Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 176 (2020) 612–621

Procedia Computer Science

www.elsevier.com/locate/procedia

24th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems

Combining textual features to detect cyberbullying in social media posts

Meisy Fortunatus[a], Patricia Anthony[a]*, Stuart Charters[a]

[a]Lincoln University, Ellesmere Junction Road, Lincoln 7647, New Zealand

Abstract

Cyberbullying has become prevalent in social media communication. To create a safe space for cyber communication, an effective cyberbullying detection method is needed. This study focuses on using combination of textual features to detect cyberbullying across social media platforms. Lexicon enhanced rule-based method was applied to detect cyberbullying on Facebook comments. The resulting algorithm was evaluated using performance measures of accuracy, precision, recall, and F1 Score, and showed promising performance with average recall of 95.981%.

Keywords: Cyberbullying & Cyber aggression;Textual aggression;Sentiment analysis;Emoticon sentiment;Emoji sentiment;Aggression detection;

1. Introduction

The growth of internet users over the years has brought many developments, both positive and negative. One of the most prevalent issues is cyberbullying. Cyberbullying can be defined as a voluntary act of violating another human through technological communication media [8, 26]. Harassment or aggression that happens through emails, cellular phone, online game, and social media can be considered as cyber aggression or cyberbullying.

Though many dismiss cyberbullying cases as insignificant, previous studies proved that it is actually just as harmful as traditional bullying [20] with effects such as school issues [8], depression [2], substance abuse [8], and even suicide attempts [2] in the worst cases. Despite this, victims are usually reluctant to report cases [4, 22] because of reasons such as: not wanting the electronic devices to be taken away [24, 18], fear of not being taken seriously

* Corresponding author. *E-mail address: patricia.anthony@lincoln.ac.nz*

10.1016/j.procs.2020.08.063

[18], fear that nobody can stop the aggression from happening [23], and even believing that the perpetrator will face any consequences [20].

Online social media has been such a great help in reconnecting people, building networks, and providing a space for freedom of speech. Ever since online social media has become a common way to interact with people, cyberbullying can be perceived as an extension of unacceptable social behaviours in offline intrapersonal relationships [27].

This study offers a more grammatical approach to textual cyberbullying detection by combining textual features together, including emoticon and emoji. This method is expected to yield a better performance compared to other methods.

2. Related Works

Cyberbullying detection can be considered as an extension of sentiment analysis tasks. Sentiment analysis can be separated into three main types: rule based method, lexicon based method, and machine learning. Rule based method on sentiment analysis task can be defined as a method to determine a sentiment of an input by enforcing a predefined set of rules [6]. Rule based methods have been applied to past studies on cyberbullying detection because the rules can be tailored to detect certain elements such as presence of swear words, insults, and excessive capitalization.

Reynolds, Kontostathis, and Edwards [21] conducted a study on cyber aggression detection using two different methods: rule-based learning and bag-of-words model, using data extracted from Formspring.me, a social media to ask 1-on-1 questions. The defined rules consisted of presence of bad words and anonymity, because they argued that anonymity might promote user's tendency to bully or harass. Although this study successfully detect cyberbullying, the high number of false positive was an issue.

Based on past studies, features like swear words, insults, excessive capitalization, and second person pronoun connection are common signs of aggression because bad words are usually used with ill intentions and constant capitalization is considered offensive and rude [17, 3]. Although rule-based method is robust and straightforward, it tends to be too broad and causing a high number of false positives, hence a refined approach is needed to compile the rules [6].

A combination of three different features was used in a cyberbullying detection research [27]: local, sentiment, and contextual. Local features were described as features extracted from the post itself, assessed using TFIDF (term frequency–inverse document frequency) rule. Sentiment feature is calculated by establishing a set of rules to detect pattern of harassment text, which highlighted the connection between bad words and personal pronouns. Finally, contextual feature was used to discern the context of the input, to anticipate non-aggressive inputs such as jokes between friends and a heated argument. The research found that utilizing combined features resulted in better performance compared to elementary TFIDF.

A research [5] used a combination of features to detect aggressive texts, including document length with fuzzy rules, number of bad words, second person pronoun frequency, NS score, ANEW score, and SentiWordNet score. NS score was calculated based on the occurrence of swearwords taken from noswearing.com, while ANEW (Affective Norms for English Words) was derived from words that show positivity. Lastly, a sentiment score was calculated using SentiWordNet lexicon. They reviewed the performance of the system utilizing individual and combined features, and it was presented that the system using linear regression utilizing combination of features performed the best.

Sentiment analysis task, which has been studied for a longer time compared to cyberbullying detection, usually utilizes more comprehensive features such as negation checking and intensifier handling. Applying these features to cyberbullying detection is expected to boost the system's performance. The features include:

- POS Tagging
  Part-of-speech tagging is a task to categorize word to its part-of-speech tag (noun, verb, adjective, adverb, etc.) from a sentence [12]. POS tagging provides information of the sentence structure hence enabling a more linguistic approach.
- Lemmatization

Lemmatization is a pre-processing step that is essential for natural language processing that focuses on reducing a word to its root form [11]. Lemmatization works by utilizing dictionary and the morphological structure of the word itself, meaning it will not falsely chop a word [11]. However, it will only reduce words found in the dictionary, making it limited in scope.

- Negation Handling

  Negation is very common in sentences. There are two important parts in negation handling: finding a negation point and how to handle a negation upon finding it. Finding negation point can be done by performing backwards search, the limit can be fixed such as a maximum of three words backwards [10] or until a boundary is found [25].

  The most basic approach on how to handle a negation is by reversing the polarity immediately after negation [14, 13]. However some studies presented that shifting the polarity using a predefined constant would be a better way to handle negation in a sentence because bad and not bad are not completely opposite in terms of polarity [25, 10].

- Intensifiers Handler

  There are at least three intensifiers in a text input: intensifying word, punctuation, and capitalization. Intensifying words are words used to intensify another word in a sentence, such as very, quite, and most [13]. Essentially, intensifier words are handled by simple addition or subtraction using a fixed value for every intensifying word[25], a specific value for different intensifying word[13], or even a percentage [25].

  Certain punctuations have also been considered as an intensifier, such as exclamation mark and question mark [10]. Intensification by presence of multiple punctuations would make a negative sentence more negative and positive sentence more positive, meaning it does not shift the polarity of the sentence.

  Sentiment analysis task considers capitalization as a way to emphasize a point, so capitalization does not shift the polarity of a sentence. VADER [10] (Valance Aware Dictionary for sEntiment Reasoning) used an empirically derived constant to intensify the sentiment of a word written in ALL-CAPS.

Detecting aggression in an input text might be affected by ambiguity because of lack of non-verbal cues [27]. A research [9] found that people tend to use emoticons to convey their real feelings, hence the sentiment of emoticons would generally dominate over text sentiment. They compiled an emoticon sentiment lexicon which was used in further research to enhance sentiment analysis task, where it was found that utilizing emoticon improved the performance of the sentiment analysis task.

With the rising popularity of emoji, a study was conducted [15] to examine classification of emoji's sentiment. They collected a large number of tweets with emojis in 13 different languages and had native speakers of each language to manually annotate each tweet. From the annotation they calculated a sentiment score for each emoji and compiled them into a lexicon called the Emoji Sentiment Ranking.

3. Method

3.1. Data Collection and Validation

For this study, the data chosen was taken from Melania Trump's Facebook post comments because Facebook has been one of the most popular social media platforms and Melania Trump, as the wife of the current president of USA, has received a lot of attention from citizens of the USA, who mostly speak English.

Extraction of the data was done using a scraping program written in Python [19], utilizing Facebook's Graph API. 202 random entries were then selected as training dataset. These entries were manually annotated, resulting in 141 aggressive entries and 61 non-aggressive. As a testing dataset, another 403 entries were selected randomly and manually annotated by two native English speakers. The result of manual annotation was then evaluated using Cohen's kappa.

Table 1. Annotation Result.

		Annotator 1	
		Y	N
Annotator 2	Y	210(a)	23(b)
	N	24(c)	146(d)

Table 1 above shows the annotation result, where:

- a is the number of entries marked as aggressive by both annotators
- b is the number of entries marked as aggressive by annotator 2 only
- c is the number of entries marked as aggressive by annotator 1 only
- d is the number of entries marked as non-aggressive by both annotators

Using the annotation result shown in Table 1, Cohen kappa was calculated as 0.76073. The Cohen kappa is deemed reliable as 0.61-0.80 was presented as "substantial" or significant[16].

3.2. Classification Algorithm

The approach selected for this study is lexicon enhanced rule-based method, using various lexica: swearwords lexicon[2], SentiWordNet [1], slang dictionary[3], positive word lexicon [7], emoticon lexicon [9], and Emoji Sentiment Ranking [15]. The method was chosen because of its flexibility in applying a combination of rules in order to classify an input [21, 5, 3].

Generally, the classification process is done in three steps, which starts with text clean up followed by processing and lastly classification. The program was written in Python 3, utilizing Python's NLTK, FuzzyWuzzy and PyEnchant and the program is run from the command line. The expected input is a csv file while the classification result will be written in an XML file for readability.

3.2.1 Text Clean Up

The text clean up stage starts with separating emoji from plain text and emojis Unicode representation from Emoji Sentiment Ranking. The emojis are stored for further processing in single classification stage while the text is extracted for any emoticon using the same technique with emoji extraction but utilizing emoticon lexicon instead of Emoji Sentiment Ranking.

The text will then be cleaned further of non-boundary punctuations. Boundary punctuations are punctuation that serve as grammatical boundary such as comma, semicolon, colon, apostrophe, and so on. All non-boundary punctuations are removed because it is considered irrelevant in this study.

The next step of text clean up stage is text normalization. Unlike formal letters or news, online communication is usually informal and cannot be expected to follow grammatical rules or spelling. In order to be able to properly handle the text in latter stages, it is essential to normalize the "noisy" text data. Text normalization is done in five steps: pronoun spelling resolution, slang resolution, laughter text resolution, elongated character reduction, and similar word replacement.

In informal communication, users are usually too lazy to type pronouns properly thus pronoun shortening is very common. The shortening varies from you to a simple u or even omitting important apostrophe in *I'm* to *im*. It might seem like a minor issue and has been overlooked by past studies, but if it is not handled properly then it will be a liability in Part-of-Speech tagging stage and further semantical processes. To handle this, a replacement algorithm to replace common pronoun spelling mistakes was added to text clean up stage.

[2] Compiled from noswearing.com, bannedwordslist.com, cs.cmu.edu, rsdb.org, and a Master's Thesis[7]

[3] Compiled from noslang.com

The next step of text normalization is slang resolution. A dictionary of known slangs and its respective translation was compiled using data from noslang.com. The dictionary is utilized to search for slangs in a text and then replace it with the correct translation.

The third step is laughter text resolution, which is a function that recognizes laughter text and converts it into a simpler and easily recognizable form for future processing. Similar to pronoun spelling resolution, this is a function that has been overlooked by past studies. The accepted laughter text is variations of *haha*, it could be as simple as *hahahaha* or as exaggerated as *hahahahahhhhhhaa*. Detection is done using regular expression and if it is accepted as a laughter text then the word will be replaced by a laughter cue of *haha* and will not be normalized further. Laughter cue serves as a possible neutralizer in classification stage.

Elongated character reduction is done by utilizing regular expression replace function, reducing repeated characters to a maximum of two repetitions only. The best case of this is reduction of *happpppy* to happy. In most cases, however, the reduction might not be as effective, such as *haaaaaaapppy* which will be reduced to *haappy*, which is not an English word. In case of invalid word after this stage, it will be processed using the last step of normalization: similar word replacement.

Word replacement is a task of finding the most similar word as a replacement to invalid word. PyEnchant library is utilized to suggest an array of similar words to be used to replace the original invalid word. Then using FuzzyWuzzy library's ratio function, the distance between each possible replacement and the original word is calculated. The suggested word with the largest ratio is then selected to replace the original word.

After normalization, the text will then be Part-of-Speech (POS) tagged using NLTK's POS tagger and ready for processing. Then it is followed by stop words removal, using a modified list of stop words from NLTK (pronouns and negators were removed from stop word list). Stop words removal is done after POS tagging to ensure that POS tagger can work as effective as possible, otherwise some words might already be removed and the POS tag will not be correct because the sentence is no longer grammatically sensible.

3.2.2 Text Processing

Cleaned up text, emoji, and emoticons are processed for different scores: text sentiment score, text aggression score, text positive word score, emoji sentiment score, and emoticon sentiment score. The goal of this stage is to produce a series of scores mentioned before to be used in classification stage. The scores will be forwarded to be classified. Separating scoring and classification process was done in order to promote the algorithm's reusability.

Text scores (sentiment, aggression, and positive word) are calculated for each sentence, while emoji and emoticon scores are calculated once for a single input. Since this study is focused on a more grammatical approach, several features are applied in this stage, including modifier handling, but check, and least check. More details regarding these features are to be discussed in later sections.

Emoji Sentiment Score. Emoji Sentiment Ranking listed 969 different emojis with details such as number of occurrences in the study's collected tweets (divided into positive, neutral, and negative occurrences), Unicode representation, emoji grouping (such as arrows, geometric shapes, emotion, animal, etc.), and overall sentiment. Then emojis with less than 5 total occurrences, sentiment score of 0.000, irrelevant emoji groups that do not represent human emotion, and neutral emojis were removed from the list. The final emoji list has 467 unique emojis.

Emoji Sentiment Ranking uses a sentiment score range of -1 to 1, the same with the algorithm's scoring standard so no scaling was needed. If an emoji appears more than once in a single input, its sentiment score will be increased/decreased by 10% every time it reoccurs. This rule was applied in accordance to the assumption that when the same emoji is being used more than once, the originator might be trying to convey a stronger feeling. In order not to greatly affect the original score in each emoji occurrence, 10% was chosen as the constant. The formula used in the algorithm to calculate each emoji's overall sentiment score is as follows:

$$S_{(e_m)} = \sum_{n=1}^{m}\left(S_{(e)} \times \sum_{a=1}^{n} 10^{1-a}\right) \tag{1}$$

Where:

- S_{em} : Aggregated sentiment of multiple occurrences m of emoji e in an input
- S_e : Sentiment score of emoji e taken from Emoji Sentiment Ranking
- m : the number of occurrences of emoji e in an input
- a,n : counter for summation

If there is more than one unique emojis in a single input, then each emoji occurrence will be calculated using Equation (1) and then it will be summed to get the emoji's whole sentiment score. If an input has an emoji that is not listed in the lexicon then it will be ignored.

Emoticon Sentiment Score. Calculating emoticon sentiment score is simpler compared with emoji's. Because emoticon needs to be typed manually, it is not common to use the same emoticon more than once in a single entry, thus the formula to subtly increase the sentiment score for each emoticon's occurrence is not applied.

Emoticon sentiment score is calculated as a sum of sentiment score of each emoticon that can be found in a single input. The lexicon used is an emoticon sentiment lexicon compiled by Hogenboom, Bal et al. (2013), where the scores are either -1, 0, or 1.4. Online license transfer

Text Aggression Score. Text aggression score is calculated using the help of a swearword lexicon containing 1,717 words. It was compiled from various sources such as noswearing.com, bannedwordlist.com, cs.cmu.edu, rsdb.org, and bad words list from Engman's thesis [7]. The list was updated manually during the development period and some words that are considered to be too ambiguous to be listed as swear words such as *America, adult, Africa, Asian, bigger, taboo, toilet* were removed which cannot really be considered as aggression if they are not followed or preceded by explicit insult words.

If a word is found inside the swearword list, the algorithm will firstly check the POS of the word. If it is noun, it will be considered an aggression. This was done because noun is self-explanatory and can only refer to a human being, with or without pronoun. For adjectives and verbs, pronoun checking will be done to decide whether it is a form of aggression or not.

If a word is considered as an aggression, it will be given a score of -1 which can be modified during modifier handling function. Modifier handler function is a shared function used when calculating text aggression score, text sentiment score, and text positive term score. More details regarding this function can be found in later section.

Overall sentence aggression score is calculated as the sum of each word's aggression score, with laughter cue giving a +1 score as a possible neutralizer. If a laughter cue is found in a sentence without aggressive term then the sentence aggression score will be +1.

Positive Word Score. Positive word score is calculated similarly with text aggression score, but instead of a score of -1 for each occurrence, a score of +1 will be given instead. Positive words are words that have positive connotations such as able, agree, prosper, and so on. There are 710 positive words in the list, adapted from Engman's[7] study on cyberbullying detection.

Sentiment Score / Sentiscore. Sentiscore is a score calculated using SentiWordNet as the sentiment lexicon. SentiWordNet was compiled using a synset structure, such that each word in the entry can have one or more synsets or word sense. In order to identify cyberbullying texts, the synset with the most negative sentiment score is chosen when calculating sentiscore.

Modifier Handler Function. Modifier handler function is a function shared between all three text scoring processes and is arguably the most important feature of the algorithm's text processing stage. There are several modifiers included in the study, such as capitalization, modifier words, negation, *least* check, and *but* check.

Excessive capitalization has been considered as an amplifier. Therefore, if a word is written in 50% or more capital letters, the score will be amplified with a scalar. The scalar was adapted from VADER [10], in which they calculated the scalar by comparing sentiment intensity from grammatical and syntactical cues and came up with a scalar of 0.733.

Modifier words are words that can cause shifts in a sentiment laden word's score. In this research, modifier words are divided into increment and decrement words. For example, in capitalization, any modifier word found will modify the affected word's score by an empirically derived scalar from VADER, which is +0.293 for increment and -0.293 for decrement. Searching method for modifier word presence is adapted from SO-CAL [25], where backward steps is done until a boundary is found. A boundary could be a punctuation or a word.

Combining approaches used in VADER and SO-CAL, the distance between modifier word and sentiment laden word will define the effect of the modifier scalar. If the modifier is found directly before a sentiment laden word, then 100% of the scalar will be used in the calculation. If a modifier word is found 2 steps away, the scalar will be reduced by 5%. If the distance is 3 or more, then the scalar will be reduced by 10%.

Negation handling was adapted from VADER's approach [10], where a list of negator words were compiled and used to search for presence of negation. If a negator word is followed by a sentiment laden word, the score will be multiplied using an empirically derived scalar of -0.74. Instead of addition or subtraction, multiplication is applied because of the needs to shift the polarity of the score, so if a word with positive score is negated it will be negative and vice versa.

Least check is actually an extended negation handler. If negator word *least* is preceded by *at* or *very*, it will not be considered a negation. This was done to avoid false negation in sentences such as *He is at least as handsome as Jesse*. Like negation handling, *least* check was inspired by VADER's [10] sentiment analysis approach.

But check is another grammatical approach to strengthen sentiment analysis that is adapted from VADER by [10]. It is a simple yet powerful way to handle the presence of the word *but* by shifting the polarity of words before and after finding *but* in a sentence. Each word score before *but* will be halved and multiplied by 1.5 if it is after the word *but*.

3.2.2 Classification

A single input typically consists of one or more sentences and each sentence has its own text scores (aggression, positive word, and sentiscore). In single type classification stage, the scores generated from previous stage (Text Processing) are utilized to classify whether a single input is an aggression or not.

The final version of the algorithm utilizes a filter-like approach to classify a single input. The sequence of the filtering is as follows:

1. If no aggression present and positive word score is more than 0, then the sentence will get positive score.
2. If the aggression score is negative (meaning there is aggression detected), it will be added with positive word score as possible neutralizer. If the addition results in a score equal or less than 0, the sentence will be scored negative.
3. If there is no aggression present, no positive word score, and sentiscore is less than 0 and less than sentiscore's mean threshold, the sentence score will be negative.
4. If there is no aggression, no positive word, and no sentiscore calculated then the sentence score will be 0 or neutral.
5. Apply punctuation as an intensifier of the sentence overall score.

If any sentence scored less than 0, then emoji and emoticon scores will be added to the sum of negative sentence scores to check for possible indicator of jokes between friends. If after addition it is still less than 0, then the single input will be marked as an aggressive text. The outputs are sentence scores in a list and a verdict of whether or not a single input is aggression.

Table 2. Training set performance measure.

Performance Measure	Score	Other Research Result[21]
Accuracy	71.782	Not Specified
Precision	71.875	30.600
Recall	97.872	40.500
F1 Score	82.883	34.861

4. Result and Discussion

To measure the performance of the algorithm, four measurements of accuracy, precision, recall, and F1 score are used. This section discusses the details of the performance results on both training and testing stage.

4.1 Final Training Performance Review

The performance of the algorithm by the end of training stage, when it was used to process training datasets, is shown in table 2. The algorithm managed to reach a high recall of 97.872 without sacrificing precision score, this meaning the algorithm was able to grasp the essence of cyberbullying detection and avoid false positives.

For comparison, we chose the research with data taken from Formspring.me, which is quite similar with Facebook comments. The research [21] claimed that the software managed to reach a high recall percentage of 96.6, although not without sacrificing the precision percentage. In the most balanced approach that was implemented, the recall was only 40.5% and even then, the precision was still recorded quite low at 30.6%. Comparing that, our algorithm managed to reach a high recall of 89.055% while keeping a high precision of 88.614%. This shows that overall, our algorithm outperformed the past research.

Table 3. Testing data sets performance measures.

Dataset	TP	TN	FP	FN	TotalActualPos	TotalActualNeg	Accuracy	Precision	Recall	F1Score
1	53	28	18	1	54	46	81	74.648	98.148	84.8
2	58	25	17	0	58	42	83	77.333	100	87.218
3	88	44	14	10	98	58	84.615	86.275	89.796	88
						Average	82.872	79.419	95.981	86.673

4.2 Testing Performance Review

This section presents information on how the final algorithm performed against testing dataset. The 356 annotated entries were divided into two groups of 100s and 156 entries. To calculate accuracy, precision, recall, and F1 score, the classification results were separated into four categories:

- True positive (TP): number of inputs correctly identified as aggression
- True negative (TN): number of inputs correctly identified as non-aggressive text
- False positive (FP): number of inputs falsely identified as aggression
- False negative (FN): number of inputs falsely identified as non-aggressive text

The algorithm showed an improved performance compared to when it was used to process the training dataset as shown in Table 3. It achieved 100% recall for the second dataset, meaning it actually managed to correctly identify all cyberbullying entries in the dataset. In terms of accuracy percentage, testing stage measure of 81% - 84.615% is comparably higher than training stage 71.782%.

On average, the algorithm reached an accuracy of 82.872%, higher than the training stage's accuracy percentage. This was also the same for precision and F1 Score, which are higher compared to the scores in training stage. However, there is a slight decline from 97.872% in training stage to 95.981% in testing stage for recall. These comparisons show that the performance improved in overall, while the algorithm's ability to correctly identify cyberbullying entries declined slightly. In summary, it can be argued that the performance remains stable when used against training and testing datasets.

Lastly, a comparison of testing dataset performance to another study's performance record is also needed. Past research [21] reported a high recall of 96.6% with the cost of a very low precision of less than 10% only. Our average recall score is slightly lower with 95.981% but with an improved average precision of 79.419%. If compared to the past research's most balance score, our algorithm still shows an improved performance with average F1 score of 86.876% compared to the past's 34.861%.

5. Conclusion and Future Works

Detecting cyberbullying is an important task, given the growing popularity of social media communication. The algorithm resulting from this study uses combination of text aggression score, text positive word score, text sentiment score, emoticon sentiment score, and emoji sentiment score to classify an input.

The algorithm was validated using performance measures of accuracy, recall, precision, and F1 Score, with F1 Score considered as the most important measure because it shows the more balanced measure on how the algorithm distinguish positive to negative inputs. Looking at the testing datasets performance result, the algorithm shows a promising performance with F1 Score of 86.673% and Recall of 95.981%. The substantial scores show that the algorithm managed to both detect cyberbullying and discern non-cyberbullying input.

To enhance the algorithm in the future, functionality could be added, including, but not limited to, exploring connection between sentences, translating emoji to its representative words or phrase, exploring phrases, compiling a cyberbullying dictionary with more grammatical details, and utilizing elongated words as intensifier.

References

[1] Baccianella, Stefano & Esuli, Andrea & Sebastiani, Fabrizio. (2010). SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining.. *Proceedings of LREC*. 10.

[2] Bauman, S., Russell B. Toomey, & Jenny L. Walker (2013). Associations Among Bullying, Cyberbullying, and Suicide in High School Students. *Journal of Adolescence*, 36(2), 341–350.

[3] Bayzick, J., April Kontosthathis, & Lynne Edwards (2018). Detecting the Presence of Cyberbullying Using Computer Software. Master's thesis, Ursinus College.

[4] Dehue, F., Catherine Bolman, & Trijntje V¨ollink (2008). Cyberbullying: Youngsters ′ Experiences and Parental Perception. *CyberPsychology & Behavior*, 11(2), 217–223.

[5] Del Bosque, L., & Garza, S. (2014). Aggressive Text Detection for Cyberbullying. In *Human-Inspired Computing and Its Applications* (pp. 221–232). Springer International Publishing.

[6] Devika, M., C. Sunitha, & Amal Ganesh (2016). Sentiment Analysis: A Comparative Study on Different Approaches. *Procedia Computer Science*, 87, 44–49.

[7] Engman, L. (2016). Automatic Detection of Cyberbullying on Social Media. Master's thesis, UmeåUniversity, Sweden, June.

[8] Hinduja, S., & Justin W. Patchin (2008). Cyberbullying: An Exploratory Analysis of Factors Related to Offending and Victimization. *Deviant Behavior*, 29(2), 129–156.

[9] Hogenboom, A., Daniella Bal, Flavius Frasincar, Malissa Bal, Franciska de Jong, & Uzay Kaymak (2013). Exploiting Emoticons in Sentiment Analysis. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing* (pp. 703–710). ACM.

[10] Hutto, C.,& Eric Gilbert (2015). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text.

[11] Jivani, A. (2011). A Comparative Study of Stemming Algorithms. *International Journal of Computer Technology and Applications*, 2(6), 1930–1923.

[12] Jurafsky, D., & James H. Martin (2016). Part-of-speech Tagging. *Speech and Language Processing*.

[13] Jurek, A., Maurice D. Mulvenna, & Yaxin Bi (2015). Improved Lexicon-based Sentiment Analysis for Social Media Analytics. *Security Informatics*, 4, 1–13.

[14] Kaushik, C., & Atul Mishra (2014). A Scalable, Lexicon Based Technique for Sentiment Analysis. *International Journal in Foundations of Computer Science & Technology*, 4.

[15] Kralj Novak, P., Jasmina Smailovic, Borut Sluban, & Igor Mozetic (2015). Sentiment of Emojis. In *PloS one*.

[16] Landis, J., & Gary G. Koch (1977). The Measurement of Observer Agreement for Categorical Data. *Biometrics*, 33(1), 159–174.

[17] Nalinipriya, G., & M. Asswini (2015). A Dynamic Cognitive System For Automatic Detection and Prevention of Cyber-bullying Attacks. *ARPN Journal of Engineering and Applied Sciences*, 10(10), 4618–4626.

[18] Parris, L., Kris Varjas, Joel Meyers, & Hayley Cutts (2012). High School Students' Perceptions of CopingWith Cyberbullying. *Youth & Society*, 44(2), 284-306.

[19] Paulo. (2017). How to Scrape Facebook Page Posts and Comments to Excel (with Python). https://nocodewebscraping.com/facebook-scraper/.

[20] Pettalia, J., Elizabeth Levin, & Jo¨el Dickinson (2013). Cyberbullying: Eliciting Harm Without Consequence. *Computers in Human Behavior*, 29(6), 2758–2765.

[21] Reynolds, K., April Kontostathis, & Lynne Edwards (2011). Using Machine Learning to Detect Cyberbullying. *2011 10th International Conference on Machine Learning and Applications and Workshops*, 2, 241–244.

[22] Slonje, R., & Peter Smith (2008). Cyberbullying: Another Main Type of Bullying?. *Scandinavian journal of psychology*, 49, 147–54.

[23] Smith, P., Jess Mahdavi, Manuel Carvalho, Sonja Fisher, Shanette Russell, & Neil Tippett (2008). Cyberbullying: Its Nature and Impact in Secondary School Pupils. *Journal of child psychology and psychiatry, and allied disciplines*, 49(4), 376–385.

[24] Stacey, E. (2009). Research into Cyberbullying: Student Perspectives on Cybersafe Learning Environments. *Informatics in Education*, 8, 115–130.

[25] Taboada, M., Julian Brooke, Milan Tofiloski, Kimberly D. Voll, & Manfred Stede (2011). Lexicon-Based Methods for Sentiment Analysis. *Computational Linguistics*, 37, 267–307.

[26] Whittaker, E., & Robin Kowalski (2014). Cyberbullying Via Social Media. *Journal of School Violence*, 14, 11-29.

[27] Yin, D., & Brian D. Davison (2009). Detection of Harassment on Web 2.0.