Learning Deep on Cyberbullying is Always Better Than Brute Force

Michal Ptaszynski†, **Juuso Kalevi Kristian Eronen**‡† and **Fumito Masui**†
†Kitami Institute of Technology, Kitami, Japan
‡Tampere University of Technology, Tampere, Finland
{ptaszynski,f-masui}@cs.kitami-it.ac.jp juuso.eronen@student.tut.fi

Abstract

In this paper we present our research on detection of cyberbullying (CB), which stands for humiliating other people through the Internet. CB has become recognized as a social problem, and its mostly juvenile victims usually fall into depression, self-mutilate, or even commit suicide. To deal with the problem, school personnel performs Internet Patrol (IP) by reading through the available Web contents to spot harmful entries. It is crucial to help IP members detect malicious contents more efficiently. A number of research has tackled the problem during recent years. However, due to complexity of language used in cyberbullying, the results has remained only mildly satisfying. We propose a novel method to automatic cyberbullying detection based on Convolutional Neural Networks and increased Feature Density. The experiments performed on actual cyberbullying data showed a major advantage of our approach to all previous methods, including the best performing method so far based on Brute-Force Search algorithm.

1 Introduction

Recent years brought to light the problem of **cyberbullying** (CB), defined as exploitation of open online means of communication, such as Internet forum boards, or social network services (SNS) to convey harmful and disturbing information about private individuals, often children and students [Patchin and Hinduja, 2006]. The problem was further exacerbated by the popularization of smartphones and tablet computers, which allow nearly constant use of SNS at home, work/school or in motion [Bull, 2010].

Cyberbullying messages commonly ridicule someone's personality, appearance or spread rumors. It can lead its victims to self mutilation, even suicides, or on the opposite, to attacking their offenders in revenge [Hinduja and Patchin, 2010]. Global increase of cyberbullying cases opened a public debate on whether such messages could be spotted earlier to prevent the tragedies, and on the freedom of speech on the Internet in general.

In some countries, such as Japan, the problem has become serious enough to be noticed on a ministerial level [MEXT, 2008]. As one of the ways to deal with the problem school personnel have started Internet Patrol (IP) to detect Web forum sites and SNS containing cyberbullying contents. Unfortunately, as IP is performed manually, reading through countless amounts of Websites makes it an uphill struggle.

Some research have started developing methods for automatic detection of CB to help in this struggle [Ptaszynski *et al.*, 2010; Dinakar *et al.*, 2012; Ptaszynski *et al.*, 2016]. Unfortunately, even with multiple improvements, the results have remained merely partially satisfying. This is caused by a multitude of language ambiguities and styles used in CB.

In this paper we propose a novel, Convolutional Neural Networks (CNN) approach to automatic cyberbullying detection. Moreover, based on the analysis of the characteristics of CNN and the initial results, we propose an optimization of CNN by increasing Feature Density of training data.

The rest of the paper is organized int he following way. Firstly, we describe the problem of cyberbullying and present some of the previous research related to ours. Next, we describe the proposed method and other methods used for comparison. Further, we present the dataset used in this research, and explain the evaluation settings, followed by analysis of experiment results and discussion.

2 Research Background

2.1 Cyberbullying: Description of a Problem

The choice of media used in communication can cause increased psychological distance between interlocutors [Rutter, 1987], which can lead to empathy deficit, especially in Internet behavior [Zheng, 2012]. This is one of the reasons offensive messages have existed for many years on the Internet. With the increase of our dependence on technology in everyday lives, the problem gained on seriousness, and conceptualized itself in the form of online harassment, or cyberbullying (CB) [Patchin and Hinduja, 2006; Hinduja and Patchin, 2010; Pyżalski, 2012; Lazuras *et al.*, 2012].

Some of the first research on CB, based on numerous surveys [Patchin and Hinduja, 2006] revealed that such 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 could disclose humiliating private information of the victim defaming and ridiculing them publicly. Some re-

Table 1: Summary of previous research in CB detection.

Authors, Year	**Language of processing**	**Feature Extraction Method**	**Classification Method**
[Ptaszynski *et al.*, 2010]	Japanese	unigrams (BoW), harmful word lexicon	SVM
[Sood *et al.*, 2012]	English	unigrams, bigrams, stems	SVM, various weighting (presence, freq, tf-idf)
[Dinakar *et al.*, 2012]	English	unigrams, hand-crafted word-lists (Ortony lexicon of negative words, profane words, etc.), POS	SVM, JRip, Naïve Bayes, J48
[Nitta *et al.*, 2013]	Japanese	seed words in 3 categories	SO-PMI-IR maximized for category
[Cano Basave *et al.*, 2013]	English	violence-related words (derived from violence-related topics from Twitter and DBPedia)	VDM (weakly supervised Bayesian model)
[Sarna and Bhatia, 2015]	English (?)	"bad words", positive and negative sentiment words, pronouns, proper nouns, links	Naïve Bayes, kNN, Decision Trees, SVM
[Ptaszynski *et al.*, 2015a]	Japanese	Brute-Force search algorithm	custom pattern matching classifier
[Ptaszynski *et al.*, 2015b]	Japanese	Brute-Force search algorithm with various features (tokens, POS, lemmas, etc.)	custom pattern matching classifier
[Ptaszynski *et al.*, 2016]	Japanese	seed words grouped in 3 categories	SO-PMI-IR maximized for category with seed word optimization

ported that CB happens for up to eight percent of children in schools in: Australia [Cross *et al.*, 2009], United States [Kowalski and Limber, 2007], or Finland [Sourander *et al.*, 2010]. Studies on CB across Europe indicate that even one in five young people (not limited to school environment) could be exposed to cyberbullying [Hasebrink *et al.*, 2008; Pyżalski, 2012]. As of 2015 the urgent need to deal with CB has even made insurance companies offer policies from costs that could occur as a result of cyberbullying[1].

In Japan, after a several suicide cases of CB victims, Ministry of Education, Culture, Sports, Science and Technology (MEXT) increased the priority of the problem, provided a yearly updated manual for handling CB cases and incorporated it in school staff education program [MEXT, 2008].

To actively deal with the problem, school staff are engaged in Internet Patrol (IP). 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 the event to the police. Unfortunately, since IP has been performed manually as a voluntary work, and the amounts of Internet fora and SNS to read through grows exponentially, manual Web surveillance has been an uphill task, and a psychological burden for the IP members.

2.2 Previous Research on Cyberbullying Detection

Although the problem of CB has been studied in social sciences and child psychology for over ten years[Patchin and Hinduja, 2006; Pyżalski, 2012], only few attempts were made so far to detect and study the problem with the help of information technology. Below we present the most relevant research to this day (also summarized in Table 1).

As the first recorded study, [Ptaszynski *et al.*, 2010] performed affect analysis on a small dataset of CB entries to find out that distinctive features for CB were vulgar words. They applied a lexicon of such words to train an SVM classifier. With a number of optimizations they were able to detect cyberbullying with 88.2% of F-score. However, increasing the data caused a decrease in results, which made them abandon SVM as not ideal for language ambiguities typical for CB.

[Sood *et al.*, 2012] focused on detection of personal insults, negative influence of which could at most cause the Internet community to fall into recession. In their research they used as features single words and bigrams, weighted them using either presence (1/0), term frequency or tf-idf, and used them to train an SVM classifier. As a dataset they used a corpus of six thousand entries they collected from various online fora. To prepare gold standard for their experiments they used a crowd-sourcing approach with untrained layperson annotators hired for a classification task through Mechanical Turk.

Later, [Dinakar *et al.*, 2012] proposed their approach to detection and mitigation of cyberbullying. An improvement of this paper in comparison to previous research was its wider perspective, in which they did not only focus on the detection, but also proposed some ways for mitigation. The classifiers they used scored up to 58-77% of F-score depending on the kind of detected harassment. Their best proposed classifier was SVM, which confirmed considerably high effectiveness of SVM for cyberbullying in English, similarly to the research done by Ptaszynski *et al.,* for Japanese in 2010.

An interesting work was done by [Kontostathis *et al.*, 2013], who performed a thorough analysis of cyberbullying entries on Formspring.me. They were able to identify common cyberbullying terms, and applied them in classification with the use of a machine learning method based on Essential Dimensions of atent Semantic Indexing (EDLSI).

[Cano Basave *et al.*, 2013] proposed Violence Detection Model (VDM), a weekly supervised Bayesian model. They did not however focused strictly on cyberbullying, but widened their scope to more generally understood "violence," which made the problem more understandable, and thus feasible for untrained annotators. The datasets were extracted from violence-related topics on Twiter and DBPedia.

[Nitta *et al.*, 2013] proposed a method to automatically detect harmful entries with an extended SO-PMI-IR score [Turney, 2002] to calculate the relevance of a document with harmful contents. They also grouped the seed words into three categories (abusive, violent, obscene) and maximized the relevance of categories. Their method was evaluated comparatively high with the best achieved Precision around 91% (although with Recall less then 10%).

Unfortunately, a re-evaluation of their method done by [Ptaszynski *et al.*, 2016] two years later, indicated that the method lost most of its Precision (over 30 percentage-point drop) in that time. They hypothesized that this was caused by external factors such as Web page re-ranking, or changes in SNS user policies, etc. They improved the method by automatically acquiring and filtering new harmful seed words with some success (P=76%). Unfortunately, they were unable to revive the method to their original performance.

[Sarna and Bhatia, 2015] based their method on a set of features like "bad words", positive/negative sentiment words, and other common features like pronouns, etc., to estimate

[1] http://news.na.chubb.com/2016-03-30-Cyber-Bullying-Insurance-Now-Available-to-Chubbs-U-S-Homeowners-Customers-2 [accessed on 2017/02/19]

user credibility. They applied those features to four standard classifiers (Naïve Bayes, kNN, Decision Trees, SVM). The results of the classification were further used in User Behavior Analysis model (BAU), and User Credibility Analysis (CAU) model. Unfortunately, although their approach suggested inclusion of phenomena such as irony, or rumors, in practice they only focused on messages containing "bad words." Moreover, neither these words, the dataset, nor its annotation schema were sufficiently described in the paper.

Finally, [Ptaszynski *et al.*, 2015a] proposed a method of pattern-based language modeling. The patterns, defined as ordered combinations of sentence elements, were extracted with a Brute-Force search algorithm and used in classification. They reported encouraging initial results, and further improved the method by applying multiple data preprocessing techniques [Ptaszynski *et al.*, 2015b]. At present their method is considered as the best performing method, thus we will use it in comparison with the method proposed herein.

2.3 Research Gaps

Dataset preparation Some of the above-mentioned methods suffer from subjective data preparation. In [Cano Basave *et al.*, 2013] or [Dinakar *et al.*, 2012], the problem was not defined strictly enough and annotated by laypeople, while CB is a complex social phenomenon and needs to be handled by experts. [Sood *et al.*, 2012; Cano Basave *et al.*, 2013] reformulated the problem to be feasible by laypeople. [Dinakar *et al.*, 2012] focused on overlapping concepts like sexual or racial harassment. Finally, [Sarna and Bhatia, 2015] collected the datasets with no specific standard.

Feature selection Previous research included as features mostly words, or simple n-grams (bigrams). Some [Nitta *et al.*, 2013] applied only a small number of features, while others [Dinakar *et al.*, 2012] build up more complex models, however still based mostly on words. Moreover, using only top-down selected features [Nitta *et al.*, 2013; Sarna and Bhatia, 2015], while somewhat reasonable (e.g., violent or obscene words) requires human workload and background knowledge on the dataset, thus being inefficient.

Classification methods Although various classifiers have been tested (SVM, Naive Bayes, or Decision Trees), usually SVM reached highest, though mildly satisfying scores. To overcome the performance of previous methods, we apply Convolutional Neural Networks in classification and optimize them by studying correlation of results with Feature Density.

3 Proposed Methods

3.1 Data Preprocessing

The dataset used in this research (see sect. 4.1) was in Japanese. In transcription of Japanese language, spaces (" ") are not used. Therefore we needed to preprocess the dataset and make the sentences separable into elements for feature extraction. We used MeCab[2], a Japanese morphological analyzer and CaboCha[3], a Japanese dependency structure analyzer to preprocess the dataset in the following ways[4].

- **Tokenization:** All words, punctuation marks, etc. are separated by spaces (later: `TOK`).
- **Lemmatization:** Like the above but the words are represented in their generic (dictionary) forms, or "lemmas" (later: `LEM`).
- **Parts of speech:** 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`).
- **Lemmas with POS:** Like the above but with lemmas instead of words (later: `LEM+POS`).
- **Tokens with Named Entity Recognition:** Words encoded together with with information on what named entities (private name of a person, organization, numericals, etc.) appear in the sentence. The NER information is annotated by CaboCha (later: `TOK+NER`).
- **Lemmas with NER:** Like the above but with lemmas (later: `LEM+NER`).
- **Chunking:** Larger sub-parts of sentences separated syntactically, such as noun phrase, verb phrase, predicates, etc., but without dependency relations (later: `CHNK`).
- **Dependency structure:** Same as above, but with information regarding syntactical relations between chunks (later: `DEP`).
- **Chunking with NER:** Information on named entities is encoded in chunks (later: `CHNK+NER`).
- **Dependency structure with Named Entities:** Both dependency relations and named entities are included in each element (later: `DEP+NER`).

Five examples of preprocessing are represented in Table 2. Theoretically, the more generalized a sentence is, the less unique and frequent patterns it will contain, but the produced patterns will be more frequent (e.g., there are more `ADJ N` patterns than "pleasant day"). We compared the results for different preprocessing methods to find out whether it is better to represent sentences as more generalized or specific.

3.2 Feature Extraction

From each of the eleven dataset versions a Bag-of-Words language model is generated, producing eleven different models (Bag-of-Words/Tokens, Bag-of-Lemmas, Bag-of-POS, Bag-of-Chunks, etc.). Sentences from the dataset processed with those models are used later in the input layer of classification. We also applied traditional weight calculation scheme, namely term frequency with inverse document frequency (tf*idf). Term frequency $tf(t,d)$ refers here to the traditional

[2] http://taku910.github.io/mecab/

[3] https://taku910.github.io/cabocha/

[4] Performance of MeCab is reported around 95-97% [Mori and Neubig, 2014], and Cabocha around 90% [Taku Kudo, 2002] for normal language. Although we acknowledge that in some cases the language of the Web could cause errors in POS tagging and word segmentation, we did not want to retrain the basic tools to fit our data because we wanted the method to work using widely available resources, so it was easily reproducible. Also, we assumed that even if such errors occur, as long as they are systematic, they will not cause trouble.

Table 2: Three examples of preprocessing of a sentence in Japanese; N = noun, PP = postpositional particle, ADV = adverb, ADJ = adjective, AUX = auxiliary verb, SYM = symbol, 1D, 2D, ... = depth of dependency relation, *0, *1, *2, ... = phrase number.

Sentence: 今日はなんて気持ちいい日なんだ！
Transcription in alphabet: *Kyōwanantekimochiiihinanda!*
Glosses: Today TOP what pleasant day COP EXCL
Translation: What a pleasant day it is today!

Preprocessing examples

–TOK: *Kyō | wa | nante | kimochiii | hi | nanda | !*
–POS: `N | PP | ADV | ADJ | N | AUX | SYM`
–TOK+POS: Kyō_N|wa_PP|nante_ADV|kimochi_ii_ADJ| hi_N| nanda_AUX|!_SYM
–CHNK: Kyō_wa | nante | kimochi_ii | hi_nanda!
–DEP: ⋆0_3D_Kyō_wa | ⋆1_2D_nante | ⋆2_3D_kimochi_ii | ⋆3_−1D_hi_nanda!

raw frequency, meaning the number of times a term t (word, token) occurs in a document d. Inverse document frequency $idf(t,D)$ is the logarithm of the total number of documents $|D|$ in the corpus divided by the number of documents containing the term n_t. Finally, $tf*idf$ refers to term frequency multiplied by inverse document frequency as in equation 1.

$$idf(t,D) = log\frac{|D|}{n_t} \tag{1}$$

3.3 Classification methods

SVM or Support-vector machines [Cortes and Vapnik, 1995] are a set of classifiers well established in AI and NLP. SVM represent data, belonging to specified categories, as points in space, and find an optimal hyperplane to separate the examples from each category. SVM were often used in cyberbullying detection (see Table 1). We used four types of SVM functions, namely, linear - the original function which finds the maximum-margin hyperplane dividing the samples; plynomial kernel, in which training samples are represented in a feature space over polynomials of the original variables (also used in [Dinakar *et al.*, 2012]); radial basis function (RBF) kernel, which approximates multivariate functions with a single univariate function, further radialised to be used in higher dimensions; and sigmoid, i.e., hyperbolic tangent function [Lin and Lin, 2003].

Naïve Bayes classifier is a supervised learning algorithms applying Bayes' theorem with the assumption of a strong (naive) independence between pairs of features, traditionally used as a baseline in text classification tasks.

kNN or the k-Nearest Neighbors classifier takes as input k-closest training samples with assigned classes and classifies input sample to a class by a majority vote. It is often applied as a baseline, next to Naïve Bayes. Here, we used k=1 setting in which the input sample is assigned to the class of the first nearest neighbor.

JRip also known as Repeated Incremental Pruning to Produce Error Reduction (RIPPER) [Cohen, 1995], which learns rules incrementally to further optimize them. It is efficient in classification of noisy text [Sasaki and Kita, 1998] and for this purpose was used in cyberbullying detection previously [Dinakar *et al.*, 2012].

J48 is an implementation of the C4.5 decision tree algorithm [Quinlan, 1993], which firstly builds decision trees from a labeled dataset and each tree node selects the optimal splitting criterion further chosen to make the decision.

Random Forest in training phase create multiple decision trees to output the optimal class (mode of classes) in classification phase [Breiman, 2001]. An improvement of RF to standard decision trees is their ability to correct overfitting to the training set common in decision trees [Hastie *et al.*, 2013].

SPEC or **S**entence **P**attern **E**xtraction ar**C**hitecture [Ptaszynski *et al.*, 2015a] is a custom feature extraction and classification system. The features are defined as ordered combinations of sentence elements and contain patterns of tokens and n-grams with disjoint elements. The way the features are extracted (combinatorial approach) resembles brute-force search algorithms. Pattern occurrences for each side of binary class dataset are used to calculate normalized weight of patterns. Next, the score of a sentence is calculated as a sum of weights of patterns found in input sentence. With multiple modifications, such as deletion of ambiguous patterns, or various dataset preprocessing [Ptaszynski *et al.*, 2015b] were able to optimize the method to achieve somewhat high results, and has been considered as best performing method so far for cyberbullying detection.

In comparison we used their results optimized either for F1 or BEP (break-even point of Precision and Recall).

CNN or Convolutional Neural Networks are a type of feed-forward artificial neural network are an improved neural network model, i.e., multilayer perceptron. Although originally, CNN were designed for image recognition, their performance has been confirmed in many tasks, including NLP [Collobert and Weston, 2008] and sentence classification [Kim, 2014].

We applied a Convolutional Neural Network implementation with Rectified Linear Units (ReLU) as a neuron activation function [Nair and Hinton, 2010], and max pooling [Scherer *et al.*, 2010], which applies a max filter to non-overlying sub-parts of the input to reduce dimensionality and in effect correct over-fitting. We also applied dropout regularization on penultimate layer, which prevents co-adaptation of hidden units by randomly omitting (dropping out) some of the hidden units during training [Hinton *et al.*, 2012].

We applied two version of CNN. First, with one hidden convolutional layer containing 100 units was applied as a proposed baseline. Second, the final proposed method consisted of two hidden convolutional layers, containing 20 and 100 feature maps, respectively, both layers with 5x5 size of patch and 2x2 max-pooling, and Stochastic Gradient Descent [LeCun *et al.*, 2012].

4 Evaluation Experiments

4.1 Dataset

As the dataset for experiments we used the one created originally by [Ptaszynski *et al.*, 2010], and also widely used by [Nitta *et al.*, 2013; Ptaszynski *et al.*, 2015a; 2015b; 2016]. It contains 1,490 harmful and 1,508 non-harmful entries in Japanese collected from unofficial school Web sites and fora. The original data was provided by the Human Rights Research Institute Against All Forms for Discrimination and Racism in Mie Prefecture, Japan[5]. The harmful and non-harmful sentences were collected and manually labeled by Internet Patrol members (expert annotators) according to instructions included in the governmental 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 (also initials and nicknames), names of institutions and affiliations, private information (address, phone numbres, entries revealing personal information, etc.)

Moreover, literature on cyberbullying indicates vulgarities as one of the most distinctive features of cyberbullying [Patchin and Hinduja, 2006; Hinduja and Patchin, 2014; Ptaszynski *et al.*, 2010]. Also according to MEXT vulgar language is distinguishable for cyberbullying, due to its ability to convey offenses against particular persons. 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 3.

4.2 Experiment Setup

The preprocessed original dataset provides eleven separate datasets for the experiment see sect. 3.1 for details). Thus the experiment was performed eleven times, one time for each kind of preprocessing. Each of the classifiers (sect. 3.3) was tested on each version of the dataset in a 10-fold cross validation procedure The results were calculated using standard Precision (P), Recall (R), balanced F-score (F1) and Accuracy (A) As for the winning condition, we looked at which classifier achieved highest balanced F-score.

Feature Density

To get a better grasp on the results we also analyzed the influence of how a dataset was preprocessed on the results. A dataset is the more generalized, the fewer number of frequently appearing unique features it produces. Therefore to estimate dataset generalization level we applied the notion of Lexical Density (LD) [Ure, 1971]. It is a score representing an estimated measure of content per lexical units for a given corpus, calculated as the number of all unique words divided by the number of all words in the corpus. Since in our research we use a variety of different features, not only words, we will further call this measure *Feature Density* (FD).

After calculating FD for all used datasets we calculated Pearson's correlation coefficient (ρ-value) to see if there is any correlation between dataset generalization (FD) and the results (F-scores).

4.3 Results and Discussion

All results were summarized in Table 4. The results of the baselines (kNN, Naïve Bayes) were low, as assumed. Although these classifiers can be tuned to high scores in typical sentiment analysis, they were not able to grasp the noisy language used in cyberbullying. However, it must be noticed that, especially with the help of named entities (NER), NB performed rather well, comparably to J48 or JRip.

When it comes to decision trees-based classifiers, J48 scored low, similarly as in [Dinakar *et al.*, 2012]. However, Random Forest usually scored better even than SPEC. Unfortunately, RF is highly time-inefficient, especially compared to SVM, and thus impractical.

In many previous research on CB detection SVM were most commonly used with various success. As we can observe, choice of appropriate function with good preprocessing makes SVM comparable even to the proposed CNN. The best setting was linear-SVM trained on lemmatized dataset (F1=.825). Moreover, although not scoring the highest, when the ratio of time-performance to the results is considered, SVM can be considered as the most efficient classifier[6].

As for the method of preprocessing, most often TOK+NER and LEM+NER scored highest. This can be explained by the fact that the data, which was annotated by expert annotators following official governmental definition of cyberbullying, often contained revealing of private information. As named entity recognition covered most of these cases, it is reasonable that it helped extracting meaningful features. Only for SPEC the results for the two above settings were not available since [Ptaszynski *et al.*, 2015b] did not apply them in their research.

The best so far method, SPEC, was in fact scoring high, second best after the proposed here CNN. SPEC was also better then SVM on every dataset, except one (LEM). Although SPEC is highly time inefficient in the training phase (generation of all combinatorial patterns), it is easy to implement and even in its fresh-trained form can be applied without any additional packages to any external media. This could be an advantage when including it in CB detection software, such as smartphone application, etc.

When it comes to the proposed method, the initial baseline-CNN with only one hidden layer did not perform well, although still was better than the baselines and comparable to most of the classifiers.

However, the final proposed method, namely, the CNN with two hidden layers, 5x5 patch size, max-pooling and Stochastic Gradient Descent, outperformed all of the classi-

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

[6] Training SVM, and simple classifiers (NB, kNN) was blazing fast (several seconds). Simple CNN, Random Forest and JRip was slower (several minutes to 1-2 hours). SPEC and CNN-2L were the longest (about one week).

Table 3: Three 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.

>>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 4: Results of all applied classifiers (Scores averaged for positive and negative prediction calculated separately; best classifier for each dataset in **bold type fond**; best dataset generalization for each classifier – underlined).

		LEM +POS	TOK +POS	LEM	TOK	CHNK +NER	POS	DEP +NER	DEP	CHNK	LEM +NER	TOK +NER
kNN (k=1)	**Prec**	.639	.630	.644	.630	.578	.544	.593	.628	.576	.668	.663
	Rec	.636	.627	.640	.628	.546	.543	.529	.528	.550	.663	.659
	F1	.633	.625	.637	.626	.505	.542	.446	.427	.494	.658	.656
	Acc	.636	.627	.640	.628	.546	.543	.529	.528	.550	.663	.659
Naïve Bayes	**Prec**	.678	.671	.686	.682	.666	.570	.652	.672	.685	.708	.703
	Rec	.674	.669	.682	.678	.627	.569	.578	.555	.598	.705	.701
	F1	.673	.668	.681	.677	.599	.568	.511	.453	.539	.705	.701
	Acc	.674	.669	.682	.678	.627	.569	.578	.555	.598	.705	.701
JRip	**Prec**	.606	.614	.604	.603	.628	.553	.643	.505	.685	.699	.700
	Rec	.606	.613	.603	.603	.555	.553	.533	.510	.598	.675	.672
	F1	.606	.613	.603	.603	.469	.553	.408	.345	.539	.663	.658
	Acc	.606	.613	.603	.603	.555	.553	.533	.510	.598	.675	.712
J48	**Prec**	.672	.671	.683	.675	.615	.566	.652	.260	.645	.711	.707
	Rec	.671	.666	.681	.672	.548	.566	.533	.510	.517	.708	.704
	F1	.670	.663	.680	.669	.458	.566	.408	.344	.365	.706	.707
	Acc	.671	.666	.681	.672	.548	.566	.533	.510	.517	.708	.663
Random Forest	**Prec**	.816	.803	.818	.809	.662	.547	.623	.619	.639	.818	.809
	Rec	.809	.795	.809	.801	.632	.546	.607	.582	.580	.809	.802
	F1	.808	.794	.808	.799	.610	.544	.590	.540	.522	.807	.800
	Acc	.809	.795	.809	.801	.632	.546	.607	.582	.580	.809	.802
SVM linear	**Prec**	.777	.768	**.827**	.777	.679	.563	.651	.639	.606	.820	.781
	Rec	.777	.766	**.825**	.776	.645	.563	.615	.577	.603	.818	.781
	F1	.776	.766	**.825**	.775	.623	.563	.586	.531	.597	.818	.780
	Acc	.777	.766	**.825**	.776	.645	.563	.615	.577	.603	.818	.781
SVM plynomial	**Prec**	.262	.499	.262	.263	.260	.553	.260	.260	.260	.260	.260
	Rec	.512	.499	.512	.513	.510	.545	.510	.510	.510	.510	.510
	F1	.346	.450	.347	.348	.344	.528	.344	.344	.344	.344	.344
	Acc	.512	.499	.512	.513	.510	.545	.510	.510	.510	.510	.510
SVM radial	**Prec**	.797	.753	.262	.793	.260	.565	.260	.260	.260	.752	.779
	Rec	.771	.747	.512	.756	.510	.565	.510	.510	.510	.516	.778
	F1	.765	.746	.347	.746	.344	.565	.344	.344	.344	.358	.778
	Acc	.771	.747	.512	.756	.510	.565	.510	.510	.510	.516	.778
SVM sigmoid	**Prec**	.757	.746	.262	.752	.260	.562	.260	.260	.260	.260	.771
	Rec	.549	.736	.512	.538	.510	.562	.510	.510	.510	.510	.606
	F1	.425	.733	.347	.403	.344	.561	.344	.344	.344	.344	.530
	Acc	.549	.736	.512	.538	.510	.562	.510	.510	.510	.510	.606
SPEC (highest BEP)	**Prec**	.802	.786	.784	.770	.655	.614	.548	.591	.633	N/A	N/A
	Rec	.802	.786	.784	.770	.655	.614	.548	.591	.633	N/A	N/A
	F1	.802	.786	.784	.770	.655	.614	.548	.591	.633	N/A	N/A
	Acc	.775	.780	.600	.765	.790	.635	.525	.640	.548	N/A	N/A
SPEC (highest F1)	**Prec**	.807	.756	.713	.724	.563	**.528**	.500	**.491**	.490	N/A	N/A
	Rec	.798	.839	.885	.842	.879	**.946**	.982	**1.000**	1.000	N/A	N/A
	F1	.803	.796	.790	.778	.686	**.677**	.663	**.658**	.658	N/A	N/A
	Acc	.808	.784	.770	.766	.603	**.550**	.510	**.491**	.490	N/A	N/A
CNN (1 hidden layer)	**Prec**	.700	.678	.724	.704	.506	.544	.616	.499	.496	.720	.739
	Rec	.700	.678	.724	.704	.510	.544	.525	.505	.509	.716	.738
	F1	.700	.677	.724	.704	.433	.544	.393	.461	.374	.714	.738
	Acc	.700	.678	.724	.704	.510	.544	.525	.505	.509	.716	.738
CNN (2 hidden layers)	**Prec**	**.867**	**.824**	.820	**.833**	**.936**	.500	**.929**	N/A	**.899**	**.847**	**.849**
	Rec	**.866**	**.821**	.819	**.831**	**.935**	.500	**.927**	N/A	**.893**	**.847**	**.849**
	F1	**.866**	**.821**	.819	**.830**	**.935**	.495	**.927**	N/A	**.892**	**.847**	**.849**
	Acc	**.866**	**.821**	.819	**.831**	**.935**	.500	**.927**	N/A	**.893**	**.847**	**.849**

Table 5: Left part: Information on features (unique features, Feature Density) for each dataset. Right part: Pearson Correlation Coefficient (ρ-value) of Feature Density and all classifier results, with statistical significance (2-sided p-value).

	Dataset Preprocessing	# unique 1grams	# all 1grams	Feature Density	Classifier	ρ-value	2-sided p-value
Fetaure sophistication high ↑	**DEP**	12802	13957	0.917	**CNN-2L**	0.685	*p=0.035
	DEP+NER	12160	13956	0.871	**SVM-pol**	-0.431	p=0.185
	CHUNK	11389	13960	0.816	**SVM-sig**	-0.534	p=0.091
	CHUNK+NER	10657	13872	0.768	**SPEC-BEP**	-0.550	p=0.133
	TOK+POS	6565	34874	0.188	**RandForest**	-0.560	p=0.073
	TOK	6464	36234	0.178	**SVM-lin**	-0.564	p=0.076
	LEM+POS	6227	36426	0.171	**SPEC-F1**	-0.636	p=0.066
	LEM	6103	36412	0.168	**SVM-rad**	-0.639	*p=0.034
	TOK+NER	5967	38672	0.154	**CNN-1L**	-0.709	*p=0.019
	LEM+NER	5581	38672	0.144	**JRip**	-0.729	*p=0.011
↓ low	**POS**	13	26650	0.001	**NB**	-0.736	*p=0.013
					J48	-0.791	**p=0.006
				$*p \leq 0.05$, $**p \leq 0.01$ →	**kNN**	-0.809	**p=0.004

fiers in almost all settings. The only situation where SPEC scored higher (only POS features) reveals well known characteristics of Neural Nets, which perform poorly on a small number of unique features. As for the second situation, using only dependency features (DEP), which on the other hand contained the largest number of features, generation of the model was not feasible on the applied computer, and thus the results were not calculated. In the near future we plan to repeat the experiment in a more efficient environment, such as a cloud computing service (Google Cloud Platform[7], Microsoft Azure[8], or Amazon EC2[9]).

As for the top three best performing settings, 2-layer CNN trained on chunks alone scored high, close to 90% of F-score. Dependency features with NER was second best with F1=92.7%. However, the most optimal setting was the 2-layer CNN trained on chunks with named entities and reached F-score equal 93.5%, which is a result far more satisfying then expected, and exceeds second non-NN classifier (SVM on lemmas) over 10-percentage points.

Next, we analyzed the correlation of data preprocessing with Feature Density (FD). The results were represented in table 5.

The results clearly divided the classifiers into three groups. First group of the lowest performing classifiers (kNN, NB,

[7] https://cloud.google.com/

[8] https://azure.microsoft.com

[9] https://aws.amazon.com/ec2/

J48, JRip, CNN-1L) was strongly negatively correlated with FD, which means these classifiers lose their general performance the more feature-dense is the model. This suggests that such classifiers should be fed with a feature set of limited density.

Second group contained the classifiers (all SVMs, Random Forest and both SPEC) that performed somewhat high. Their correlation with FD was negative from weak (-0.431) to somewhat high (-0.639). This, supported by the lack of statistical significance, means that FD is does not correlate well with such classifiers and some other characteristics should be used for optimization of dataset preprocessing used in those classifiers.

Finally, we made an interesting discovery about the correlation between FD and our proposed method (2-layer CNN). The classifier correlated positively nearly strongly with FD. This suggests that the performance could be improved by increasing the feature density of the applied dataset. We plan to follow this path in the nearest future.

5 Conclusions and Future Work

In this paper we presented our research on cyberbullying (CB) detection. Cyberbullying has become a serious problem in modern society always connected to the Internet. Manual measures, such as Internet Patrol, have been undertaken to deal with CB, unfortunately, reading through the whole Internet to find CB entries is like looking for a needle in the haystack, while keeping the CB victims exposed to harmful messages leads to serious consequences.

To help quickly respond to ever-growing CB problem, automatic cyberbullying detection research has started to sprout, unfortunately, the results have been only partially satisfying. We proposed a Deep Learning approach to the problem, based on Convolutional Neural Networks (CNN).

The proposed optimized CNN model not only outperformed other classifiers by over 11-percentage-points, scoring a close to ideal F-score (93.5%), but also revealed an unusual characteristics, by nearly strongly positively correlating with Feature Density. This provides an informative hint on how to improve further not only the proposed method (by increasing FD of dataset), but also other classifiers (decreasing FD, etc.).

In the near future we plan to test the limits of potential optimization, also by applying different dataset preprocessing methods (sentiment, etc.), and different language models (n-gram, skip-gram, language combinatorics, etc.). We also plan to implement the developed model into a smartphone application for "in-the-field" testing, and further practical research on cyberbullying and ways of its mitigation.

References

[Breiman, 2001] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[Bull, 2010] Glen Bull. The always-connected generation. *Learning & Leading with Technology*, 38(3):28–29, 2010.

[Cano Basave *et al.*, 2013] Amparo Elizabeth Cano Basave, Yulan He, Kang Liu, and Jun Zhao. A weakly supervised bayesian model for violence detection in social media. 2013.

[Cohen, 1995] William W Cohen. Fast effective rule induction. In *Proceedings of the twelfth international conference on machine learning*, pages 115–123, 1995.

[Collobert and Weston, 2008] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM, 2008.

[Cortes and Vapnik, 1995] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[Cross *et al.*, 2009] Donna Cross, Therese Shaw, Lydia Hearn, Melanie Epstein, Helen Monks, Leanne Lester, and Laura Thomas. Australian covert bullying prevalence study. 2009.

[Dinakar *et al.*, 2012] Karthik Dinakar, Birago Jones, Catherine Havasi, Henry Lieberman, and Rosalind Picard. Common sense reasoning for detection, prevention, and mitigation of cyberbullying. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 2(3):18, 2012.

[Hasebrink *et al.*, 2008] Uwe Hasebrink, Sonia Livingstone, and Leslie Haddon. Comparing children's online opportunities and risks across europe: Cross-national comparisons for eu kids online. 2008.

[Hastie *et al.*, 2013] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics. 2013.

[Hinduja and Patchin, 2010] Sameer Hinduja and Justin W Patchin. Bullying, cyberbullying, and suicide. *Archives of suicide research*, 14(3):206–221, 2010.

[Hinduja and Patchin, 2014] Sameer Hinduja and Justin W Patchin. *Bullying beyond the schoolyard: Preventing and responding to cyberbullying*. Corwin Press, 2014.

[Hinton *et al.*, 2012] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0580, 2012.

[Kim, 2014] Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746—-1751, 2014.

[Kontostathis *et al.*, 2013] April Kontostathis, Kelly Reynolds, Andy Garron, and Lynne Edwards. Detecting cyber bullying: query terms and techniques. In *Proceedings of the 5th annual acm web science conference*, pages 195–204. ACM, 2013.

[Kowalski and Limber, 2007] Robin M Kowalski and Susan P Limber. Electronic bullying among middle school students. *Journal of adolescent health*, 41(6):S22–S30, 2007.

[Lazuras *et al.*, 2012] Lambros Lazuras, Jacek Pyżalski, Vassilis Barkoukis, and Haralambos Tsorbatzoudis. Empathy and moral disengagement in adolescent cyberbullying: Implications for educational intervention and pedagogical practice. 2012.

[LeCun *et al.*, 2012] Yann A LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–48. Springer, 2012.

[Lin and Lin, 2003] Hsuan-Tien Lin and Chih-Jen Lin. A study on sigmoid kernels for svm and the training of non-psd kernels by smo-type methods. *submitted to Neural Computation*, pages 1–32, 2003.

[MEXT, 2008] MEXT. *'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). 2008.

[Mori and Neubig, 2014] Shinsuke Mori and Graham Neubig. Language resource addition: Dictionary or corpus? In *LREC*, pages 1631–1636, 2014.

[Nair and Hinton, 2010] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[Nitta *et al.*, 2013] Taisei Nitta, Fumito Masui, Michal Ptaszynski, Yasutomo Kimura, Rafal Rzepka, and Kenji Araki. Detecting cyberbullying entries on informal school websites based on category relevance maximization. In *IJCNLP*, pages 579–586, 2013.

[Patchin and Hinduja, 2006] J. W. Patchin and S. Hinduja. Bullies move beyond the schoolyard a preliminary look at cyberbullying. *Youth violence and juvenile justice*, 4(2):148–169, 2006.

[Ptaszynski *et al.*, 2010] M. Ptaszynski, Dybala P., Matsuba T., Masui F., Rzepka R., Araki K., and Momouchi Y. In the service of online order: Tackling cyber-bullying with machine learning and affect analysis. *International Journal of Computational Linguistics Research*, 1(3):135–154, 2010.

[Ptaszynski *et al.*, 2015a] Michal Ptaszynski, Fumito Masui, Yasutomo Kimura, Rafal Rzepka, and Kenji Araki. Brute force works best against bullying. In *IJCAI 2015 Workshop on Intelligent Personalization (IP 2015)*, pages 28–29, 2015.

[Ptaszynski *et al.*, 2015b] Michal Ptaszynski, Fumito Masui, Yasutomo Kimura, Rafal Rzepka, and Kenji Araki. Extracting patterns of harmful expressions for cyberbullying detection. In *Proceedings of 7th Language and Technology Conference(LTC'15)*, pages 370–375, 2015.

[Ptaszynski *et al.*, 2016] M. Ptaszynski, F. Masui, T. Nitta, S. Hatakeyama, Y. Kimura, R. Rzepka, and K. Araki. Sustainable cyberbullying detection with category-maximized relevance of harmful phrases and double-filtered automatic optimization. *International Journal of Child-Computer Interaction*, 8:15–30, 2016.

[Pyżalski, 2012] Jacek Pyżalski. From cyberbullying to electronic aggression: Typology of the phenomenon. *Emotional and behavioural difficulties*, 17(3-4):305–317, 2012.

[Quinlan, 1993] J.R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann series in machine learning. Morgan Kaufmann Publishers, 1993.

[Rutter, 1987] D.R. Rutter. *Communicating by Telephone*. International Series in Experimental Social Psychology, Vol 15. Elsevier Science Limited, 1987.

[Sarna and Bhatia, 2015] Geetika Sarna and MPS Bhatia. Content based approach to find the credibility of user in social networks: an application of cyberbullying. *International Journal Of Machine Learning and Cybernetics*, pages 1–13, 2015.

[Sasaki and Kita, 1998] Minoru Sasaki and Kenji Kita. Rule-based text categorization using hierarchical categories. In *Systems, Man, and Cybernetics, 1998. 1998 IEEE International Conference on*, volume 3, pages 2827–2830. IEEE, 1998.

[Scherer *et al.*, 2010] Dominik Scherer, Andreas Müller, and Sven Behnke. Evaluation of pooling operations in convolutional architectures for object recognition. In *International Conference on Artificial Neural Networks*, pages 92–101. Springer, 2010.

[Sood *et al.*, 2012] Sara Owsley Sood, Elizabeth F Churchill, and Judd Antin. Automatic identification of personal insults on social news sites. *Journal of the American Society for Information Science and Technology*, 63(2):270–285, 2012.

[Sourander *et al.*, 2010] A. Sourander, A.B. Klomek, M. Ikonen, J. Lindroos, T. Luntamo, M. Koskelainen, T. Ristkari, and H. Helenius. Psychosocial risk factors associated with cyberbullying among adolescents: A population-based study. *Archives of general psychiatry*, 67(7):720–728, 2010.

[Taku Kudo, 2002] Yuji Matsumoto Taku Kudo. Japanese dependency analysis using cascaded chunking. In *CoNLL 2002: Proceedings of the 6th Conference on Natural Language Learning 2002 (COLING 2002 Post-Conference Workshops)*, pages 63–69, 2002.

[Turney, 2002] Peter D Turney. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of ACL 2002*, pages 417–424. Association for Computational Linguistics, 2002.

[Ure, 1971] J. Ure. Lexical density and register differentiation. *Applications of Linguistics*, pages 443–452, 1971.

[Zheng, 2012] R. Zheng. *Evolving Psychological and Educational Perspectives on Cyber Behavior*. Premier reference source. Information Science Reference, 2012.