an n-gram model, we judge it to be a nasty comment. We also use simple pattern matching.

step 1. A Input Sentence	マスゴミのクズどもって，何でこうなる事. . . .
	(masugomi no kuzu domo te, nande kou naru koto. . . .)
step 2. Morphological Analysis	マス ゴミ の クズ どもる て , 何で こう なる 事. . . .
	(masugomi no kuzu domoru te, nande kou naru koto. . . .)
step 3. Matching the n-gram	どもる て ， (domoru te ,)
step 4. Judgment	Nasty Comment !

3 Experiment

Experiments were conducted to verify the effectiveness of our proposed method, in which we judged whether input sentences are nasty comments. We made a test set for the evaluation. The test set was extracted from the 2ch, that consists of 378 nasty comments and 380 non-nasty comments. we manually judged whether a sentence is nasty comments or non-nasty comments. Three different raters judged extracted sentences, and we used majority decision as test set.

[2] http://chasen.naist.jp/hiki/ChaSen/ (in Japanese)

We evaluate the result by precision,recall, and F-measure values, defined as follows.

$$Precision = \frac{correctly\ classified\ sentences}{total\ number\ of\ sentences\ classified\ by\ the\ system} \tag{1}$$

$$Recall = \frac{correctly\ classified\ sentences}{total\ number\ of\ sentences} \tag{2}$$

$$F-measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{3}$$

3.1 Results and Discussion of Proposed Method

The result of our proposed method is shown in Figure 2 and Table 3. The threshold is a value of the probability to limit n-gram patterns to use.

If the threshold value is high such as 0.9 or 0.8, the proposed method is surely able to detect nasty comments. In other words, the precision is very high. However, the recall is very low in that case.

We guessed that nasty comment has fixed form, so we focused on neighboring words using n-gram. The result describes that the nasty comment does not have much fixed form. Number of n-gram patterns which were more than probability 0.9 among our n-gram models was approximately 5,600 patterns. We consider that these are few, because these were made from approximately 200,000 nasty comments. However, when the threshold value is low such as 0.1, the F-measure is high. So we understand that the most of the element of the model are effective.

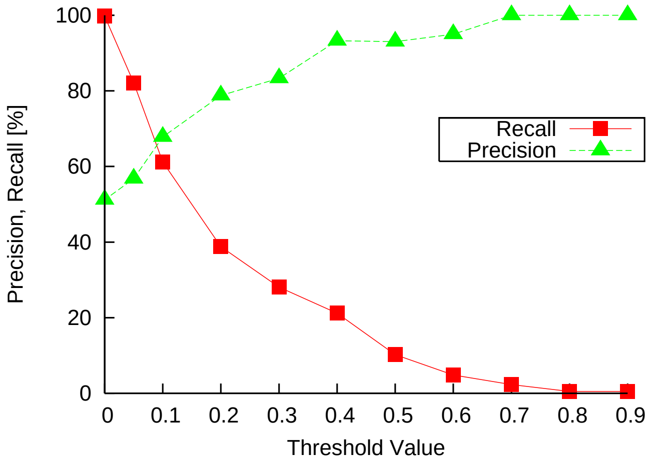

Figure 2: Result of Our Proposed Method

Table 3: Result of F-measure of Our Proposed Method

Threshold Value	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1	0.05
F-measure	1.02	2.52	4.49	9.22	18.39	34.51	41.98	51.97	64.15	**67.65**

3.2 Comparative Method

Lee et al. used an SVM for the filtering of harmful information. We reproduce their method and use it as a baseline for comparison. They experimented on six patterns of feature combination with the SVM. As a result, they described the highest accuracy using a combination of the features of Term Frequency - Inverse Document Frequency (TF-IDF) and Chi-square. We have attempted to apply their method to the classification of nasty comments using the SVM[3]. We replaced IDF with

[3] We used the TinySVM implementation from http://chasen.org/ taku/software/TinySVM/

ISF(Inverse Sentence Frequency) because our target is sentences (comments). The TF-ISF is a statistical measure that is used to evaluate how important a word is to a sentence in our comment set from 2ch. The Chi-square was used to calculate the the dependence relationship between two words. Our Chi-square seeks a quantity of term importance by measuring the dependence relationship between a term and the nasty comment set. We calculate relationship between top 10,000 occurred words in the nasty comment set and the nasty expression. If an input sentence has one of the top 10,000 occurred word, the relationship value is potentiality of nasty comment. When an input sentence has some the top 10,000 appeared word, the average of relationship value is defined as potentiality of nasty comment. We calculate the TF-ISF value and the Chi-square value. The SVM classifies input sentences as nasty comment or non-nasty comments, and evaluates them automatically using the test set which is tagged as the correct answer.

3.3 Results and Discussion of Comparative Method

The classification result of the SVM is shown in Figure 3 and Table 4.

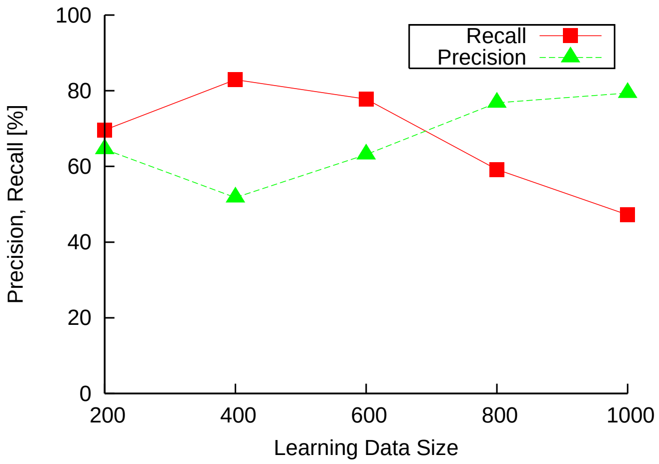

Figure 3: Result of SVM

Table 4: Result of SVM F-measure

Training data size	200	400	600	800	1000
F-measure	66.99	63.73	**69.71**	66.86	59.20

Results show that accuracy does not have the huge difference between the proposed method and comparative method. However, different type of comments were detected. Our proposed method was able to detect a nasty comment based on nasty phrases and over-segmented nasty coined words, but the detection accuracy of the nasty comment based on nasty word was low. On the other hand, the SVM can obtain nasty comments based on nasty words, however the detection accuracy of nasty comments based on nasty phrases was low.

4 Combination Experiment

We guess that the detection accuracy was improved by combining two methods. Combination experiments were conducted to verify the improvement of the accuracy.

4.1 Addition of the feature

We used the TF-ISF and the Chi-square as the features in the Section 3.2. We added the n-gram probability to the feature. The classification result of the SVM adding the feature of n-gram probability is shown in Figure 4 and Table 5.

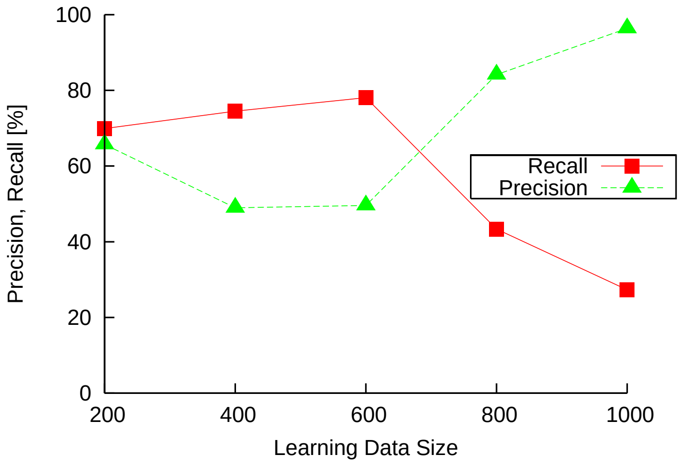

Figure 4: Result of SVM adding n-gram probability

Table 5: Result of F-measure of SVM adding n-gram probability

Training data size	200	400	600	800	1000
F-measure	**67.74**	59.11	60.65	57.24	42.54

Results illustrate that accuracy did not change much. It is still not able to detect nasty comments that each method could not detect, even if a feature of n-gram probability is added to SVM. In addition, there were nasty comments that were not able to detect, although each methods can detect them.

4.2 Sequential Processing

If the threshold value is high such as 0.9 or 0.8, the proposed method is surely able to detect nasty comments. Therefore, we first detect the nasty comments by the proposed method, and the SVM classify the nasty comments which was not detected. It can detect all the nasty comments that two methods can detect. We used only n-gram model in which the threshold value are 0.3 to 0.9, because the precision was more than 80% in section 3.1. We use SVM with 600 sentence training, that performs highest F-measure.

The result of the sequential processing is shown in Figure 5 and Table 6.

Table 6: Result of F-measure of Sequential Processing

Threshold Value	0.3	0.4	0.5	0.6	0.7	0.8	0.9
F-measure	57.40	**72.25**	68.84	70.24	70.75	70.51	70.95

The F-measure was 72.25, which was the highest when the threshold value is 0.4. We can detect the nasty comments including nasty phrases and nasty words. But there was a type of nasty comment that both methods cannot detect. One of them includes nasty comment using a metaphor. We guess that the metaphor does not have fixed neighboring words, therefore both methods cannot detect them. We must think about the nasty comment of the type that we was not able to detect.

5 Conclusion

We have reported a method of detecting nasty comments using an n-gram from the posts on a BBS, because nasty comments cause some social problems. Our proposed method can detect

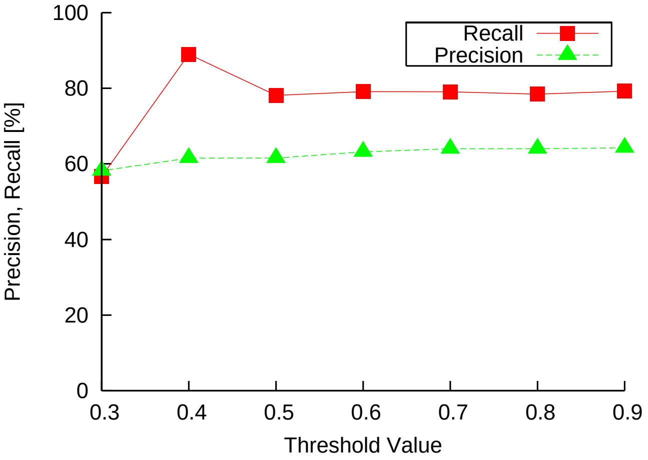

Figure 5: Result of Sequential Processing

nasty comments based on nasty phrases and over-segmented words. We described that a technique of detecting harmful sites is inadequate for detecting nasty comments. In addition, we also described that the detection accuracy is improved by sequential processing using each method. Nasty comments include a type that does not have fixed neighboring words, therefore, we should propose improved method.

There are victims by nasty comments on the Web all over the world. We wish the technical studies that can control nasty comments like our research become more popular.

References

Grilheres, Bruno, Stephan Brunessaux, and Philippe Leray. 2004. Combining Classifiers for harmful document filtering. *In Proceedings of the 7th international conference on Adaptivity*, pp. 173–185.

Hidalgo, José María Gómez, Ignacio Giráldez, and Manuel de Buenaga. 2003. Text Categorization for Internet Content Filtering. *Revista Iberoamericana de Inteligencia Artificial*, pp. 34–52.

Hidalgo, José María Gómez, Enrique Puertas Sanz, Manuel de Buenaga Rodríguez, and Francisco Carrero García. 2002. Text filtering at POESIA: a new Internet content filtering tool for educational environments. *Procesamiento del Lenguaje Natural*, pp. 291–292.

Kolari, Pranam, Akshay Java, Tim Finin, Tim Oates, and Anupam Joshi. 2006. Detecting Spam blogs: A machine learning approach. *In Proceedings of the 21st National Conference on Artificial Intelligence*.

Lee, Wonhee, Samuel Sangkon Lee, Seungjong Chung, and Dongun An. 2007. Harmful Contents Classification Using the Harmful Word Filtering and SVM. *In Proceedings of the 7th International Conference on Computational Science Part III*, pp. 18–25.

Mori, Shinsuke and Makoto Nagao. 1996. Word Extraction from Corpora and Its Part-of-Speech Estimation Using Distributional Analysis. *In Proceedings of the International Conference on Computational Linguistics*, pp. 1119–1122.

Stolcke, Andreas. 2002. SRILM - an extensible language modeling toolkit. *In Proceedings of the International Conference on Spoken Language Processing*, pp. 901–904.