Extracting Patterns of Harmful Expressions for Cyberbullying Detection

Michal Ptaszynski*, Fumito Masui*, Yasutomo Kimura†, Rafal Rzepka‡, Kenji Araki‡

*Department of Computer Science, Kitami Institute of Technology
{ptaszynski,f-masui}@cs.kitami-it.ac.jp

†Department of Information and Management Science, Otaru University of Commerce,
kimura@res.otaru-uc.ac.jp

‡Graduate School of Information Science and Technology, Hokkaido University,
{rzepka,araki}@ist.hokudai.ac.jp

Abstract

Cyberbullying, or humiliating and slandering people through Internet, has been recently noticed as a serious social problem disturbing mental health of Internet users. In Japan, to deal with the problem, voluntary members of Parent-Teacher Association (PTA) manually read through the Web to spot cyberbullying entries. To help PTA members in their uphill task we propose a novel method for automatic detection of malicious contents on the Internet. The method is based on a combinatorial approach resembling brute force search algorithms with application to language classification. The method extracts sophisticated patterns from sentences and uses them in classification. We tested the method on actual data containing cyberbullying provided by Human Rights Center. The results show our method outperformed previous methods. It is also more efficient as it requires minimal human effort.

1. Introduction

Recent years have brought to light a problem greatly impairing public mental health, often in young Internet users. It is the problem of cyberbullying, defined as exploitation of online means of communication, such as Internet forum boards, or social networks to convey harmful and disturbing information about private individuals, often children and students. Messages classifiable as cyberbullying, include ridiculing someone's personality, body type, or appearance, slandering or spreading rumors and insinuations. Some cases of cyberbullying lead the victims to self mutilation, suicides, or attacking their offenders.

In Japan the problem has become serious enough to be noticed by the Ministry of Education (MEXT 2008). In 2007 Japanese school personnel and members of Parent-Teacher Association (PTA) have started monitoring activities under the general name Internet Patrol (later: net-patrol) to spot Web sites containing such inappropriate contents. However, the net-patrol is performed manually as a volunteer work. Countless amounts of data on the Internet make this an uphill task.

This situation motivated us to take up a long term project, in which we aim to contribute to solving the problem of cyberbullying. In the present research we aim at developing a solution which would help and ease the burden of the net-patrol members and create a net-patrol crawler automatically spotting cyberbullying entries on the Web and reporting them to appropriate organs. In this paper we specifically focus on developing a systematic approach to automatically detecting and classifying cyberbullying entries.

The outline of this paper is as follows. Firstly, we define the problem of cyberbullying and present some of the previous research related to ours. Next, we describe our method and the dataset used in this research. Finally, we explain the evaluation settings, thoroughly analyze and discuss the results.

2. Background

2.1. Cyberbullying – A Social Problem

The problem of harmful and offending messages on the Internet has existed for many years. One of the reasons such activities evolved was the anonymity of communication on the Internet, giving users the feeling that anything can go unpunished. Recently the problem has been officially defined and labeled as cyberbullying (CB). The National Crime Prevention Council states that CB happens "when the Internet, cell phones or other devices are used to send or post text or images intended to hurt or embarrass another person."[1].

Some of the first robust research on CB was done by Hinduja and Patchin, who performed numerous surveys about the subject in the USA (Patchin & Hinduja 2006). They found out that the harmful information may include threats, sexual remarks, pejorative labels, or false statements aimed to humiliate others. When posted on a social network, such as Facebook or Twitter, it may disclose humiliating personal data of the victim defaming and ridiculing them personally.

In Japan, after a several cases of suicides of CB victims Japanese Ministry of Education, Culture, Sports, Science and Technology (later: MEXT) considered the problem as serious and began a movement against it. In a manual for handling the CB cases (MEXT 2008), the Ministry puts a great importance on early detection of suspicious entries, especially on Social Networking Services (SNS) and informal school Websites. A movement of Internet Patrol (later: net-patrol) was founded to deal with the problem. Its participants are usually teachers and PTA members. Based on the MEXT definition of CB, they read through all Internet contents, and when they find a harmful entry they send a deletion request to the Web page administrator and report about the event to the Police.

[1] http://www.ncpc.org/cyberbullying

Unfortunately, at present net-patrol is performed manually as a voluntary work. This includes reading the countless entries, deciding about their harmfulness, printing out or taking photos of the pages, sending deletion requests and reports to appropriate organs. The surveillance of the whole Web is an uphill task for the small number of net-patrol members. Moreover, the task comes with great psychological burden on mental health to the net-patrol members. With this research we aim to create a tool allowing automatic detection of CB on the Internet to ease the burden of net-patrol volunteers.

2.2. Previous Research

There has been a small number of research on extracting harmful information from the Internet. For example, (Ishisaka and Yamamoto 2010) developed a dictionary of abusive expressions based on a large Japanese electronic bulletin board (BBS) *2channel*. In their research they labeled words and paragraphs in which the speaker explicitly insults other people with words and phrases like *baka* ("stupid"), or *masugomi no kuzu* ("trash of mass-mudia"). Based on which words appeared most often with abusive vocabulary, they extracted abusive expressions from the surrounding context.

(Ptaszynski et al. 2010) performed affect analysis of small dataset of cyberbullying entries to find out that distinctive features for cyberbullying were vulgar words. They applied a lexicon of such words to train an SVM classifier. With a number of optimizations the system was able to detect cyberbullying with 88.2% of F-score. However, increasing the data caused a decrease in results, which made them conclude SVMs are not ideal in dealing with frequent language ambiguities typical for cyberbullying.

Ikeda and Yanagihara manually collected a set of harmful and non-harmful separate sentences (Ikeda and Yanagihara 2010). Based on word occurrence within the corpus they created a list of keywords for classification of harmful contents. However, they struggled with variations of the same expressions differing with only one or two characters, such as *bakuha* "blow up" and *baku–ha* "blooow up". All variations of the same expression needed to be collected manually, which was a weakness of this method.

Fujii et al. proposed a system for detecting documents containing excessive sexual descriptions using a distance between two words in a sentence (Fujii et al. 2010). They defined as harmful "black words" those in close distance to words appearing only in harmful context, rather than in both harmful and non-harmful context ("grey words").

Next, (Matsuba et al.2011) proposed a method to automatically detect harmful entries, in which they extended the SO-PMI-IR score (Turney 2002) to calculate relevance of a document with harmful contents. With the use of a small number of seed words they were able to detect large numbers of candidates for harmful documents with an accuracy of 83% on test data.

Later, (Nitta et al. 2013) proposed an improvement to Matsuba et al.'s method. They used seed words from three categories (abusive, violent, obscene) to calculate SO-PMI-IR score and maximized the relevance of categories. Their method achieved 90% of Precision for 10% Recall. We used both of the above methods as a baselines for comparison due to similarities in used datasets and experiment settings. Unfortunately, method by (Nitta et al. 2013), based on *Yahoo!* search engine API, faced a problem of a sudden drop in Precision (over 30 percentage-points) across two years, since being originally proposed. This was caused by change in information available on the Internet. In section 4.5. we discuss the possible reasons for this change. Recently (Hatakeyama et al. 2015) tried to improve the method by automatically acquiring and filtering harmful seed words, with a considerable success.

In our research we aimed at minimization of human effort. Most of the previous research assumed that using vulgar words as seeds will help detecting cyberbullying. However, all of them notice that vulgar words are only one kind of distinctive vocabulary and do not cover all cases. We assumed that this kind of vocabulary could be extracted automatically. Moreover, we did not restrict the scope to words, or even phrases (ngrams). We extended the search to sophisticated patterns with disjoint elements. To achieve this we developed a pattern extraction method based on the idea of brute force search algorithm.

3. Method Description

We assumed that applying sophisticated patterns with disjoint elements should provide better results than the usual bag-of-words or n-gram approach. Such patterns can be defined as ordered combinations of sentence elements.

To extract such sophisticated patterns we applied a language modeling method based on the idea of language combinatorics (Ptaszynski et al. 2011). This idea assumes that linguistic entities, such as sentences can be perceived as bundles of ordered non-repeated combinations of elements (words, punctuation marks, etc.). Furthermore, the most frequent combinations appearing in many different sentences can be defined as sentence patterns.

In this method, firstly, ordered non-repeated combinations are generated from all elements of a sentence. In every n-element sentence there is k-number of combination clusters, such as that $1 \leq k \leq n$, where k represents all k-element combinations being a subset of n. The number of combinations generated for one k-element cluster of combinations is equal to binomial coefficient. In this procedure the system creates all combinations for all values of k from the range of $\{1, ..., n\}$. Therefore the number of all combinations is equal to the sum of all combinations from all k-element clusters of combinations, like in equation 1.

$$\sum_{k=1}^{n} \binom{n}{k} = \frac{n!}{1!(n-1)!} + \frac{n!}{2!(n-2)!} + ... + \frac{n!}{n!(n-n)!} = 2^n - 1 \quad (1)$$

Next, all non-subsequent elements are separated with an asterisk ("*"). All patterns generated this way are used to extract frequent patterns appearing in a given corpus. Their occurrences O is used to calculate their normalized weight w_j according to equation 2. The score of a sentence is calculated as a sum of weights of patterns found in the sentence, like in equation 3.

$$w_j = \left(\frac{O_{pos}}{O_{pos} + O_{neg}} - 0.5 \right) * 2 \quad (2)$$

$$score = \sum w_j, (1 \geq w_j \geq -1) \quad (3)$$

The weight can be later calculated in several ways. Two features are important in weight calculation. A pattern is the more representative for a corpus when, firstly, the longer the pattern is (length k), and the more often it appears in the corpus (occurrence O). Thus the weight can be modified by

- awarding length (later: `LA`),
- awarding length and occurrence (later: `LOA`).

The list of generated frequent patterns can be also further modified. When two collections of sentences of opposite features (such as "positive" vs. "negative") are compared, a generated list of patterns will contain patterns that appear uniquely in only one of the sides (e.g. uniquely positive or negative patterns) or in both (ambiguous patterns). Therefore the pattern list can be further modified by

- erasing all ambiguous patterns (later: `AMB`),
- erasing only ambiguous patterns which appear in the same number in both sides (later zero patterns, or `0P`).

Moreover, a list of patterns will contain both the sophisticated patterns (with disjoint elements) as well as more common n-grams. Therefore the experiments were performed either with patterns (`PAT`), or n-grams (`NGR`) only. If the initial collection of sentences was biased toward one of the sides (e.g., more sentences of one kind, or the sentences were longer, etc.), there will be more patterns of a certain sort. Thus to avoid bias in the results, instead of applying a rule of thumb, threshold is automatically optimized. The above settings are automatically verified in the process of evaluation (10-fold cross validation) to choose the best model. The metrics used in evaluation are standard Precision (P), Recall (R) and balanced F-score (F). Finally, to deal with the combinatorial explosion mentioned on the beginning of this section we applied two heuristic rules. In the preliminary experiments we found out that the most valuable patterns in language are up to six element long, therefore we limited the scope to $k \leq 6$. Thus the procedure of pattern generation will (1) generate up to six elements patterns, or (2) terminate at the point where no frequent patterns were found.

4. Evaluation Experiment

4.1. Dataset

At first we needed to prepare a dataset. We used the dataset created originally by (Matsuba et al. 2010) and developed further by (Matsuba et al.2011). The dataset was also used by (Ptaszynski et al. 2010) and recently by (Nitta et al. 2013). It contains 1,490 harmful and 1,508 non-harmful entries. The original data was provided by the Human Rights Research Institute Against All Forms for Discrimination and Racism in Mie Prefecture, Japan[2] and contains data from unofficial school Web sites and fora. The harmful and non-harmful sentences were manually labeled by Internet Patrol members according to instructions included in the MEXT manual for dealing with cyberbullying (MEXT 2008). Some of those instructions are explained shortly below.

The MEXT definition assumes that cyberbullying happens when a person is personally offended on the Web. This includes disclosing the person's name, personal information and other areas of privacy. Therefore, as the first feature distinguishable for cyberbullying MEXT defines private names. This includes such information as:

- Private names and surnames,
- Initials and nicknames,
- Names of institutions and affiliations,

As the second feature distinguishable for cyberbullying MEXT defines any other type of personal information. This includes:

- Address, phone numbers,
- Questions about private persons (e.g. "Who is that tall guy straying on Computer Science Dept. corridors?"),
- Entries revealing other personal information (e.g. "I hate that guy responsible for the new project against cyberbullying.").

Also, according to MEXT, vulgar language is distinguishable for cyberbullying, due to its ability to convey offenses against particular persons. This is also confirmed in other literature (Patchin & Hinduja 2006; Ptaszynski et al. 2010). Examples of such words are, in English: *sh*t*, *f*ck*, or *b*tch*, in Japanese: *uzai* (freaking annoying), or *kimoi* (freaking ugly).

In the prepared dataset all entries containing any of the above information was classified as harmful. Some examples from the dataset are represented in Table 1.

4.2. Dataset Preprocessing

The language combinatorics method takes as an input sentences separated into elements (words, tokens, etc.). Therefore we needed to preprocess the dataset and make the sentences separable into elements. We did this in several ways to check how the preprocessing would influence the results. We used MeCab[3], a standard morphological analyzer for Japanese to preprocess the sentences from the dataset in the following ways:

- **Tokenization:** All words, punctuation marks, etc. are separated by spaces (later: `TOK`).
- **Parts of speech (POS):** Words are replaced with their representative parts of speech (later: `POS`).
- **Tokens with POS:** Both words and POS information is included in one element (later: `TOK+POS`).

The examples of preprocessing are represented in Table 2. Theoretically, the more generalized a sentence is, the less unique patterns it will produce, but the produced patterns will be more frequent. This can be explained by comparing tokenized sentence with its POS representation. In the sentence from Table 2, the phrase *kimochi ii* ("pleasant") can be represented by a POS pattern `N ADJ`. We can easily assume that there will be more `N ADJ` patterns than *kimochi ii*, because many word combinations can be represented by this pattern. Therefore POS patterns will come in less variety but with higher occurrence frequency. By comparing the result of the classification using different preprocessing methods we can find out whether it is better to represent sentences as more generalized or as more specific.

[2] http://www.pref.mie.lg.jp/jinkenc/hp/

[3] https://code.google.com/p/mecab/

Table 1: Four examples of cyberbullying entries gathered during Internet Patrol. The upper three represent strong sarcasm despite of the use of positive expressions in the sentence. English translation below Japanese content. Harmful patterns recognized automatically – underlined (underlining in English was made to correspond as closely to Japanese as possible).

>>104 Senzuri koi te shinu nante? sonna hageshii senzuri sugee naa. "Senzuri masutaa" toshite isshou agamete yaru yo. >>104 Dying by 'flicking the bean'? Can't imagine how one could do it so fiercely. I'm gonna worship her as a 'master-bator', that's for sure.
2-nen no tsutsuji no onna meccha busu suki na hito barashimashoka? 1-nen no anoko desuyo ne? kimogatterunde yamete agete kudasai Wanna know who likes that awfuly ugly 2nd-grade Azalea girl? Its that 1st-grader isn't it? He's disgusting, so let's leave him mercifully in peace.
Aitsu wa busakute sega takai dake no onna, busakute se takai dake ya noni yatara otoko-zuki meccha tarashide panko anna onna owatteru She's just tall and apart of that she's so freakin' ugly, and despite of that she's such a cock-loving slut, she's finished already.
Shinde kureeee, daibu kiraware-mono de yuumei, subete ga itaitashii... Please, dieeee, you're so famous for being disliked by everyone, everything in you is so pathetic

Table 2: Three examples of preprocessing of a sentence in Japanese; N = noun, TOP = topic marker, ADV = adverbial particle, ADJ = adjective, COP = copula, INT = interjection, EXCL = exclamative mark.

Sentence: 今日はなんて気持ちいい日なんだ！ **Transliteration:** *Kyōwanantekimochiiihinanda!* **Meaning:** Today TOP what pleasant day COP EXCL **Translation:** What a pleasant day it is today!
Preprocessing examples
1. Tokenization: *Kyō wa nante kimochi ii hi nanda !* **2. POS:** `N TOP ADV N ADJ N COP EXCL` **3.Tokens+POS:** *Kyō*`[N]` *wa*`[TOP]` *nante*`[ADV]` *kimochi*`[N]` *ii*`[ADJ]` *hi*`[N]` *nanda*`[COP]` *!*`[EXCL]`

4.3. Experiment Setup

The preprocessed original dataset provides three separate training and test sets for the experiment (tokenized, POS-tagged and tokens with POS together). The experiment was performed three times, one time for each kind of preprocessing to choose the best option. For each version of the dataset a 10-fold cross validation was performed and the results were calculated using standard Precision, Recall and balanced F-score for the whole threshold span. In one experiment 14 different versions of the classifier are compared with 10-fold cross validation condition. Since the experiment is performed for three different versions of preprocessing, we obtained overall number of 420 experiment runs. There were several evaluation criteria. Firstly, we looked at which version of the algorithm achieved the top score within the threshold span. We also looked at break-even points (BEP) of Precision and Recall. Finally, we checked the statistical significance of the results. We used paired t-test because the classification results could represent only one of two classes (harmful or non-harmful). To chose the best version of the algorithm we compared separately the results achieved by each group of modifications, eg., "different pattern weight calculations", "pattern list modifications" and "patterns vs n-grams". We also compared the performance to the baseline (Nitta et al. 2013).

4.4. Results and Discussion

When it comes to Precision, the highest score of all was achieved by the feature sets: `POS/NGR/LA` (P=.93), `POS/NGR`, `POS/NGR/0P` (P=.92) and `POS/NGR/LA/0P` (P=.91). Unfortunately, all with low Recall (R=.02–.03). Despite these occasional top scores for Precision, the POS-tagged dataset achieved in general the lowest balanced F-score (up to F=.78).

Also high Precision with much higher Recall was achieved by feature sets: `TOK+POS/PAT|NGR` and `TOK+POS/PAT|NGR/OP` (P=.89, R=.34). The dataset preprocessing containing both tokens and POS tags also achieved the highest general results in balanced F-score (F=.8 for `TOK+POS/PAT|NGR/0P` and F=.79 for `TOK+POS/PAT|NGR`). Dataset which was only tokenized achieved moderate scores in general. From the fact that the general results ideally corresponded with the sophistication of preprocessing, we infer that the method could be further improved by more sophisticated preprocessing.

Tokenization with POS tagging also provided the highest scores when it comes to break-even point (BEP) of Precision and Recall. The highest scores were achieved by `TOK+POS/PAT|NGR` and `TOK+POS/PAT|NGR/0P` (P=.79, R=.79, F=.79). Since this corresponds to the best results in F-score, we consider the two feature sets as optimal. There were small differences in detailed results between these datasets, however, as they occurred statistically insignificant, we consider both of them as optimal. It could be further noticed that, since `TOK+POS/PAT|NGR/0P` uses less patterns (no zero-patterns), this feature set could be more time-efficient.

When it comes to other modifications, in most cases deleting ambiguous patterns yielded worse results, which suggests that such patterns, despite being ambiguous to some extend (appearing in both cyberbullying and non-cyberbullying entries), are in fact useful in practice. Also, awarding pattern length or occurrence in weight calculation, although causing statistically significant differences in results, did not come with performance improvement.

4.5. Comparison with Previous Methods

After specifying optimal settings for the proposed method, we compared it to previous methods. In the comparison we used the method by (Matsuba et al.2011), (Nitta et al. 2013), and its most recent improvement by (Hatakeyama et al. 2015). Moreover, since the latter extracts cyberbullying relevance values from the Web (in particular *Yahoo! API*), apart from comparison to the reported results we also repeated their experiment to find out how the performance of the Web-based method changed during the three years. Finally, to make the comparison more fair, we compared both our best and worst results. As the

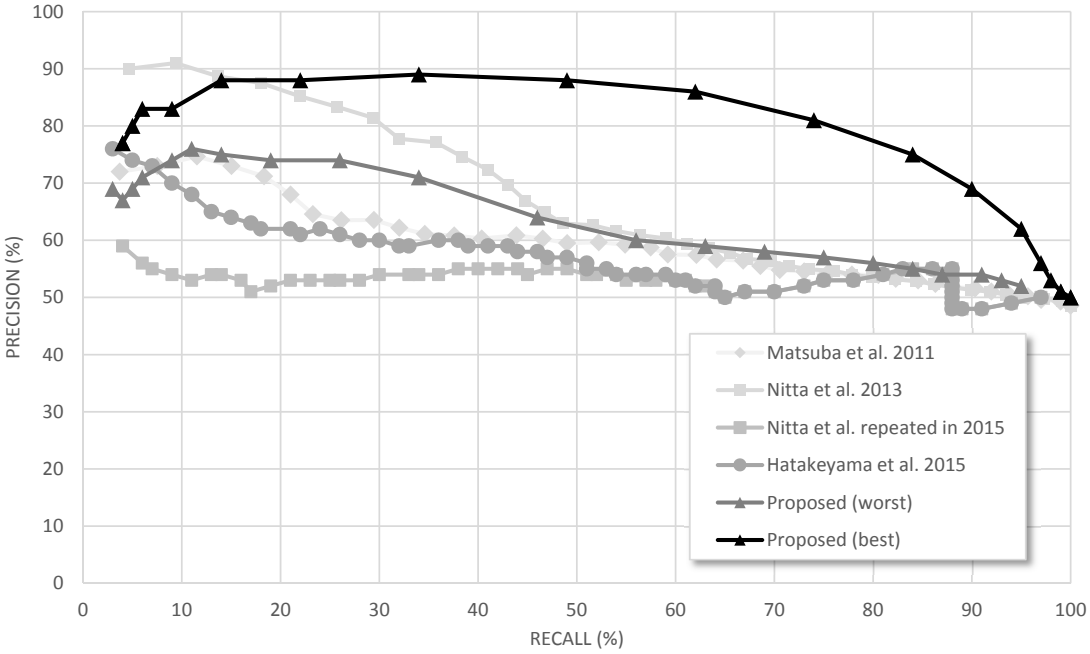

Figure 1: Comparison between the proposed method (best and worst performance) and previous methods.

evaluation metrics we used area under the curve (AUC) on the graph showing Precision and Recall. The results are represented in Figure 1.

The highest overall results when it comes to AUC were obtained by the best settings of the proposed method (tokens with POS, all patterns, no weight modification), which starts from a high 77% and retains the Precision between 80% and 90% for most of the threshold. Although the highest score was still by (Nitta et al. 2013), performance of their method quickly decreases due to quick drop in Precision for higher thresholds. Moreover when we repeated their experiment recently in January 2015, the results greatly dropped. After thorough analysis of the experiment data we noticed that most of the information extracted in 2013 was not available in 2015. This could be due to the following reasons. Firstly, fluctuation in page rankings could push the information lower making it not extractable by Nitta et al.'s method. Secondly, frequent deletion requests of harmful contents by net-patrol members could make their efforts pay off. However, the most probable is the third cause, which is the recent tightening of usage policies by most Web service providers, such as Google[4], Twitter[5] and *Yahoo!* used by (Nitta et al. 2013). This includes recently introduced DMARC[6] policies related to e-mail spoofing and general improvements in policies aimed at decreasing Internet harassment. Such changes aimed at limiting the growing problem of Internet harassment, implemented on a corporate level, are in general a positive phenomenon, despite reducing the performance of cyberbullying detection software. Moreover, as was recently shown by (Hatakeyama et al. 2015), the performance can be to some extent improved by automatically optimizing the list of seed words applied in such methods.

However, The fact that the performance of Nitta et al.'s method decreased from over 90% to less than 60% during 3 years is an important warning for other research based on Web search engines. Probability of such problems have been indicated some time ago (Kilgarriff 2007), and could become a major problem in the future. This also advocates more focus on corpus-based methods such as the one proposed in this paper.

Finally, while the numerical results were in favor of the proposed approach, we also wanted to know to what extent the patterns automatically recognized by the proposed method cover the manually selected seed words in the previous research (Matsuba et al. 2010; Matsuba et al.2011; Nitta et al. 2013). In the result, all seed words appeared in the list of automatically extracted patterns. This can be interpreted as follows. Firstly, CB definition by (MEXT 2008) and hunch of the researchers, on which previous approaches were mostly based, were generally correct. Secondly, using our automatically extracted patterns it could be possible to improve previous approaches in the future.

5. Conclusions and Future Work

In this paper we proposed a method for automatic detection of Internet forum entries that contain cyberbullying (CB) – contents humiliating and slandering other people. CB is a recently noticed social problem which influences mental health of Internet users, and might lead to self-mutilation and even suicide of CB victims.

In the proposed method we applied a combinatorial algorithm, resembling brute force search algorithms, in auto-

[4] https://www.google.com/events/policy/anti-harassmentpolicy.html
[5] https://blog.twitter.com/2014/building-a-safer-twitter
[6] http://www.dmarc.org/

matic extraction of sentence patterns, and used those patterns in text classification of CB entries. We tested the method on actual CB data obtained from Human Rights Center. The results show our method outperformed previous methods. It is also more efficient as it requires minimal human effort.

In the near future we plan to apply different methods of dataset preprocessing to find out whether the performance can be further improved and to what extent. We also plan to obtain new data to evaluate the method more thoroughly, and apply different classifiers. Finally, we plan to verify the actual amount of CB information on the Internet and reevaluate the method in more realistic conditions.

6. References

Bill Belsey. 2007. Cyberbullying: An Emerging Threat for the "Always On" Generation, http://www.cyberbullying.ca/pdf/Cyberbullying_Presentation_Description.pdf

Yutaro Fujii, Satoshi Ando, Takayuki Ito. 2010. *Yūgai jōhō firutaringu no tame no 2-tango-kan no kyori oyobi kyōki jōhō ni yoru bunshō bunrui shuhō no teian* [Developing a method based on 2-word co-occurence information for filtering harmful information] (in Japanese), In *Proceedings of The 24th Annual Conference of The Japanese Society for Artificial Intelligence (JSAI2010)*, paper ID: 3D2-4, pp. 1-4.

Suzuha Hatakeyama, Fumito Masui, Michal Ptaszynski, Kazuhide Yamamoto. 2015. Improving Performance of Cyberbullying Detection Method with Double Filtered Point-wise Mutual Information, In *Demo Session of The 2015 ACM Symposium on Cloud Computing 2015 (ACM-SoCC 2015)*, Kohala Coast, Hawaii, August 27 - 29, 2015.

Hiromi Hashimoto, Takanori Kinoshita, Minoru Harada. 2010. *Firutaringu no tame no ingo no yūgai goi kenshutsu kinō no imi kaiseki shisutemu SAGE e no kumikomi* [Implementing a function for filtering harmful slang words into the semantic analysis system SAGE] (in Japanese), *IPSJ SIG Notes* 2010-SLP-81(14),pp.1-6.

Sameer Hinduja, J. W. Patchin. 2009. *Bullying beyond the schoolyard: Preventing and responding to cyberbullying*. Corwin Press.

Kazushi Ikeda, Tadashi Yanagihara. 2010. *Kakuyōso no chūshōka ni motozuku ihō-, yūgai-bunsho kenshutsu shuhō no teian to hyōka* [Proposal and evaluation of a method for illegal and harmful document detection based on the abstraction of case elements] (in Japanese), In *Proc. of 72nd National Convention of Information Processing Society of Japan (IPSJ72)*, pp. 71-72.

Tatsuya Ishisaka, Kazuhide Yamamoto. 2010. *2chaeru wo taishō to shita waruguchi hyōgen no chūshutsu* [Extraction of abusive expressions from 2channel] (in Japanese), In *Proceedings of The Sixteenth Annual Meeting of The Association for Natural Language Processing (NLP2010)*, pp.178-181.

Adam Kilgarriff. 2007. Googleology is bad science. *Computational linguistics*, Vol. 33, No. 1, pp. 147-151.

Klaus Krippendorff. 1986. Combinatorial Explosion, In: Web Dictionary of Cybernetics and Systems. Principia Cybernetica Web.

Tatsuaki Matsuba, Fumito Masui, Atsuo Kawai, Naoki Isu. 2010. *Gakkou hikoushiki saito ni okeru yuugai jouhou kenshutsu* [Detection of harmful information on informal school websites] (In Japanese). In *Proc. of The 16th Annual Meeting of The Association for Natural Language Processing (NLP2010)*.

Tatsuaki Matsuba, Fumito Masui, Atsuo Kawai, Naoki Isu. 2001. *Gakkō hi-kōshiki saito ni okeru yūgai jōhō kenshutsu wo mokuteki to shita kyokusei hantei moderu ni kansuru kenkyū* [A study on the polarity classification model for the purpose of detecting harmful information on informal school sites] (in Japanese), In *Proceedings of The Seventeenth Annual Meeting of The Association for Natural Language Processing (NLP2011)*, pp. 388-391.

Ministry of Education, Culture, Sports, Science and Technology (MEXT). 2008. *'Netto-jō no ijime' ni kansuru taiō manyuaru jirei shū (gakkō, kyōin muke)* ["Bullying on the Net" Manual for handling and collection of cases (for schools and teachers)] (in Japanese). Published by MEXT.

Taisei Nitta, Fumito Masui, Michal Ptaszynski, Yasutomo Kimura, Rafal Rzepka, Kenji Araki. 2013. Detecting Cyberbullying Entries on Informal School Websites Based on Category Relevance Maximization. In *Proceedings of the 6th International Joint Conference on Natural Language Processing (IJCNLP 2013)*, pp. 579-586.

Justin W. Patchin, Sameer Hinduja. Bullies move beyond the schoolyard: A preliminary look at cyberbullying. *Youth Violence and Juvenile Justice*, 4(2), 148-169 (2006).

Michal Ptaszynski, Pawel Dybala, Rafal Rzepka and Kenji Araki. 2009. Affecting Corpora: Experiments with Automatic Affect Annotation System - A Case Study of the *2channel* Forum -, In *Proceedings of The Conference of the Pacific Association for Computational Linguistics (PACLING-09)*, pp. 223-228.

Michal Ptaszynski, Pawel Dybala, Tatsuaki Matsuba, Fumito Masui, Rafal Rzepka, Kenji Araki, and Yoshio Momouchi. 2010. In the Service of Online Order: Tackling Cyber-Bullying with Machine Learning and Affect Analysis. *International Journal of Computational Linguistics Research*, Vol. 1, Issue 3, pp. 135-154.

Michal Ptaszynski, Rafal Rzepka, Kenji Araki, Yoshio Momouchi. 2011. Language combinatorics: A sentence pattern extraction architecture based on combinatorial explosion. *International Journal of Computational Linguistics (IJCL)*, Vol. 2, No. 1, pp. 24-36.

Peter D. Turney. 2002. Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews, In *Proc. of the 40th Annual Meeting of the Association for Computational Linguistics*, Philadelphia, pp. 417-424.