FACULTY OF INFORMATION TECHNOLOGY AND ELECTRICAL ENGINEERING

Md Saroar Jahan

CYBER BULLYING IDENTIFICATION AND TACKLING USING NATURAL LANGUAGE PROCESSING TECHNIQUES

Master's Thesis
Degree Programme in Computer Science and Engineering
July 2020

Jahan M. (2020) Cyber Bullying Identification and Tackling Using Natural Language Processing Techniques. University of Oulu, Degree Programme in Computer Science and Engineering, 75 p.

ABSTRACT

As offensive content has a detrimental influence on the internet and especially in social media, there has been much research identifying cyberbullying posts from social media datasets. Previous works on this topic have overlooked the problems for cyberbullying categories detection, impact of feature choice, negation handling, and dataset construction. Indeed, many natural language processing (NLP) tasks, including cyberbullying detection in texts, lack comprehensive manually labeled datasets limiting the application of powerful supervised machine learning algorithms, including neural networks. Equally, it is challenging to collect large scale data for a particular NLP project due to the inherent subjectivity of labeling task and man-made effort.

For this purpose, this thesis attempts to contribute to these challenges by the following. We first collected and annotated a multi-category cyberbullying (10K) dataset from the social network platform (ask.fm). Besides, we have used another publicly available cyberbullying labeled dataset, 'Formspring,' for comparison purpose and ground truth establishment. We have devised a machine learning-based methodology that uses five distinct feature engineering and six different classifiers. The results showed that CNN classifier with Word-embedding features yielded a maximum performance amidst all state-of-art classifiers, with a detection accuracy of 93% for AskFm and 92% for FormSpring dataset. We have performed cyberbullying category detection, and CNN architecture still provide the best performance with 81% accuracy and 78% F1-score on average.

Our second purpose was to handle the problem of lack of relevant cyberbullying instances in the training dataset through data augmentation. For this end, we developed an approach that makes use of wordsense disambiguation with WordNet-aided semantic expansion. The disambiguation and semantic expansion were intended to overcome several limitations of the social media (SM) posts/comments, such as unstructured content, limited semantic content, among others, while capturing equivalent instances induced by the wordsense disambiguation-based approach. We run several experiments and disambiguation/semantic expansion to estimate the impact of the classification performance using both original and the augmented datasets. Finally, we have compared the accuracy score for cyberbullying detection with some widely used classifiers before and after the development of datasets. The outcome supports the advantage of the data-augmentation strategy, which yielded 99% of classifier accuracy, a 5% improvement from the base score of 93%.

Our third goal related to negation handling was motivated by the intuitive impact of negation on cyberbullying statements and detection. Our proposed approach advocates a classification like technique by using NegEx and POS tagging that makes the use of a particular data design procedure for negation detection. Performances using the negation-handling approach and without

negation handling are compared and discussed. The result showed a 95% of accuracy for the negated handed dataset, which corresponds to an overall accuracy improvement of 2% from the base score of 93%.

Our final goal was to develop a software tool using our machine learning models that will help to test our experiments and provide a real-life example of use case for both end-users and research communities. To achieve this objective, a python based web-application was developed and successfully tested.

Keywords: Cyberbullying detection, Disambiguation, Expansion of dataset, Negation Detection.

TABLE OF CONTENTS

FOREWORD

This thesis work is following the research under the Center for Machine Vision and Signale Analysis (CMVS), Faculty of Information Technology, in the field of Natural Language Processing (NLP).

I am thankful to my supervisor Dr. Mourad Oussalah for providing me the opportunity to work with him. While attending courses like Natural Language Processing and Social Networking, which were solely taken by Dr. Mourad, I found myself immensely interested in NLP; particularly in the topic "Hate Speech and Cyberbullying Detection".

During my thesis days while I was gathering cyberbullying dataset, extending it and working for the improvement of cyberbullying detection, I found my supervisor Dr. Mourad Oussalah always by my side with his improvised guidance which motivated me to develop my work everyday. He endlessly supervised me despite the fact that he is an extremely occupied passionate researcher.

I am thankful to Dr. Timo Ojala, (Director, Center for Ubiquitous Computing) and Dr. Denzil Teixiera Ferreira for giving me the opportunity to study at University of Oulu, Finland and to follow my dreams. In addition, I would also like to thank my course coordinator Dr. Anabela Berenguer for always being kind to me.

Finally, I would like to express my deepest appreciation to my parents, siblings, colleagues and friends for their moral support and prayers throughout this journey and to all the researchers who contributed to the topic.

Oulu, July 20th, 2020

Md Saroar Jahan

LIST OF ABBREVIATIONS AND SYMBOLS

CNN	Convolutional neural network
LSTM	Long short-term memory
LR	Linear regression
NB	Naive Bayes
RF	Random forest
SVM	Support Vector Machin
TF-IDF	Term frequency–inverse document frequency
GUI	Graphical user interface
UI	User interface
SM	Social Media
PoS	Part of speech
DL	Deep Learning
ML	Machine Learning
TP	True positives
TN	True negatives
FP	False positives
FN	False negatives
TPR	True positive rate
FPR	False positive rate
VPN	Virtual Private Network

1. INTRODUCTION

Institutions, online communities, and social media platforms have a keen attention to offensive contents that have shown pervasive in social media use. Old-fashioned controlling methods of detecting offensive materials online are becoming impossible to apply since a massive number of posts are being posted every day. One of the most common approaches to tackle the issue is to train systems that can identify offensive contents online and remove them without any human interaction. In the past few years, several studies were done about automated hate speech and cyberbullying detection. It is beyond human's imagination to keep track of all the discussions getting produced online as only Twitter users create more than a million Tweets every day[1]. Many researchers have started to explore automatic procedures for signaling harmful contents. This would allow for large-scale social media monitoring and early detection of adverse situations, including cyberbullying. For automated cyberbullying detection, systems designed using NLP, the most common way is gathering real-life data from social network websites, manually labeling them, and finally processing them using machine learning for the detection process. However, collecting and annotating a large number of dataset is challenging, especially for hate speech or cyberbullying related topics [2]. Besides, the detecting this type of speech is sometimes crucial due to the topic's abstractness, especially for negative sentences. To achieve this, we focused our efforts on- collecting the cyberbullying datasets and manually labeling and processing them, testing them with different features and models to find out the best possible results. We also focused on the extension of datasets by sense disambiguation with WordNet, and PoS tagging, analyzing whether it is feasible to use extended datasets, negation scope detection and negated datasets effect on cyberbullying, and finally developed a GUI interface to justify our system. Below our thesis goal has described.

1.1. Goals

Firstly, we will focus on the preparation of dataset (collection, preprocessing, and labeling) and finding the dataset's ground truth (by comparing the dataset with another similar cyberbullying dataset). To identify cyberbullying offensive posts and categories/types of cyberbullying, we want to conduct an exhaustive experimentation methodology. We aim to select the features that maximize the efficiency of machine learning algorithms, evaluate models, and algorithmic implementation.

The second part of our research is to find a feasible way to enrich our initial datasets and examine whether expanded datasets are good enough for use and capable of detecting cyberbullying. As the augmentation process of initial datasets, we have proposed three methods: WordNet sense disambiguation technique and Lesk-algorithm [1], PoS tagging, and simple word synonyms replacement. Initially, we had two base datasets; therefore, this proposed technique generated two additional extended datasets, which have been again separately used for cyberbullying detection and compared with initial results that we have obtained by using the initial dataset. In the result section, we have shown whether an extended dataset was suitable for cyberbullying detection or not.

Our third goal is finding the negation scope for cyberbullying detection datasets and prepare an algorithm that can identify negative sentences and develop a non-negative datasets. Similar to extended dataset creation, we have compared results of extended datasets with initial results obtained using the base dataset.

Finally, we will develop a functional software toolkit that detects and monitors potential cyberbullying traces from textual and online resources.

1.2. Outline

The dissertation is composed of 7 different chapters. Initially, in our first chapter, we briefly introduced our motivation, goals, intended approaches for cyberbullying detection, dataset extension, and negation scope detection.

The second chapter summarizes the state of the art methods for cyberbullying detection. Initially, we presented general methodologies (dataset preparation, feature engineering, classifier architecture) for cyberbullying detection and later provided a more in-depth look at the results of prepossessing, features selection, and classifications.

In the third chapter, we investigated a general overview of dataset extension, related work about dataset extension and result comparison after using expanded dataset.

In the fourth chapter, we investigated a general overview of negation detection, related work about negation detection, and result comparison after using the negated dataset.

The fifth chapter describes the Graphical User Interface (GUI) that we have implemented.

In the sixth and seventh chapters, we presented a discussion of what we have done in this thesis, whether our goals were concertized or not, and possible future work.

1.3. Language Concern

This thesis work may contain several profane words that were used to describe specific examples of our addressed topic.

2. CYBERBULLYING DETECTION

2.1. Cyberbullying Overview

The incident of cyberbullying, cited as "willful and repeated harm inflicted through the use of computers, cell phones, and other electronic devices," [2] has dramatically increased in the last couple of years, notably among the youth population, primarily due to the evolution in computerized technology.

This circumstance is not only counted deleterious as a personal threat but can harm tremendously by creating social and financial losses to organizations as well. Researchers have investigated that 10-40% of youngsters confessed of having experience with it, either as a victim or as an intimidator where they used the internet and electronic devices to harass, bully, debase, or otherwise irritate their peers [3]. Many web pages have been created, including audio, video, image, and profiles on social media platforms for harassing others. A report by ScanSafe's identified that up to 80% of online blogs contain inappropriate materials, and shockingly 74% of this include pornography in the form of video, image, or vulgar words. The open and private online chat systems and forums have significantly increased the spread of cyberbullying cases.

Not only cyberbullying messages and images can be posted anonymously and distributed rapidly to a vast audience, it can continuously happen all the time regardless of day and night. Besides, unlike physical bullying or traditional bullying, cyberbullying can survive for ages after ages on the internet and can create continuous harm to the victim for a long time. Another challenge is to trace and deleting the source, and often erasing inappropriate or harassing messages, texts, and pictures is impossible after they have posted it or sent it. Even an unintentional activity can spread so fast and get viral that it often becomes massive harassment for the victim. Cyberbullies can hide by being anonymous in the chat rooms. Most of the forums and chat rooms events don't require a real name to get registered or can use services as a guest user too. Therefore, anonymity and the lack of meaningful supervision on the internet are the most common factors that aggravate this social menace.

Fully comprehend cyberbullying is the first step to be able to study such diversity. Thus, in the following sub-chapters, we addressed this concept by providing an overall view of its origins, possible definitions, and why we want to study it.

2.1.1. Origins

The documentation of the 'bully' word is old; it can be traced back as far as the 1530s [4], in its most primary sense bullying required minimum two people, a victim and a bully or intimidator. The bully abuses the victim through verbal, mental, physical, or other means to obtain a sense of superiority and control. These activities may be direct (i.e., beating, verbally attacking in person, etc.) or indirect (i.e., rumors, gossip, etc.).

As technology and the internet has advanced, bullying has increased in many ways. The invention of cell phones in the late 1960s, and early 1970s changed the way people engage in conversation [5]. However, these mobile communication devices did not become universal or accessible for most of the youth's hands until the arrival

of the second generation of digital network phones in the 1990s. After that, the use of cell phones became more easily accessible and expanded like wildfire. Pew Research Center found out 75% of 12-17 year-olds own cell phones, increased by 45% in 2004, and among one-in-three teens sends approximately 3,000 text messages per month [6].

After the growth of second-generation web (web 2.0) in 2004 and with the growth of the social network site like launching of multiple platforms such as Wikipedia (2001), MySpace (2003), Facebook (2004), Orkut (2004) and Twitter (2005), many researchers had stated that cyberbullying was a dangerous phenomenon as offline bullying [7]. Further development of smartphones, especially the release of the first touchscreen phone in 2004, allowed users to roam around with a personal computer in their hands with an all-day power source. Things became more carefree, like sending and taking photos and recording voices. Online communications and discoveries have extended due to telecommunication advances, which rapidly brought numerous platforms, and with this came the arrival of social media. The site, Reddit, Facebook, Twitter, Askfm, and MySpace are often considered the sources of cyberbullying. Now cyberbullying became uncontrollable since every day, millions of social media posts get posted, and we don't know how many of them contain bullies in those posts.

2.1.2. Defination

Bullying and cyberbullying are both abusive conducts whose nature is to detriment another person, community, or organization, which most certainly implies to offensive social behavior. This suggests cyberbullying study may possibly be understood within social psychology. Therefore, some authors works related to moral detachment, obeying authority, power over the situation could be helpful to define bullying and social psychology [8]. Early studies of cyberbullying used their traditional definitions of this phenomenon; most developed an approach based on the explanation of traditional bullying projected by Dan Olweus (1993) [9]. However, a trivial number of them have developed widely, were acknowledged and were cited frequently in new publication (see [8] [10] [11] [12]). These definitions emphasize some underlying aspects of cyberbullying: harming intentionally, repetition over time, and power imbalance between the victim and the perpetrator(s). These meanings have recently developed the subject of a disagreement among professionals and scholars: it is still doubtful whether these benchmarks are relevant to cyberbullying. Therefore, new criteria have been proposed, such as anonymity and publicity [13].

Intention

Due to the secondary nature of cyberbullying, it is very tricky to distinguish the intentions of these behaviors [14]. However, a definition that upholds cyberbullying refers to the use of electronic communication technologies as a platform of intentional, repetitive, and aggressive conduct applied to a person or group to damage others [8]. Yet some dialogues continue as to whether it is essential or not, there is an intention to harm someone if it is required to repeat this behavior, if an imbalance of power must exist. An example of such is, for sufferers, that the vital factor for speaking about cyberbullying is not the intention but the real consequences [15].

Repetition

A general dispute against the use of the standard of repetition is the reality that publishing contents online in itself establishes repetition as they can be seen and forwarded repeatedly [16] [14]. Additionally, cyber contents are often still reachable years after the initial incident. This way, a single action of cyberbullying can drive to countless occurrences of victimization [17]. Although several cyberbullying, such as sending a nasty text message is easy, other categories (such as mimicking someone on a website) expect some more technical skills. Nevertheless, it does not take considerable knowledge to take a photograph of someone to utilize it obnoxiously and posting it onto the Internet for others to see or show around, including friends. Perhaps in particular circumstances [18], superior expertise may enable someone to become more potent than others and so deliberately harm them. However, most of the text messages and online bullying suffered and performed by pupils of school ages and teenagers, and technical expertise is questionably a trivial aspect.

Power imbalance

The failure of a victim to force bully to delete harmful content, higher media expertise, or social status of the bully within a virtual society might be understood as a power imbalance [10] [14]. However, some authors argued this criterion and stated that the sufferer is preferably in a more powerful position than it would be in conventional bullying because they can dismiss harmful interactions easily [19]. A different aspect of power imbalance in cyberbullying has proposed by Dooley, Pyzalski, and Cross (2009) [17]; that while the material exists on the internet, it is tougher to delete or to ignore it, and that this in itself can make the sufferer feel more powerless. Although it is feasible to mount a protection of the criteria of duplication and imbalance of power in the cyberbullying domain, there are undoubtedly complexities. In practice, some workers do not invoke either repetition or imbalance of power as benchmarks to define cyberbullying, which they also describe as ‘internet victimization’ [20].

Anonymity and publicity (new criteria)

Anonymity that happens when the sufferer does not know the identity of the bully may raise emotions of frustration and helplessness [17] [13] and could diminish the need for power imbalance as a benchmark [21]. In previous studies, cyberbullying acts involving a sizeable and public audience were termed as the most dangerous type of cyberbullying [13]. Incorporating these two benchmarks (anonymity and publicity) may indicate cyberbullying more satisfactorily than previous common explanations. However, numerous explanations for cyberbullying resemble one another, and most repeat the bullying description but required electronic means [22] [14]. Smith et al. (2006) characterized cyberbullying as aggressive and intentional action that engages electronic methods of contact committed continuously by an individual or group, which remains steady over time with a victim who cannot naturally protect oneself.

In summary, describing cyberbullying may not be as evident as defining traditional bullying, due to the complexities in the conditions of repetition and power imbalance. These questions, and the scope to which cyberbullying can conveniently be recognized from a more significant idea of cyber aggression or cyber harassment, are being argued.

2.1.3. Common Targets and Example

A critical hypothesis developed based on some social science and psychiatry findings, that cyberbullying case must include both Insult/Swear wording and a Second person's or Person's name [3]. This assumes when Insult/Swear wording and Person Name / Second person occurred together, cyberbullying case is considered. However, such interpretation is not workable from an NLP perspective. As an example, "John Doe is a bad person" is a typical example of cyberbullying as it contains both Insult/Swear word "bad" and Person name "John Doe" as well as a clear correlation between the word and Person's name. Another example, "This is bad" is not cyberbullying since it contains only an Insult/Swear word but no Person name/Second person. However, "This is bad, but John Doe is lucky" includes both Insult/Swear word and Person name; however, it is not a cyberbullying case as the relationship between the two is not established. Nevertheless, "John Doe is not bad" contains both Person name, Insult/Swear word, and there is a correlation between two, but it is not a cyberbullying case due to the existence of negation. Similarly, 'John Doe is not good person' does not contain swear words, but considered cyberbullying. All these examples showed the occurrence of the requirements mentioned above for cyberbullying cases are necessary conditions; though, it is not compulsory due to the variety of natural language modifiers expressing negation and opposition.

Besides for single sentences, cyberbullying could work differently if multiple sentences put together—the paragraph "John doe working hard. Ugly" is a cyberbullying case even though the second sentence "Ugly" contains only an Insult/Swear word without any Second person or Person entity. Still, since it belongs to an earlier verdict, the connection can determine from a reader viewpoint.

The above few cases explain the complications of the task of detection of cyberbullying cases applying standard NLP tools, which involves examining all the textual information of the paragraph.

2.1.4. Traditional Bullying Vs. Cyberbullying

Cyberbullying has been observed to be different from traditional bullying in a variety of ways. A work by Smith (2012) [23] defined seven important features as follow:

1. Cyberbullying depends on some degree of technical proficiency.
2. Mostly indirect rather than face-to-face, and this could be anonymous.
3. The bully does not typically see the victim's short term or instant response.
4. The diversity of perpetrator nature in cyberbullying is more complicated than traditional bullying.
5. The most common motive for traditional bullying is to gain status by showing (offensive) authority over others, in front of witnesses, but the bully will often lack this in cyberbullying.

6. The size of the potential audience is expanded, as cyberbullying can reach a sizable number of viewers in a peer group compared with traditional bullying.

7. For the victim, it is challenging to get away from cyberbullying compared to traditional bullying, as the victim may be sent messages to their smartphone or computer, or access nasty online contents, wherever they are.

Above mentioned points have discovered cyberbullying to diverge from traditional bullying in several ways. However, these are not only definite disagreements [24], but they may influence other characteristics such as motives for the bully, and impact on victims.

2.1.5. Why Study Cyberbullying

Cyberbullying profoundly impacts and undermines the right of the targeted person to equality and freedom. While this is enough motivation to go ahead and fight it, it has proven its consequences are, in the long run, potentially catastrophic if no measures are taken against it. Cyberbullying promotes prejudice and hate and might shake the foundations of societies, creating gaps between social groups, which might lead to deep fractures in the social cohesion. The popularity and continuous growth of online communities has also been contributing to the abundance of hateful behaviors. Being able to post and interact, mostly anonymously or without providing much personal information, acts as an incentive to give away unpopular and hostile opinions without many consequences. Governments and especially social media, have been trying to come up with efficient solutions to avoid Cyberbullying Nobata et al. (2016) [25]; however, the lack of studies and research automatically identifying and detecting these behaviors makes it hard to accomplish significant results. Consequently, it is rather vital to contribute with solutions for automatic Cyberbullying detection in text.

Furthermore, this has negatively impacted organizations and damaged the economy as a whole, putting extra pressure on security officers. The latter face is increasing challenges for various reasons. Example, cyberbullying can occur continuously all day to the entire year and reach a kid when alone.

2.1.6. Cyberbullying Detection: State of the Art

In the filed of automated detection, hate speech related work came first before cyberbullying. Several work has studied offensive language detection by using social network datasets examples, Twitter [26] [27], Askfm [28], Wikipedia comments, Facebook posts [29] , and Fromspring posts [30]. Several academic events and shared task competitions organized in conjunction with high impact data mining, information retrieval and computational linguistics conferences (e.g., Workshop series on Abusive Languages, Automatic Misogyny Identification, Authorship Aggressiveness Analysis, Identification of Offensive Language at GermEval, Hate Speech Detection Task at Evalita, various related SemEval tasks, etc.). In this respect, one of the most commonly employed methodologies is to train systems that can automatically identify offensive

content, which will then trigger action to remove such content without any human moderation.

Past research has also examined various characteristics of offensive language such as the cyber aggression [29], abusive language [25], hate speech [26] [31], cyberbullying [28] [32], Racism [33] and offensive language [27]. Nevertheless, automatic identification of offensive language is challenging, especially given the continuous evolution and variability of offensive language discourse and characteristics, along with the inherent limitations of the NLP-based approaches.

However, Cyberbullying is a widely covered topic in the realm of social sciences and psychology, and contains complexity compare to hate speech. Some research has been done based on the definition and prevalence of phenomenon [34], identification of different forms of cyber-bullying [12], and consequences [11]. In contrast to the efforts made in defining and measuring cyberbullying, the number of studies that focus on its annotation and automatic detection is limited [35], and not much work has done regarding automated detection of cyberbullying categories.

Research by Yin, et. al found out that the baseline approach (using a bag-of-words method) was significantly improved by including sentiment and contextual features. Even with the combined model, a support vector machine learner could produce a recall level of 61.9% [36]. Another work describes an online system for automatic detection and monitoring of cyberbullying cases from Askfm datasets. The approach depends on the detection of three necessary natural language components corresponding to Insults, Swears and Second Person [28]. Dinakar et al. (2012) conducted text classification experiments on YouTube data [37]. They adopted a bag-of-words supervised machine learning (SVM) classification approach to detect cyberbullying from the SM posts (i.e., intelligence, sexuality, race, and culture) and achieved an F1 score of 0.63. Furthermore, Reynolds et al. (2011) compared a rule-based model to a bag-of-words model for detecting cyberbullying posts and found that rule-based learning with several lexical features (e.g., the number of curse words in a post) outperformed the bag-of-words model [30]. Dadvar et al. (2014) combined the potential of machine learning algorithms with information from social studies for the automatic recognition of cyberbullying [38]. User information and expert views were used in addition to textual features, which resulted in a classification performance of F1 = 0.64. Nahar et al. (2014) applied a fuzzy SVM algorithm for cyberbullying detection [39]. They implemented some lexical features (e.g., the number of swear words and capitalized words), sentiment features, and features based on metadata (e.g., the user's age and gender) and report an F-score of 47%. In all of the studies mentioned above, cyberbullying detection was approached as a binary classification task (cyberbullying versus non-cyber bullying).

Furthermore, Hitesh Kumar Sharma, T.P. Singh, K Kshitiz, Harsimran Singh, and Prince Kukreja were involved for determining ways to identify bullying in the text by analyzing and experimenting with different methods for classifying bullying comments [25]. They have proposed an efficient algorithm to identify the bullying test and aggressive comments, and later they also wanted to analyze these comments for checking the validity. They used NLP and machine learning for analyzing the social comments and identified the aggressive effect of an individual or a group. They also tried to notice their audiences that the best performing classifier acts as the core component in a final prototype system that can detect cyberbullying on social media.

Nevertheless, some fundamental progress has been made in the domain over the past few years. However, most of the cyberbullying related works are based on baseline classifier and very few related to deep learning, and related to vast feature analysis [40].

2.2. Datasets Description

We have used two different types of datasets as original/base dataset: a dataset from the Askfm website, which we manually labeled, and a publicly available dataset related to cyberbullying, called Formspring dataset. These two base datasets would be further used in cyberbullying detection, categories detection, textual feature analysis, best feature selection, extended datasets creation, and negation datasets creation.

2.2.1. AskFm Datasets

The first original dataset that we have used in this thesis was collected from Askfm website[1]. Askfm is mainly used for asking questions publicly and getting answers from other Askfm users. Questions can also be asked anonymously. To collect each user's questions and answers, we have crawled each of the profiles using Python web crawler library, Beautiful Soup[2]. One possible way was to collect usernames by crawling Askfm website's home page. Askfm 'home' page provides 20 random users for each of the browser's sessions. However, it is not guaranteed that all our collected users would be English speakers. Therefore, we have only collected those usernames who were located in UK, USA, and CANADA. For changing locations, we have used a virtual private network (VPN). We have collected almost 3720 Askfm usernames to put forward the next step, collecting questions and answers from each of the user's profiles.

Questions and answers associated with each user's profile are saved in a CSV file. Question-answer pairs of a profile are only extracted if they contain cyberbullying swear words, further filtered by string matching technique. During the dataset collection, we crawled 3720 user-profiles and over 400,000 question-answer pairs. Applying swear words string matching technique reduced the data to 10k unique posts, containing at least one swear words either in question or in answer.

We have manually labeled the resulting 10k Askfm dataset. Labeling involves identifying whether each sentence contains cyberbullying or not. If the sentence includes cyberbullying, we have given the label '1' and '0', if otherwise, examples shown in Table 1. Once labeled, 21.3% of the dataset was identified as cyberbullying and rest was not cyberbullying.

After labeling the dataset, we have identified 2k number of questions and answers that contain cyberbullying. We have observed that user questions include 13% more cyberbullying compare to the answers of those posts, Figure 1.

[1] `https://ask.fm/` (accessed Oct 03, 2019)

[2] `https://www.crummy.com/software/BeautifulSoup/bs4/doc/` (accessed Oct 03, 2019)

Table 1. Posts labelling example for Askfm datasets

Posts	Posts label
how to tell if a guy is gay if they seem super straight.	0
You are gay	1

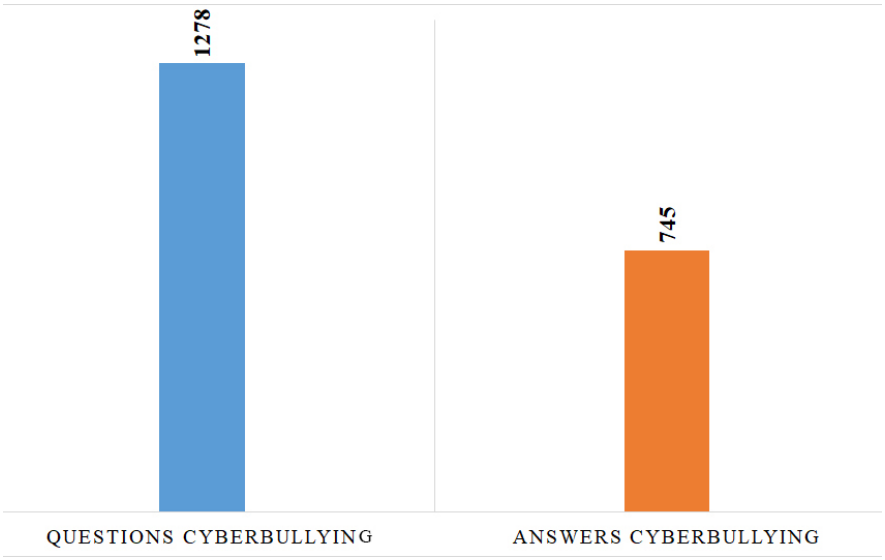

Figure 1. Numbers of cyberbullying contains in Questions and Answers of user posts

We have also identified eight different types of cyberbullyings from those sentences [7]. Among 10k dataset, 7.5k classified as 'none' cyberbullying data, and rest were categorized in different types as shown below, Figure 2.

1. Threat: expressions contain physical or psychological threats or indications of blackmail (ex. I will post your nude in Reddit).
2. Insult: expressions meant to hurt or offend the victim (ex. you are an ugly, useless little h*e!).
3. Curse: expressions of a wish that some form of misfortune will happen to the victim (ex. why not you go to hell).

Table 2. Categories labelling example for Askfm datasets

Posts	Posts label
I will post ur nude	T (Threat)
No he a bi*ch	I (Insult)
Why not you go to hell	C (Curse)
She slept with her ex behind his girlfriend's back	D (Defamation)
What was the last person you suck di*k?	S (Sexual)
Dont call her h*e	DE (Defend)
She's failed as f**k. punish her.	E (Encourage)
Haha sometimes after I got my butt toasted I ran to my room and looked at my bu*t in the mirror to see how red it was! xD did you ever do that	O (Others)

4. Defamation: expressions that tell secrete or defamatory information about the victim to a large audience (ex. she slept with her ex behind his girlfriend's back).
5. Sexual: expressions with a sexual meaning or intention. However, innocent sexual talk and sexual harassment consider different (ex. I wanna f**k you hard).
6. Defense: expressions in support of the victim by the victim himself or by a bystander (ex. don't call her h*e).
7. Encourage: expressions that contain inspiration of bullying for others. (She's failed as fuck. punish her.)
8. Other: expressions that contain any other form of cyberbullying related behavior than the ones described here.

Furthermore, we have labeled each type of the categories as example Table 2: Threat (T), Sexual (S), Curse (C), Insult (I), Defamation (D), Encourage (E), Defend (DE) Others (O). Figure 2 depicts the number of categories counted in the cyberbullying. Among all posts, sexual and Insult related cyberbullying were high in number (870) and (834). Defamation holds the middle position in quantity (334); however, encourage, curse, defend, threat, and others are almost the same in quantity, each less than a hundred in number.

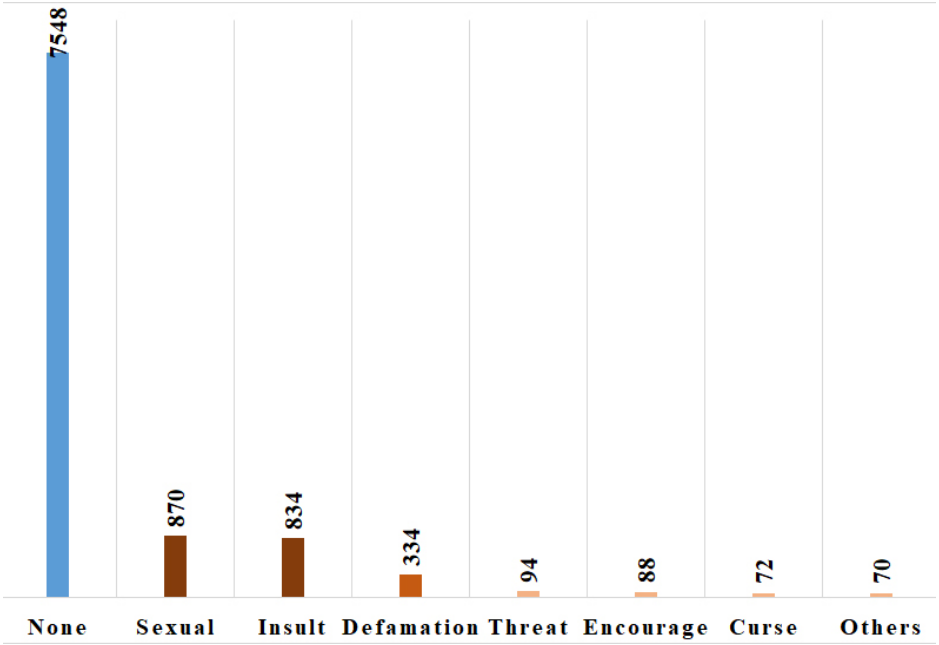

Figure 2. Types of cyberbullying among 10k of datasets

2.2.2. FormSpring Datasets

The second dataset that we have used as our base dataset is collected from Fromspring website [30], which is publicly accessible[3]. The data represented 50 IDs from

[3]https://www.kaggle.com/swetaagrawal/formspring-data-for-cyberbullying-detection

Formspring.me that were crawled in the Summer of 2010. For each user ID, the profile information and each post (question and answer) were extracted. Posts were uploaded into Amazon's Mechanical Turk and annotated by three workers for cyberbullying content. This dataset was labeled similarly the way we created ours, as in Table-1. The Formspring dataset contains 12k posts of which 17% contains cyberbullying.

2.3. Data Pre-Processing

2.3.1. Text Processing Technique

Most of these are standard techniques used in natural language processing (lowercase, reduction to words' root form, and removal of words and characters), with the addition of a couple that specifically address social media posts (emojis and hashtags treatment). The groups of preprocessing tested are briefly described in the paragraphs below, along with their potential value in detecting Cyberbullying in posts.

Lowercase

All characters were converted to lower-cases. This is helpful in reducing the dimension of the data, since capitalized words are interpreted equally to non capitalized words. In a model without characters converted to lower-cases, Hate, hate and HATE would be interpreted (eventually tokenized) as three different words. Despite its usefulness, bully is often correlated with the usage of capital words and characters [41].

Reduction to words' root

Reducing words to their root forms consists of removing their suffixes and reducing the words' expressions to their root forms. This is yet another feature usually beneficial to the reduction of the data's dimensionality, since words with similar meanings are converted to the same stem [42], e.g., the words affects, affection, affected and affecting are converted to the stem "affect" using the Porter's stemming algorithm [43]. Although this is useful in most cases, there are exceptions in which the meaning of the word might be altered. An example is the stemming of "plane" and "planned," which is "plane." Planned is the past tense of to "plan" and "plane" (in this case) refers to an "Airplane." Although both of the words share the same stem, their meaning is different. Aiming to reduce words to their root form, we used :

1. **Lemmatization:** the process is similar to stemming's, although it makes sure that the root form generated belongs to the language, using a dictionary. For that purpose, we used WordNet's [44].

Words and characters removal

Removing certain words and characters may also be beneficial in reducing the size of the data. Stop words that frequently occur in text data and typically convey no meaning to the message being passed, hence removing them is a technique quite

common in natural language processing. Common approaches tend to use pre-compiled dictionaries or other methods for their dynamic identification; however, the technique used to identify them may be crucial in truly separating noise from useful data [45]. In our experiments we used NLTK's stop words pre-compiled dictionary Steven Bird (2001a)[4]. We also considered these pre-processing features in our experiments to assess the removal of Emojie.

Hashtags

Hashtags are user-generated metadata that is group related messages with a specific topic. These are usually helpful in identifying the topic being addressed in that specific posts; hence their presence may be relevant in detecting cyberbullying. Aiming to test their influence in automatic cyberbullying detection, we recognized two different features. We removed them from the post and simplified them by removing their hashtags (e.g. #ugly was converted to ugly). To do so, we removed 46 Extractions and selection of textual features of the hash and divided the compound words into their constituents using Python's compound-word splitter library Kampik (2017)[5], e.g. #hateyou is converted to hate you. This approach poses some limitations, since splitting streams of texts into the corresponding words is an ambiguous task. For the same hashtag, there lie multiple splitting possibilities.

All datasets (AskFm and FromSpring) are preprocessed using standard NLP preprocessing tools. Preprocessing was done in the following manners:

1. Converted words to lowercases,
2. Filtered out stop-words,
3. Abbreviated words and short forms of social network slangs are replaced with original words, ex. fag to faggot,
4. Removal of unidentified characters, symbols ,
5. Tab token and multiple spacing has been removed,
6. Removal of single characters (except those characters could have abbreviated meaning ex. 'F' character),
7. Removal of URLs, and
8. Removal of hashtag (#) and user (user).

The results that we highlighted in Table-3 for pre-processing task, indicate the following:

- For both languages the use of uppercase to lowercase, abbreviated words and usernames in the preprocessing stage does not affect much the overall result.
- Stop-word and emoji removal works for all languages and increases the accuracy by 1%.
- A newline + Tab Token, and URL + Special Characters removal work well for all datasets and improved almost the performance accuracy by 2% each.
- When applying all preprocessings, both datasets showed improvement in classifier's accuracy almost by 2%. This indicates that language preprocessing

[4] `http://www.nltk.org/_modules/nltk/corpus.html`
[5] https://pypi.org/ project/compound-word-splitter/

Table 3. Changes of accuracy scores after preprocessing for both datasets using LR Model and TF-IDF word level feature

Pre-processing Type	**Askfm (%)**	**Fromspring (%)**
All	91	92
URL, Special Character	91	92
USERNAME (@user)	89	90
Lowercase	89	90
Stop word	90	91
Newline + Tab Token	91	92
Abbreviated word	89	90
Emoji	90	91
No Pre-processing	89	90

techniques for NLP impact the performance of offensive language detection for all languages and motivates to use them all together in the subsequent part of the experiment.

2.4. Feature Engineering

Aiming to deal with the lack of standardization in the extraction and selection of features for cyberbullying detection, we identified a set of features, and extracted and selected those that performed well in recognition of cyberbullying in Askfm and Fromspring comments (posts). We categorized the sets of features corresponding to their nature as follows:

1. **Sentiment:** features linked to the dataset's sentiment (e.g. posts' sentiment score).
2. **Semantic:** features include all that associated with the semantic of the corpora (e.g., the number of words per comment, average word length).
   (a) Punctuation: Considers punctuation related features (e.g. number of full stop marks).
   (b) Word: Features related to the words individually (e.g., average word length).
   (c) Character: Features related to the characters individually (e.g. number of capitals letters).
3. **Vectorization:** Features which vectorize the tokens and characters of the posts (N-Grams, Tf-IDF, Word Embedding).

2.4.1. Sentiment Analysis

On social network platforms, people express opinions on a variety of topics: love, hate, reviews, ratings, recommendations, and other forms of online expression. Often distinguishing the sentiment(s) behind these views usually turns out to be useful in separating understandings from the data. The critical idea behind sentiment analysis is to identify whether there is any positive and/or negative words or expressions, trusting firmly based on the straightforward meaning of words [46]. Furthermore, regular texts may contain negative words, hate synonyms, or the words (hate) itself, but the context associated may not be related to hate discourse. As an example, 'I hate to get up early in the morning! I hate my life!'. The word 'hate' has been used on this basic example, and the text itself maintains an overall negative meaning, although this is clearly not an example of cyberbullying. Therefore, the usage of sentiment analysis methods is questionable for cyberbullying detection in texts. To justify this claim, we have followed two procedures—overall sentiment score and word sentiment score.

Overall sentiment score

We computed three different features to extract the overall sentiment of posts, as follows:

1. **Sentiment score:** this feature is the single overall sentiment score of the posts, computed using TextBlob[6] Loria (2013).

2. **Sentiment subjectivity score:** the sentiment subjectivity score represents both the sentiment score of the posts and the subjectivity of the classification; this was also computed using TextBlob. The subjectivity introduces the concept of ambiguity when scoring the words' sentiment.

3. **Multiple sentiment score:** This feature was extracted using method VaderSentiment [47]. This provides a wide range of outputs that is the combination of positive, negative, neutral, and compound sentiment scores, providing a more elaborate approach.

Words sentiment score

Here we calculated the sentiment for each word which was computed by using TextBlob, unlike the sentence sentiment score and the results combined differently, generating a set of new features as bellow:

1. Positive words score: for each word with a sentiment score > 0.2 (the threshold goes from -1 to 1), the score of the positive word is incremented with the value.

2. Negative words score: similar to the score of the positive word, but with negative words, sentiment score < - 0.2.

3. Positive words count: number of words with sentiment scores > 0.2.

[6]https://textblob.readthedocs.io/en/dev/

4. Negative words count: number of words with sentiment scores < - 0.2.

5. Slang words score: sum of the sentiment scores for each slang word in the sentence.

6. Negative verbs count: number of verbs with negative sentiments < - 0.2.

The above mentioned, features are based on Linguistic Inquiry and Word Count (LIWC), that provide more than 90 features. In this work, we restricted the LIWC features to only the above categories. The LIWC program includes the main text analysis module, along with a group of built-in dictionaries. Once the processing module has read and accounted for all words in a given text, it determines the portion of total words that match each of the lexicon categories. For example, if LIWC received a document which contains 100 words and then compares them with its built-in dictionary, it might find that there are 15 negative emotion words. It would then convert this number to a percentage of 15% negative emotion.

The results displayed in Table-4 show that sentiment-based features barely had an impact (individually) on the classification task (15% less performance compared to the baseline results), where the multiple sentiment score yielded better results compared to other sentiment features. These results suggest it may not be useful to use sentiment analysis as a feature individually, especially for detecting cyberbullying posts since its gives significantly lower performance compared to the baseline results; however, sentiment feature may be combined with baseline feature, which we have experimented and the results have been explained in section 2.6.

Table 4. Results obtained on individual sentiment features tested against a baseline with no (sentiment) features. Baseline result was obtained by TF-IDF word level feature with Logistic Regression algorithm. Best result is given in bold.

Sentiment features	**Accuracy %**	**F1%**
Baseline (no sentiment)	**90**	**88**
Sentiment score	75.2	73.1
Sentiment subjectivity score	75.21	73.12
Multiple sentiment score	**76.3**	**74.2**
Positive words score	75.24	73.12
Negative words score	75.23	73.12
Positive words count	75.22	73.1
Negative words count	75.21	73.12
Slang words score	75.24	73.13
Negative verbs count	75.21	73.12

2.4.2. Semantic Analysis

Sentiment analysis considers only the sentiment conveyed by text; however, semantic analysis considers every aspect of the sentence. Although cleaning and preprocessing the data is mostly beneficial, it's inevitable to lose some valuable information during the process (e.g., URL's). It is common to see the use of punctuation or employment of capitalization associated with aggressive or even hateful discourse [46]. A practical example is the following Askfm post: 'You are a BITCH.'

The usage of the capitalized word in such posts aims to emphasize the insult to a person. A lower-casing word is a preprocessing feature that would ignore this subtle occurrence. This is a single example of how keeping track of punctuation, capitalized words, etc., might be useful in detecting cyberbullying online.

Punctuation marks

There are a lot of possible ways to conduct a semantic analysis. Most commonly, we extracted four different features: The overall number of punctuation marks, the number of exclamation marks, question marks, and full stops. We tested each feature individually, and results are displayed in Table-5, show that they bring no advantage in detecting cyberbullying for this dataset in particular. Thus, we considered the 'Number full stops' as the best semantic feature that yielded 64.3% of accuracy which is 9% better than other semantic features. These results suggest, it may not be usefull to use punctuation marks as features individually, especially for detecting cyberbullying posts since it yielded an average of 30% less performance compared to baseline results.

Table 5. Results obtained on individual semantic (punctuation) features tested against the base line with no (semantic) features. Baseline result obtained by TF-IDF word level feature with Logistic Regression algorithm. Best result is in bold.

Semantic Features	**Accuracy %**	**F1%**
Baseline (no sentiment)	**90**	**88**
Number of exclamation marks	57.2	53
Number of question marks	55.21	51
Number of full stops	**64.3**	**60.2**

Word features

Word-based semantic features may be relevant in classifying text, especially considering some subtleties are discarded upon cleaning and preprocessing the data. As mentioned before, lowercasing characters will automatically ignore any possible capital letters or words, hence acknowledging them may be relevant. We considered a word to be a set of characters, with a size larger than 1. Words may be 1 character long if that character is alphabetical. For each, we extracted a set of features as described below:

1. Number of all-capitalized words,
2. Ratio between all-capitalized words and total number of words,
3. Number of words, and
4. Average word length.

For each feature listed above, we conducted individual experiments using a term "frequency bag of words" and a Logistic Regression algorithm. Results, displayed in Table 3, show that acknowledging capitalized words and their relative frequencies in each posts has no positive impact on the overall performance of the model, however the identification of cyberbullying posts is worse when compared to the baseline (decreases by 33%). Thus, we considered the 'Number of words' as the best semantic feature that yielded 57.3% of accuracy (increased by 4-7%) compared to other semantic features.

Table 6. Results obtained with both individual and combined semantic (word) features tested against the baseline with no (semantic) features. Baseline result obtain by TF-IDF word-level feature with Logistic Regression algorithm. Best result is in bold.

Semantic features	**Accuracy %**	**F1%**
Baseline (no sentiment)	**90**	**88**
Number of all-capitalized words	50.2	45
Ratio between all-capitalized words and total number of words	52	49
Number of words	**57.3**	**54.2**
Average word length	53.24	50

Character features

Target characters individually instead of sets or words. For this, we extracted the number of capital letters, characters and special characters and computed the results for each feature individually as shown in the Table 7.

For each feature listed above, we conducted individual experiment using a TF-IDF word-level features with Logistic Regression algorithm. Results displayed in Table 7, show that Character features in each post has no positive impact on the overall performance of the model; however, the identification of cyberbullying posts is worse when compared to the baseline (decreased by 30%). However, among character features, 'Number of special characters' showed the best performance (4-7% better compared to other character features).

Table 7. Results obtained with both individual and combined semantic (character) features tested against the baseline with no (semantic) features. Baseline result obtained by TF-IDF word level feature with Logistic Regression algorithm. Best result is in bold.

Semantic Features	**Accuracy %**	**F1%**
Baseline (no sentiment)	**90**	**88**
The number of capital letters	57.2	53
Number of characters	55.21	51
Number of special characters	**64.3**	**60.2**
Number of punctuation marks	60.24	57.12

2.4.3. Count Vectors as Features

The notion of count vector is documentation of the dataset in which each line appointed to a record from the corpus and each column appointed to a term from the corpus, and each cell appointed to the identification of a specific term a specific document. The most straightforward count vector feature is that the vectorizer calculates the number of times a token appears in the document and uses this count as its weight, and this is what we are going to use in the thesis.

2.4.4. TF-IDF Vectors as Features

TF-IDF score speaks to the comparative significance of a term in the document and the whole corpus. TF-IDF score is made by two terms: the first calculate the normalized Term Frequency (TF), and the Inverse Document Frequency (IDF), figured as the logarithm of the quantity of the documents in the corpus ratio by the number of documents where the explicit term shows up. All experiments of this thesis have used three types of TF-IDF vector features, namely-Word level, N-Gram level and Character level.

Word level TF-IDF represents a score of every term in different documents, while N-grams are the combination of N terms together, and Character Level TF-IDF is the matrix representation of TF-IDF scores of character level N-grams in the corpus.

TF-IDF

The approach represents each post as a vector of terms and each term is represented in the vector by its TF-IDF value. Words that appear in the corpus but not in a given post will receive a zero- weight TF-IDF value. More specifically, the weight of the term i in post j is:

$$TFIDF_{i_j} = TF_{i_j}, IDF$$

with

$$TF_{i_j} = \frac{n_{i_j}}{\sum_k n_{k_j}}$$

Where n_{i_j} is the number of occurrences of term i in post j, and the denominator is the count of the occurrences of all terms in post j.

$$IDF_i = \log\left(\frac{|P|}{|P_j : t_i \in p_j|}\right)$$

Where $|P|$ stands for the total number of posts in the whole dataset, $|P_j : t_i \in p_j|$ is the number of posts in which the term t_i appears. Especially, TF presents a measure of how important a distinct term is in a given post (a local weighting). While, the IDF provides a scale of how significant a particular term is within the entire corpus (a global weighting). IDF scores are higher for terms which are good discriminators between posts (i.e., terms appearing in many posts will receive lower IDF score).

N-gram features

Traditional n-grams are sequences of n elements (where n is often less than five) as they appear in texts. These elements can be words, characters (number, letter, etc.), POS tags, or any other elements as they appear one after another in texts. Independence assumption is made such that each word depends only on the last n-1 words. Typically, the set of n-grams generated by moving a window of n words along the document under consideration, one word at each time. Then, the number of occurrences of each n-gram is counted. A key advantage of such a feature is that we do not need to perform advanced segmentation, neither to adopt a dictionary or language-specific technique, but, on the other hand, for large corpus, the number of n-grams becomes extremely huge, and many will have no discrimination power.

The tokens bagged by a bag of words depend on the n-grams used (contiguous sequence of tokens). Word n-grams consider words, or sets of words (when n is higher than 1), as tokens, while character n-grams consider characters or sets of characters (when n is higher than 1). Unlike (word-level) TF-IDF, n-gram features allow us to account for ordering among the tokens. Several combinations of TF-IDF, N-gram features have been tested where n ranges from a lower bound and upper bound. Our experiment [2,3] and [3,4]-grams are found to be the features that improved the most the detection. We have used three different combinations of TF-IDF: word-level, N-Gram word-level (for N=2, 3), and N-Gram Character level (for N=3, 4). Word level TF-IDF feature assigns a score to every term in documents, while word-level N-gram feature applies TF-IDF scoring to all 2-grams and 3-grams tokens extracted from the whole corpus dataset. Character Level TF-IDF provides a matrix representation of TF-IDF scores of character-level n-grams in the corpus. We restricted to 5000 features for each type to avoid the computational cost.

2.4.5. Word Embedding as Features

A word embedding is a procedure of demonstrating words and documents with a dense vector representation (example Figure 3). The place of a word inside the vector space

learned from text and constructed the word surrounded by similar categories of words. We can train word embedding using the input corpus itself; however, for our work we have used pre-trained word-embedding, namely FastText[7].

There are other pre-trained word-embeddings like Glove, Word2Ve etc, these are open source. However, FasText has the ability to produce embeddings for missing words. It's able to do this by learning vectors for character n-grams within the word and summing those vectors to produce the final vector or embedding for the word itself. By recognizing words as a sum of parts, it can predict representations (scores) for new words by simply adding the vectors for the character n-grams it knows about in the new word.

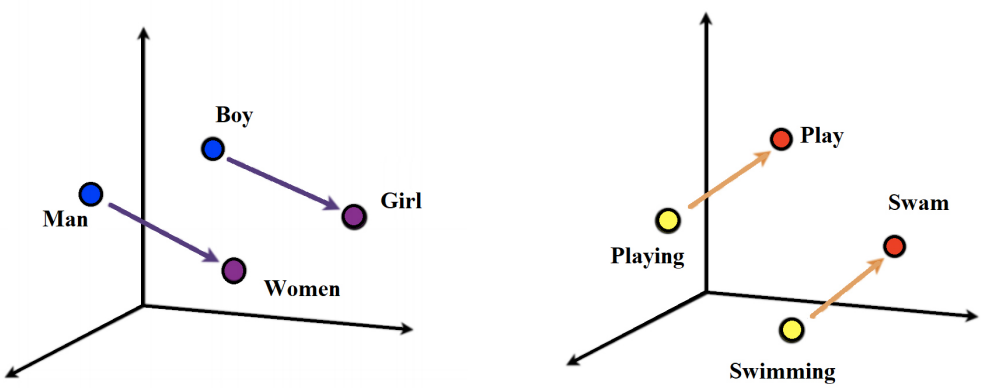

Figure 3. Word vector representation example, male-female and verb-tense

2.4.6. Concatenation of Features

One of the goals of this thesis is to improve the classifier result. One possible approach could be concatenated different features and apply them in classifier. In this part, we have concatenated Count and LIWC, Count and TFIDF, Count and TFIDF, Count and Ngram, TFIDF and LIWC, TFIDF and LIWC, Ngram and Ngram Char, TFIDF and Ngram Char, and sentiment features. This concatenation carried out for Linear classifier, Naive Bayes, SVM and RF. Among all of the concatenation, CharLevel + WordLevel + Sentiment yields the best accuracy, the result has been added to result section 2.6.2.

2.5. Classification Architecture

Initially, we employed a random split of the original dataset into 70% for training and 30% for testing and validation ensuring the same proportion of dataset for all kind of datasets (both original and artificially generated datasets) in order to ensure a balanced training.

[7] `https://s3-us-west-1.amazonaws.com/fasttext-vectors/wiki-news-300d-1M.vec.zip` (accessed Dec 30, 2018)

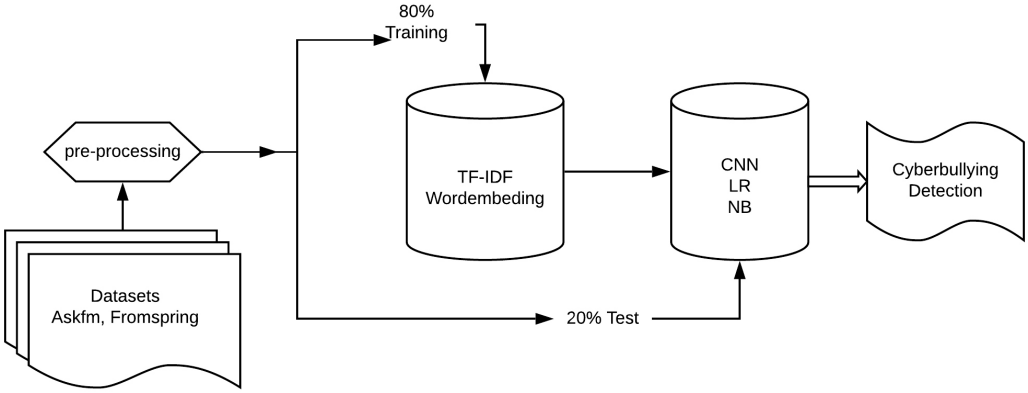

Figure 4. A general synoptic of the system.

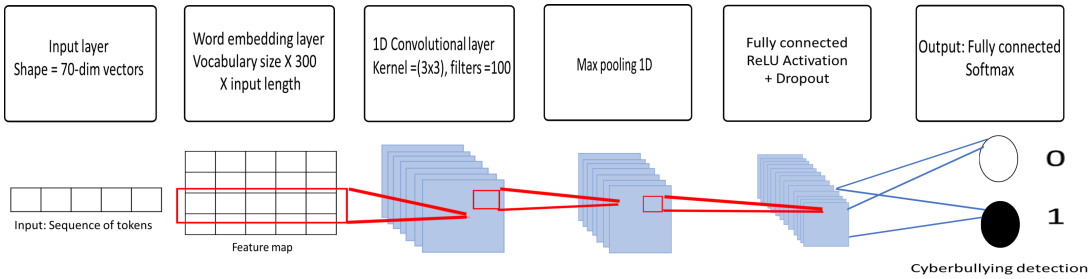

Figure 5. The architecture of our proposed cyberbullying detection CNN

Five types of classifiers implemented: Convolution Neural Network (CNN), the recurrent neural network LSTM models, Linear regression, Naive Bayes, and Random Forest. We adopted (Kim, 2014) [48] CNN, architecture, where the input layer was represented as a concatenation of the words forming the post (up to 70 words), except this case, each word was represented by its FastText embedding representation with 300 embedding vector (Figure 5). A convolution 1D operation with kernel size three has bean used with a max-over-time pooling operation over the feature map with a layer dense 50. Dropout on the penultimate layer with a constraint on l2-norms of the weight vector was used for regularization. Similarly, the LSTM scheme is similar to the word embedding representation, as in our CNN model. However, we have followed, where LSTM layers have 128 units, followed by a dropout of 10%. On the other hand, two baselines algorithms that use Linear Classifier (Logistic Regression) and Naive Bayes Classifiers considered for comparison purposes [49]. The details of the implementation reported in our GitHub page of this project with datasets and codes[8]. The various features were examined by each classifier to test its accuracy and robustness.

[8] `https://github.com/saroarjahan/cyberbullying` (accessed June 03, 2020)

2.5.1. Linear Classifier

In the field of ML, the goal of statistical analysis is to use an object's features to recognize which class (or group) it relates to. A linear classifier attains this by making a classification determination based on the value of a linear combination of the features. An object's features are also known as feature values and typically presented to the machine in a vector called a feature vector. Such classifiers show better performance for practical problems such as text classification, and more generally, problems with multiple variables (features). Moreover, reaching efficiency levels comparable to non-linear classifiers while requiring less time to train when using Linear Classifier (Logistic Regression). Here, Logistic measures the relationship between the categorical dependent and independent variables by measuring probabilities using a logistic/sigmoid function.

2.5.2. Naive Bayes

In machine learning, Naive Bayes (NB) classifiers are a combination of simple "probabilistic classifiers" based on employing Bayes' theorem with strong (naive) with an assumption of independence among predictors. A Naive Bayes classifier undertakes that the existence of a specific feature in a class is distinct to the existence of any other feature. For example, an egg may be considered a chicken egg if it is round, white, and about two (2) inches in diameter. Even if these characteristics depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability that this egg is a chicken egg, and that is why it is known as 'Naive.'

2.5.3. Support Vector Machine

Another important model training we have used in our project, namely the Support Vector Machine (SVM). We have used it because it's mainly a controlled machine learning procedure which can be used for together taxonomy and regression challenges. The model extracts the finest probable hyper-plane / line that isolates the two classes. In short, SVM specified a set of training samples, each manifest as referring to one or the other of two classes, an SVM training algorithm assembles a model that allocates new samples to one class or the other, making it a non-probabilistic binary linear classifier. Moreover, the SVM model is a demonstration of the samples as points in space, plotted so that the samples of the distinct categories are divided by a clear gap that is as inclusive as possible. New samples are then plotted into that same space and predicted to belong to a category constructed on which side of the gap they belong.

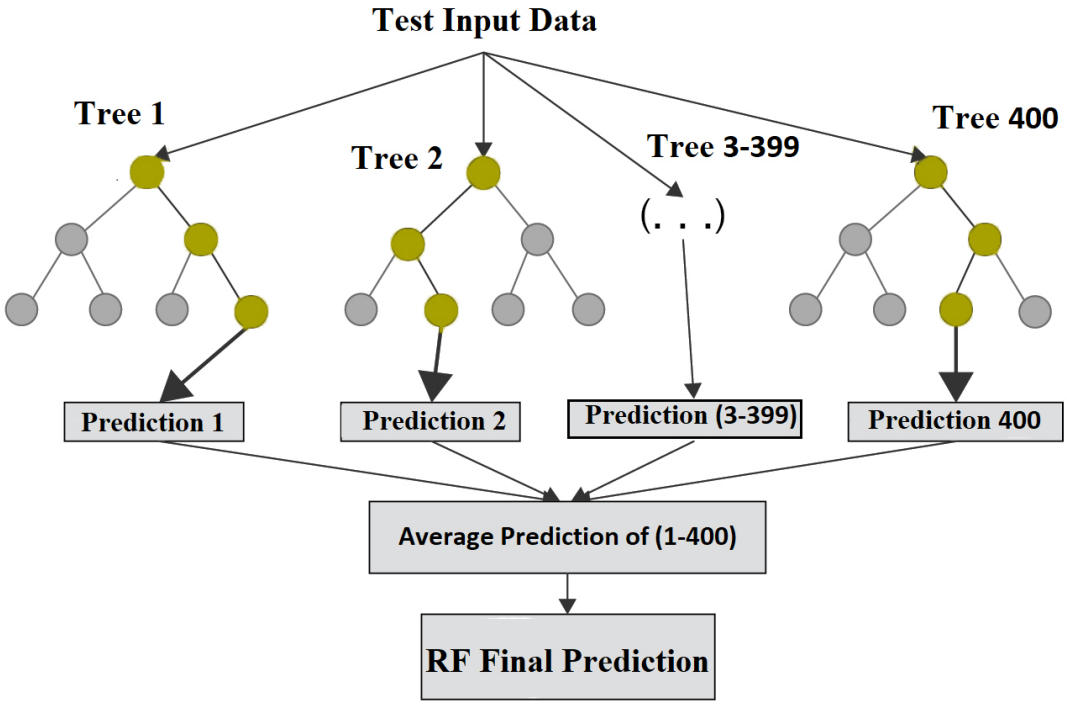

Figure 6. The architecture of Random Forest model

2.5.4. RF

Random Decision or Random forests (RF) are an ensemble learning technique for classification, regression, and other tasks that operate by creating a gathering of decision trees during training and resulting popular prediction of the individual trees, for example, Figure-6. Random decision forests use overfitting to their training set. The term "random forests" originally comes from wherein 1995 Ho et al [50]; proposed a model that aggregates a set of decision trees built upon a random subset. However, this approach encounters an issue in prediction due to over-fitting (also know as over-learning) after that bagging technique introduced, which tends to resolve the problem of over-learning. Bagging develops the estimate or prediction itself by calculating the mean of this prediction over a collection of bootstrap (random training set) samples. By joining ideas of the ensemble methods and decision trees, Breiman in 2001, gave use to decision trees, that is, sets of randomly trained decision trees [51]. Random Forest improves the bagging technique by reducing the relationship between the sampled trees by various sources of randomness.

Another benefit of using RF classifier is to find out the best features. In our work, we wanted to know which features are most important for cyberbullying detection. RF model finds the best features by calculating which features are getting used repeatedly in different decision trees for predicting final results.

2.5.5. CNN

A convolutional neural network (CNN) is one of the most popular and widely used types of deep neural networks. CNN model has a successful application in image and video identification, recommender systems, face recognition classification, medical image analysis, text classification, and financial time series. However, recently, the CNN architecture has proven popular over baseline classifier in several NLP related projects due to its high accuracy. In this thesis would like to verify this claim as well.

CNN's considered the regularized variants of multilayer perceptrons. Multilayer perceptrons usually mean wholly joined networks; that is, each neuron in one layer is connected to all neurons in the next layer. The "fully-connectedness" of these networks offers them prone to overfitting data. Conventional ways of regularization involve adding some form of magnitude measure of weights to the loss function. CNNs take a distinct path towards regularization: they benefit from the hierarchical pattern in data and assemble more complicated patterns using smaller and simpler patterns. Therefore, on the order of connectedness and difficulty, CNN is on the lower edge.

2.5.6. Long Short-Term Memory

Similar to artificial Recurrent Neural Network (RNN), Long short-term memory (LSTM) is a structure applied in the domain of deep learning. However, unlike conventional feed-forward neural networks, LSTM has feedback relationships. It can prepare not only single data features (such as images), instead of an entire series of data text. For example, LSTM applies to tasks such as unsegmented, connected handwriting verification, and speech recognition. A standard LSTM unit composed of a cell, an input gate, an output gate, and a forget gate. The cell learns values over arbitrary time intervals, and the gates control the flow of information into and out of the cell. LSTM networks are well-suited to classifying and processing.

2.6. Results

In this result section, we will discuss four types of results: cyberbullying detection, categories detection, concatenation of features, and finding the best features. Cyberbullying detection will provide ideas about which classifier and features worked best to detect cyberbullying. Other-hand, categories detection may provide a sense of whether it is possible to identify categories of different types of cyberbullying. We have also run the test for concatenation of various features since the Sentiment feature, and CharLevelTDIDF performed better, therefore, we have only shown these two features in the result section.

2.6.1. Cyberbullying Detection Results

The results highlighted in Table 8 indicates the following:

Table 8. Classifier Accuracy & F1 scores (%) by using Askfm , Fromspring, and both datasets (best is in Bold).

Classifier	Askfm		Fromspring		Both Dataset	
Feature Name	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**
NB + Count Vector	84	80	87	85	87	84
NB + WordLevel TF-IDF	88	82	89	88	89	87
NB + N-Gram TF-IDF	87	82	86	85	87	86
NB + CharLevel TF-IDF	88	83	89	88	89	87
LR + Count Vector	89	87	91	90	91	89
LR + WordLevel TF-IDF	90	88	91	91	91	90
LR + N-Gram TF-IDF	88	83	90	89	90	88
LR + Char Level TF-IDF	**91**	**89**	**92**	**91**	**91.5**	**90**
SVM + Count Vector	86	82	89	88	89	87
SVM + WordLevel TF-IDF	87	83	88	87	88	87
SVM + N-Gram TF-IDF	86	83	89	88	89	88
SVM + CharLevel TF-IDF	87	84	90	88	90	89
RF + Count Vector	88	86	91	87	91	86
RF + WordLevel TF-IDF	88	87	91	87	91	89
RF + N-Gram TF-IDF	88	85	87	84	92	90
RF + CharLevel TF-IDF	89	87	91	88	91	90
CNN + WordEmbedding	**91**	**89**	**92**	**92**	**92**	**90**
LSTM + WordEmbeddings	90	88	91	90	91.5	89

- Among all six types of classifiers, CNN works best for both Askfm and Formspring datasets, which indicates that neural network models work better than baseline classifiers. However, LR with Char Level TF-IDF shows similar performance as CNN 91% accuracy for Askfm and 92% for Formspring datasets. These results indicate that in NLP based cyberbullying detection CNN model is always preferable with word-embedding features. Since we have used pre-trained word-embedding and since it has provided the best accuracy and F1 scores, we assume that pre-trained word-embedding could be a reliable choice in this case.
- Among baseline classifiers, Linear Regression models outperformed all baseline classifiers, including NB, RF, SVM, in terms of accuracy and F1 scores. In all cases, NB and SVM performed lower than others; other hand RF classifiers

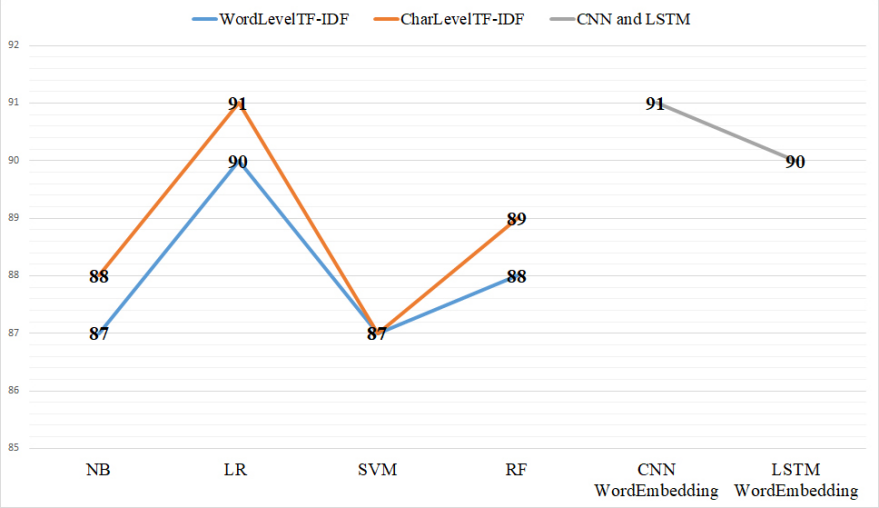

Figure 7. Cyberbullying detection comparison among different classifiers (NB, LR, SVM, RF, CNN, and LSTM) using wordLevelTF-IDF, charLevelTF-IDF, and word-embedding features for Askfm dataset. CNN+word-embedding and LR+charLevelTF-IDF outperformed others.

Table 9. Classifier Accuracy & F1 scores (%) after concatenation of Sentiment feature for Askfm , Fromspring, and both datasets (best in bold).

Classifier	Askfm		Fromspring		Both Dataset	
Feature Name	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**
NB + Multiple sentiment score	74	73	75	74	76	73
NB + CharLevel TF-IDF + Sentiment	88.5	83.4	89.5	88.3	89.3	87.3
LR + Sentiment	77	75	78	76	77	74.6
LR+Char LevelTF-IDF+Sentiment	**91.7**	**89.6**	**92.9**	**91.8**	**92.7**	**90.7**
SVM + Sentiment	74	73	76	74	75	73
SVM + CharLevel TF-IDF+Sentiment	88.4	84.3	90.4	88.9	90.4	89.6
RF + Sentiment	76	74	77	75	76	74
RF + CharLevel TF-IDF + Sentiment	89.6	87.4	91.5	88.4	91.5	90.4

performed 1% better than NB and SVM (Figure-7 shows the performance Comparision among different classifier for Askfm dataset).

- Among all vectorize features, "n-gram characterLevelTF-IDF" outperformed wordLevelTF-IDF, Count Vector and n-gram wordLevelTF-IDF. Though "n-gram wordLevelTF-IDF" performed much better, and in some cases it has performed similar to "n-gram character LevelTF-IDF," therefore, we have used these two features mostly throughout the project. Since, "Count Vector" has performed lowest in terms of accuracy and F1 scores in all cases, we don't find prominent to use these features in our further experiments.

- Finally, this cyberbullying detection experiment carried out with Askfm and Formspring datasets, and we can see both datasets yield almost similar accuracies with only 1% deviation. This comparison shows strong support for the reliability of our datasets ground truth and results.

2.6.2. Concatenation of Features

In the previous section, 2.6.1 (Cyberbullying Detection Results), we have observed among all vectorize features, 'n-gram characterLevelTF-IDF' and 'n-gram wordLevelTF-IDF' performed better than others. Here these two features used as a concatenation, and in section 2.4 (Feature Engineering), only the 'Sentiment' feature showed decent results; therefore, we have concatenated sentiment feature as well.

The results are highlighted in the Table 9 & 10 indicates the concatenation of sentiment feature and multiple features (CharLevel + WordLevel+ Sentiment) as following:

- In Table 9, among all classifier models with 'Semantic' feature, LR classifier outperformed all other classifiers with 77% accuracy. NB was the lowest performed classifier in terms of accuracy, yielded 74% accuracy.
- In all cases, concatenation of sentiment feature provides the best result and improved by .9% in terms of accuracy and F1-measure compared to individual feature (with out concatenation).
- In Table 10, when three features (C.Level + W.Level + Sentiment) concatenated all together, it yielded .6% better accuracy than the concatenation of only two features. This seems true for both Askfm and Fromspring datasets, which inspires the use of concatenation of multiple features for cyberbullying detection and similar NLP tasks.

Table 10. Classifier Accuracy & F1 scores by using concatenation of features (CharLevel + WordLevel+ Sentiment) for Askfm, Fromspring, and both datasets (best in Bold).

Classifier	Askfm		Fromspring		Both Dataset	
Feature Name	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**
NB + C.Level + W.Level+ Sentiment	89	83.5	90	88.5	90	87.6
LR + C.Level + W.Level+ Sentiment	**92.2**	**90**	**94**	**92**	**93.3**	**91**
SVM + C.Level + W.Level+ Sentiment	88	84.3	91	88.6	90.8	89.7
RF + C.Level + W.Level+ Sentiment	89.8	87.6	91.9	88.7	91.7	90.7

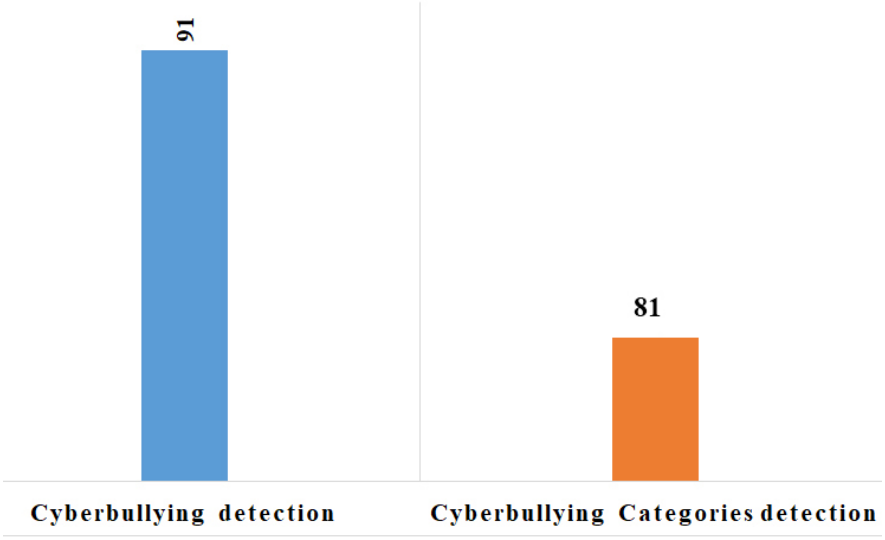

Figure 8. Classifier accuracy (%) for cyberbullying detection Vs. cyberbullying categories detection (CNN architecture with Askfm datasets).

2.6.3. Cyberbullying Categories Detection Results

For cyberbullying category detection, we have applied labeling based on Table 2 and used the same classifier architecture that has been used for identifying cyberbullying. Classifier Accuracy & F1 scores for cyberbullying categories detection by using Askfm as follow:

- Among all four types of classifiers, CNN works best for cyberbullying category detection, similarly to bullying detection, which indicates that neural network models work better than baseline classifiers. However, LR with charLevel TF-IDF shows similar performance as CNN 81% accuracy and 78% F1 scores.
- For category detection similar performance was observed among baseline classifiers (NB Vs. LR), Linear Regression models outperformed 6% compared to Naive Bayes model in terms of accuracy and F1 scores.
- Among the TF-IDF features, 'n-gram characterLevel TF-IDF' out performed 1-2% compared to 'wordLevel TF-IDF' and 'n-gram wordLevel TF-IDF.'
- Classifier results compared to cyberbullying detection (Table-8) Vs cyberbullying categories detection (Table-11) shows categories detection has 10% less accuracy compared to cyberbullying post detection. These results indicate that cyberbullying category detection is tough due to the fact of training the classifier with multiple labeling at a time. One of the reasons could be, when we were labeling categories, the dataset was distributed to eight categories, making fewer posts labeled for each category. This lower number of data might encounter difficulties in training the models compared to only bullying detection.

Individual cyberbullying categories detection shown in Table 2, where we have used LR + Char Level TF-IDF Classifier. Performances for individual categories discuss as follow:

Table 11. Classifier Accuracy & F1 scores for cyberbullying Categories detection by using Askfm (best in Bold).

Classifier and Feature	**Accuracy**	**F1 score**
NB + WordLevel TF-IDF	75	72
NB + CharLevel TF-IDF	76.5	72.4
LR + WordLevel TF-IDF	80.4	78
LR + Char Level TF-IDF	**81**	**78**
SVM + WordLevel TF-IDF	76	72
SVM + CharLevel TF-IDF	76.5	72
RF + WordLevel TF-IDF	78	76
RF + CharLevel TF-IDF	78.3	76.6
CNN + WordEmbedding	**81.2**	**78.3**
LSTM + WordEmbeddings	81	78

- Table 12 shows that all eight different type categories yield quite close performance in terms of accuracy and F1 scores; however, Defamation, Threat, and Curse performances were best compare to rest of the categories have yielded average 82% accuracy and 79% F1 score, where a maximum to minimum range of accuracy and F1 scores were 74% - 82.5%, and 72% - 80% respectively. One possible explanation could be rooted back the inherent difficulty to distinguish the scope of vocabulary employed by these categories, so that a random genuine annotator would equally attribute a given utterance to either Threat, Defamation, or Curse, which renders any classification based approach yield almost similar results. On the other hand, Sexual and Insult categories performed close to each other (81% accuracy and 78% F1 score) which is 1% lower than maximum performance, because many Sexual and Insult posts share the same contextual meaning and there are not many distinguishable instances in training dataset either, which makes the model harder to train to distinguish these classes. For example, the utterance, "I wanna f**k you bit*h" could be an Insult as well as a Sexual bullying. In Table 13 showed that 100 posts are common between Sexual and Insult categories.

  Similarly, the Encourage and Defend category performed 79% in terms of accuracy, which is 3% lower than the best performance. For example, the posts "Don't call her bit*h" and "Call her bit*h," extracted from Encourage and Defend categories, differ only through the statement "don't." Therefore, possibly these small differences were hard for the classifier to train the model and, accordingly, yielded relatively lower performance value compared to others.

- Finally, we have observed the "Other" category that represents those posts that do not belong to any of the seven categories. Here, it has yielded 74% accuracy, which is the 8% lower performance compared to the best performance (82%

accuracy). One possible explanation could be that the "Others" category contains a minimal number of posts (only 72 in number), and at the same time, it does not follow any pattern of posts. Simply those posts have not fulfilled any categories to place in, we have put those in "Other" categories section; therefore, there exist amalgamation of different types of posts, which were hard to classify.

- Based on the above analysis of categories detection, we may say that categories detection varies performance regarding the resemblance of one category posts to others other categories. For example, in Table 13, Insult and Sexual categories have the highest number of posts, logically it should perform better; however, it shows less performance due to having a 5.5% common post between them.

Table 12. Classifier Accuracy & F1 scores for cyberbullying indiviudal Categories detection by using Askfm dataset (best in Bold). LR classifier with Char Level TF-IDF feature been used.

Categories Name	**Accuracy**	**F1 score**
Sexual	80	78.4
Insult	80.2	78.7
Defamation	82	80
Threat	82.2	79.2
Encourage	79	77.4
Curse	82.3	79
Defend	79.3	77
Others	74	72

2.6.4. Selection of Best Feature

Features are important when we are training a machine learning model: by getting a better understanding of the model's logic, and improving it by filtering unnecessary features or focusing on the important variables. Furthermore, it can be useful to eliminate variables which are not that vital and have comparably better performance in much shorter training time. To find the best features, we have used RF classifier. The results are highlighted in the Table-14 indicate the following:

- For "wordLevel TF-IDF" in both datasets, we have found "You," and "F**k" has the top score approximately .30. However, among the top 10 features in WordLevelTF-IDF, we have found 50% feature common in both datasets. For example, "You" word feature exists in both datasets, indicating some words highly important for cyberbullying.

Table 13. Number and percentage of common posts between two different categories.

Categories	Sexual	Insult	Defamation	Threat	Encourage	Curse	Defend	Others
Sexual (870)	-	100 5.5%	21 1.7%	3 .32%	3 .3%	5 .5%	2 .21%	3 .3%
Insult (834)	100 5.5%	-	15 1.3%	4 .4%	5 .6%	2 .5%	4 .44%	4 .44%
Defamation (334)	21 1.7%	15 1.3%	-	2 .32%	2 .3%	0	0	1 .2%
Threat (94)	3 3.2%	4 .4%	2 .32%	-	0	0	0	0
Encourage (88)	3 .3%	5 .6%	2 .3%	0	-	0	0	0
Curse (72)	5 .5%	2 .5%	0	0	0	-	0	0
Defend (71)	2 .21%	4 .4%	0	0	0	0	-	0
Others (70)	3 .3%	4 .44%	1 .2%	0	0	0	0	-

- Among the top 10 features in "n-gram wordLevel TF-IDF", we have found 40% feature common in both datasets. As example, "are you gay" word feature exists in both datasets.
- Furthermore, top 10 feature in "n-gram charLevelTF-IDF" we have found 30% feature common in both datasets. As example, "yo" character feature exists in both datasets.
- Among all three TF-IDF "You" and "F**k" word features exist as a word level, N-gram and Charlevel, which indicate "you" and "F**k" words play a great role in cyberbullying cases.
- Furthermore, we have observed some less important stop words (a, are, to, etc.) are being repeated in N-grams TF-IDF. Removal of these less important features may enrich overall feature engineering and enhance performance in much shorter training time.

Table 14. Top 10 features name and score for Askfm and Formspring datasets

Classifier Feature	**Askfm features**	**Fromspring features**	**Common features**
RF + WordLevel TF-IDF	you , 0.031 f**k , 0.030 s*x , 0.026 di*k , 0.022 bit*h , 0.021 pu**y , 0.015 a*s , 0.014 h*e , 0.010 bu*t , 0.010 nigga , 0.010	f**k', 0.039 you', 0.031 bit*h', 0.022 your', 0.020 gay', 0.019 pu**y', 0.017 di*k', 0.015 fake', 0.015 a*s', 0.013 suck', 0.012	you, 0.031, 0.031 f**k, 0.030, 0.039 di*k, 0.022, 0.015 bit*h, 0.021, 0.022 pu**y, 0.015, 0.017 a*s, 0.014 , 0.013
RF + N-Gram TF-IDF	suck my di*k', 0.015 you are a', 0.014 had s*x with', 0.009 want to f**k', 0.009 f**k with you', 0.008 s*x with tayy', 0.007 you have a', 0.007 your b*tt is', 0.007 are you gay', 0.006 i want to', 0.006 f**k your mom', 0.006	I hate you , 0.019 are you gay , 0.015 you are a , 0.014 why are you , 0.013 are you a , 0.012 stop trying to , 0.010 are you a , 0.0104 f**k your mom , 0.010 want to f**k , 0.009 do you like , 0.009	you are a', 0.014 you are a , 0.014 want to f**k', 0.009 want to f**k , 0.009 are you gay', 0.006 are you gay , 0.015 f**k your mom', 0.006 f**k your mom , 0.010
RF + CharLevel TF-IDF	fuc , 0.016 ou , 0.012 yo , 0.012 sy , 0.011 ck , 0.010 bi , 0.010 yo , 0.010 you , 0.008 s*x , 0.008 ick , 0.007	fu', 0.023 ck', 0.019 yo', 0.014 uck', 0.013 tch', 0.011 yo', 0.0102 ck ', 0.010 ga', 0.009 u ', 0.008 y', 0.008	yo , 0.012 yo', 0.014 ck , 0.010 ck', 0.019 yo , 0.010 yo', 0.0102

3. EXPANDED DATASETS EFFECT ON CYBERBULLYING

3.1. Overview

Automatic identification of cyberbullying or abusive language from the textual content is known to be a challenging task. The challenges arise from the inherent structure of offensive speech and the lack of labeled large-scale corpus, which enables efficient machine learning-based tools, including neural networks. This part of the thesis advocates a new data augmentation-based approach that would enhance machine learning tools in detecting cyberbullying in social media texts. Unlike standard under sampling approach in the literature of handling imbalanced classes, the developed approach uses both wordsense disambiguation and synonymy relation in WordNet lexical database to generate coherent equivalent utterances of hate-speech input data. The disambiguation and semantic expansions are intended to overcome several limitations of the social media posts and comments, such as their unstructured nature as well as the limited semantic content after prepossessing. Besides, to test the feasibility of the proposal, a novel protocol has been employed to collect cyberbullying traces data from ask.fm forum where about 10K size dataset has manually been labeled (same datasets used as previous section). Next, the problem of cyberbullying identification is viewed as a binary classification problem using an elaborated data augmentation procedure and an appropriate classifier. For the latter, a new CNN architecture has been put forward whose results for cyberbullying detection were compared against a set of some most widely used classifiers constituted of Naive Bayes and Linear Regression classifiers with and without data augmentation. The outcome of the research was promising which yielded almost 99% of classifier accuracy, an improvement of more than 10% with respect to the baseline results.

In this part of the thesis, we again used our cyberbullying Askfm, and another publicly available cyberbullying dataset to compare the accuracy of the final result. The descriptions of datasets have been mentioned in "Datasests Description" in section 2.2. After that, we carried out standard processing tasks on the dataset and to found a feasible way to augment our initial datasets and examine whether augmented datasets are good enough to use and if capable of detecting cyberbullying. For the augmentation process of initial datasets, WordNet sense disambiguation technique and Lesk-algorithm have been used [52]. Initially, we had two base datasets; therefore, this proposed technique generated two additional extended datasets, which have been again separately used for cyberbullying detection and compared with the initial results that we had obtained by using the initial dataset. In the result section, we have shown whether an extended dataset was suitable for cyberbullying detection or not.

3.1.1. Related Work

Cyberbullying is a widely covered topic where a fair amount of researches have focused on the definition and prevalence of the phenomenon [53], the identification of different forms of cyberbullyings [54], among others. In this respect, starting from the pioneering work of the use of machine learning-based classifiers for detecting abusive languages became popular within the information processing research community

[55], combined pre-defined language elements [25] and word embedding to train a regression model [56] proposed a word-embedding based representation. Nevertheless, the limitations of machine learning-based approaches have also been widely reported by several scholars due to:

1. the challenges associated with the definition of hate speech discourse, where the presence of a wording insult, for instance, does not necessarily entail a hate speech post, and
2. the limited scope of training samples questions the effectiveness of any machine learning-based approach due to the constant evolving of hate-speech corpus and the variety of expressions therein.

Besides, [26] reported that many of the existing hate-speech detection approaches are largely biased towards detecting content that is non-hate as opposed to detecting and discriminating real hateful contents, possibly, because the non-hate contents may not contain any discriminating features. However, some significant works have been done for automated cyberbullying detection by using social network datasets such as Ask.fm datasets [28]. [37] conducted hate-speech text classification experiments on YouTube data while an annotated cyberbullying dataset and a fine-grained classification are put forward by [54].

On the other hand, several works have been reported in the context of word-sense disambiguation using the so-called Lesk algorithm or extended lesk algorithm with wordnet [1, 57]. Typically, word-sense disambiguation is the process of automatically identifying the meaning of a given target word in its associated context. It has drawn much interest in the last decade, and much improved results are being obtained. It has been reported that the Extended-WordNet based word-sense disambiguation for noun, verb, and adjective categories achieved a precision of 85.9% [57]. This motivates our core idea of using word-sense disambiguation to cyberbully identification tasks, which, as far as we know, has not been performed previously to generate datasets artificially. Also, it provides useful insights to handle imbalanced class, created by small instances of labeled cyberbullying cases as compared to non-cyber bullying cases.

3.2. Methodology

The overall methodology includes a four-stage process: data collection, data augmentation, feature engineering and classification, and finally, testing and validation. Since the lack of availability of a well-balanced annotated cyberbullying dataset [54], methodology includes collecting our own datasets, datasets description described in section 2.2 (Datasets Description). In the second phase, a data augmentation has been performed using the wordnet-based sense disambiguation technique and Lesk-algorithm [52], PoS tagging and Synonyms replacement. The classification and feature engineering use the same as previously described in section 2.5 (Classification Architecture) and 2.4 (Feature Engineering). The final result achieved by the augmentation process has been duly evaluated with initial base results for both Askfm and Formspring dataset.

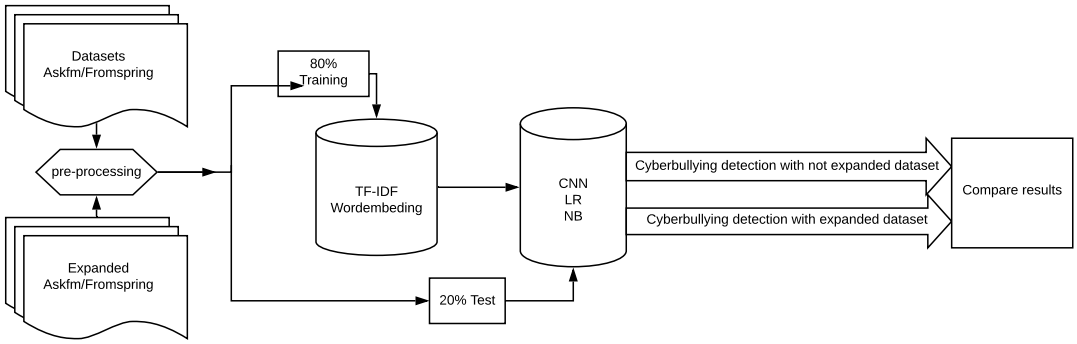

Figure 9. A general synoptic of systems for not expanded and expanded datasets

3.2.1. Expanded Datasets Creation

The artificial dataset is the enriched version of the base dataset. The aim is to expand the semantic space of each sentence of the original cyberbullying case in the datasets. For this purpose, we have proposed three possible methods briefly summarized below and detailed in the pseudo-code "Algorithm 1".

Method 1: We applied word-sense disambiguation to each word of input sentence, after the preprocessing stage that removes stopwords and other uncommon characters, so that the synonymy relation was used to extract the list of senses for each word. Next, to find out which of these senses better fit the context of the sentence, Lesk's algorithm was employed [52]. For instance, Fig 10 demonstrates this method's application to the sentence: "He is gay" and its newly generated sentences.

Method 2: We apply a Part of Speech (PoS) Tagging to each sentence. This method will then allow us to extract all meanings (synsets) and synonyms that correspond to that word #PoS combination, synoptic example shows in Fig 11. This approach could expand the semantic space bigger than the previously mentioned data augmentation approach (method 1), as one word could have multiple meanings of the same part of speech.

Method 3: We extract all possible meanings (synsets) of every complete word (excluding noise words, abbreviations,..., etc.), and then we retrieve the synonyms associated with every possible meaning. This process will significantly expand the semantic space of each sentence larger than the first two methods, as we are considering all possible meanings (including every PoS that this word may belong to) as well as the similar words of each meaning regardless of the coherence of the corresponding context.

Algorithm 1. Generate new list of sentences for expanded datasets Method 1

```
Input  : Load each Sentence
Output: Generate new list of sentences
Perform word Tokenize and make a list of words
for each word do
    switch method do
        case method1 do
            disambiguate with Lesk and find sense specific Synset;
            if sense specific Synset has Synonyms then
                for each synonym do
                    Replace sentence word with synonym;
                    Generate new sentence;
                    Append the new sentence to the expanded dataset;
                end
            end
        end
        case method2 do
            detect POS tag and find sense specific Synset;
            if sense specific Synset has Synonyms then
                for each synonym do
                    if synonym POS tag is equal to word POS tag then
                        Replace sentence word with synonym;
                        Generate new sentence;
                        Append the new sentence to the expanded dataset;
                    end
                end
            end
        end
        case method3 do
            find sense specific Synset if sense specific Synset has Synonyms
             then
                for each synonym do
                    Replace sentence word with synonym;
                    Generate new sentence;
                    Append the new sentence to the expanded dataset;
                end
            end
        end
        otherwise do
            Error: No such method
        end
    end
end
```

To apply the proposed methodology, we have written a python script that generates extended datasets. This achieved by following the above-described methods for each of

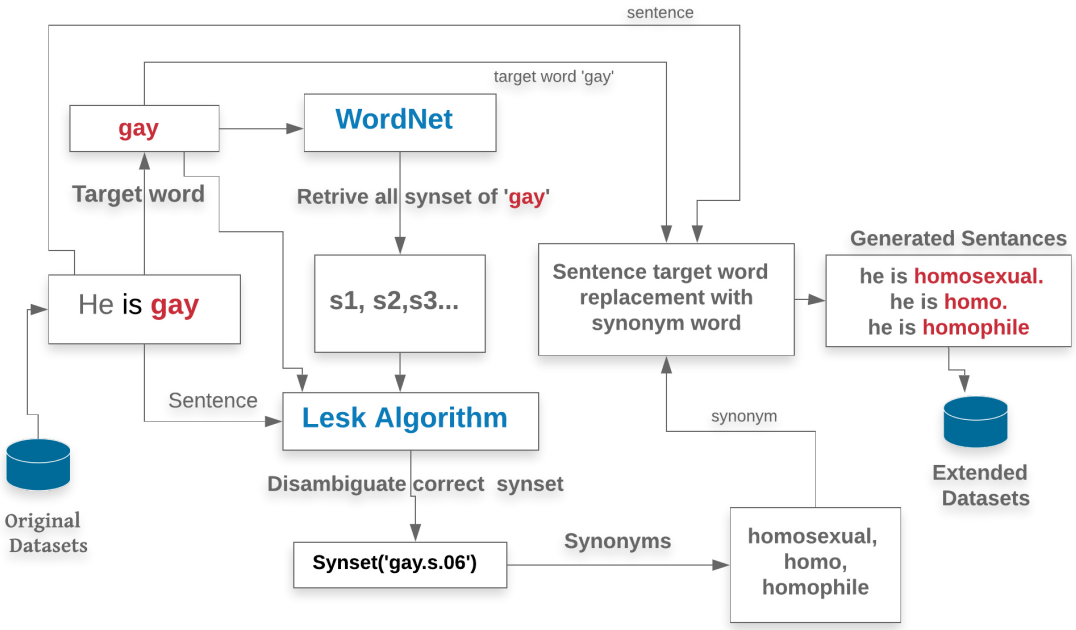

Figure 10. Example of a sentence expansion using proposed Method 1. For a target word, we calculate its corresponding list of synonyms using Wordnet. To retrieve only correct synonyms (that can be used in the context of the sentence), we insert the synonyms set and the sentence containing the target word to Lesk Algorithm. Once the disambiguation step is done, we start generating new sentences by replacing the target word with each of these synonyms and create the expanded dataset.

the original datasets. Table-15 compares the size of the original and expanded datasets. Examples of some generated sentences are provided in Table-16 by using proposed methods 1, 2, and 3. Since method-1 deals with sense disambiguation; therefore, it produces less number of sentences than the PoS tag method-2, and methods-3 generates the largest number of sentences because it considers all synonyms. One especially notices the intuitive and quality of the generated new sentences, where the algorithm successfully generated semantically similar sentences.

Table 15. Size comparison of the expanded and original Askfm dataset as well as the expanded and original Formspring dataset

Dataset Name	**Number of Sentences (size)**
Aks.fm not expanded dataset	10K
Expanded dataset 1, using Method 1	114k (11 times larger)
Expanded dataset 2, using Method 2	562k (56 times larger)
Expanded dataset 3, using Method 3	1121k (112 times larger)
Formspring not expanded dataset	12K
Expanded Dataset 4, using Method 1	136k (11 times larger)
Expanded dataset 5, using Method 2	558k (46 times larger)
Expanded dataset 6, using Method 3	1061k (88 times larger)

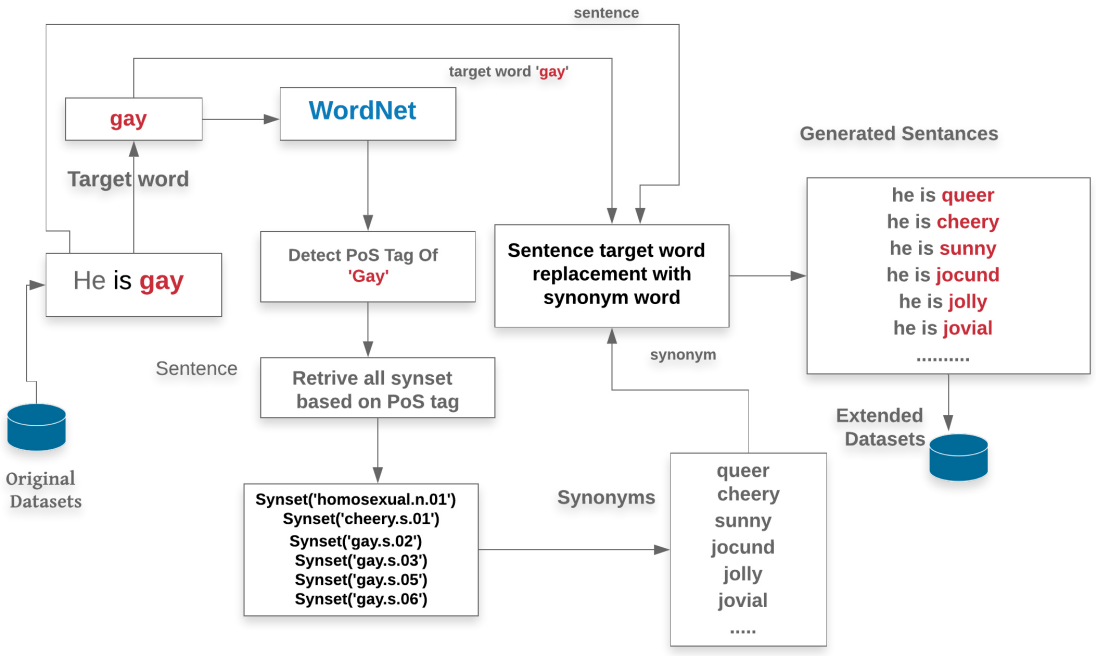

Figure 11. Example of a sentence expansion for sentence 'He is Gay', target word 'gay' by using proposed Method 2

3.3. Result Comparison

The results of the binary classification of cyberbullying identification for the original dataset and the expanded enriched datasets are summarized in Tables 17 and 18. Table 17 shows a comparison of classifier accuracy and F1 score for all four types of classifiers with 'Askfm Not Expanded Dataset,' 'Expanded Dataset 1', 'Expanded Dataset 2' and 'Expanded Dataset 3' which were generated by proposed Method 1, 2 and 3 respectively. We have observed that the CNN classifier outperformed among all other classifiers. Therefore, in Table 18, we have shown only the results of CNN classification using Word-Embeddings features and 'Askfm not expanded dataset,' 'Formspring not expanded dataset' and all 'expanded datasets' were used for result comparisons.

The results highlighted in Tables 17 and 18 indicate the following:

- Among all four types of the classifiers, CNN perform best for both initial datasets and their extended datasets. This indicates that neural network models work better than baseline classifiers.
- For base datasets, when the dataset size was small, all classifiers yield almost similar range of accuracies. However, when the same classifier was applied to extended dataset, CNN clearly outperforms when we compare to baseline classifiers. This indicates that neural network models yield better performance with large datasets compared to baseline classifiers.
- Among baseline classifiers (NB vs LR), Linear Regression models outperform Naive Bayes models in terms of accuracy and F1 scores.

Table 16. Example of generated sentences using method 1, 2 and 3

Original Sentence	**Method 1**	**Method 2**	**Method 3**
He is gay	He is homosexual He is homophile He is homo	he is cheery He is homophile He is homo he is jocund he is jolly he is jovial he is merry he is mirthful he is braw he is festal he is festive	he is queer he be gay he is festive he equal gay he constitute gay he represent gay he make up gay he comprise gay he follow gay he embody gay he personify gay he is homosexualy he is homophile he is jocund he is sunny he is jolly he is jovial he is merry he is mirthful' he is brave he is braw

Table 17. Classifier Accuracy (%) and F1 scores (%) for Askfm not expanded dataset, and its underlying expanded datasets using method 1, 2 and 3 respectively

Classifier	**Not expanded**		**Expanded D.1**		**Expanded D.2**		**Expanded D.3**	
	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1
Naive Bayes + Word Level TF-IDF	88	82	92	90	91	89	91	90
Naive Bayes + N-Gram Vectors TF-IDF	88	83	87	84	90	87	90	88
Naive Bayes + CharLevel Vector TF-IDF	88	83	90	88	88	83	88	83
Linear Classifier + Word Level TF-IDF	**90**	**88**	**93**	**96**	**94**	**94**	**95**	**95**
Linear Classifier + N-Gram Vectors TF-IDF	88	83	88	85	92	90	92	90
Linear Classifier + Char Level Vector TF-IDF	**91**	**89**	**95**	**95**	**93**	**92**	**93**	**93**
CNN + Word Embedding	**91**	**91**	**98**	**98**	**98**	**98**	**97**	**97**

Table 18. Classifier Accuracy (%) and F1 scores (%) of CNN classification using word embedding representation for original and expanded datasets

Dataset name	**Classifier**	**Acc.**	**F1**
Ask.fm not expanded	CNN + Word Embeddings	91	91
Expanded dataset 1, Method 1	CNN + WordEmbeddings	98	98
Expanded dataset 2, Method 2	CNN + WordEmbeddings	98	98
Expanded dataset 3, Method 3	CNN + WordEmbeddings	97	97
Fromspring not expanded	CNN + WordEmbeddings	95	94
Expanded dataset 4, Method 1	CNN + WordEmbeddings	99	99
Expanded dataset 5, Method 2	CNN + WordEmbeddings	99	99
Expanded dataset 6, Method 3	CNN + WordEmbeddings	98	98

- Among TF-IDF features, 'Word level TF-IDF' and 'N-gram Character Level TF-IDF' outperformed 'Word Level N-Gram TF-IDF'

- In table 18, we have observed that all the three proposed methods for data expansion yield similar scores with negligible deviation. However, the proposed method 3 shows 0.01% less accuracy compared to other 2 methods, which was dataset expansion based solely on synonyms and without considering sense. Possible explanation could be, 'Method 3' may cover some meanings that are not relevant to the words as they occur in the text. Therefore, Methods 1 & 2 work slightly better because they have been using sense disambiguation and POS tagging that are capable of targeting more sense specific synonyms.

- Classification results for extended datasets have been improved way better compared to classifiers results for initial Ask.fm and Formspring datasets. For CNN, initial Accuracy score has increased from 91% to 98% for Ask.fm dataset, and from 95% to 99% for Formspring dataset. This improvement is obvious for all four types of other classifiers. This outcome clearly indicates that semantic meaning expansion by using disambiguation and Wordnet worked very well.

Table 19. Fromspring datasets result comparison using CNN acrhitecture between Zhang et al. (2016) and ours expanded Formspring datastes

Authors name	**Classifier**	**Accuracy**	**F1 score**
Zhang	CNN	0.964	0.48
Zhang	PCNN	0.968	0.56
Ours (Expanded dataset 4, Method 1)	CNN	0.99	0.99
Ours (Expanded dataset 5, Method 2)	CNN	0.99	0.99
Ours, (Expanded dataset 6, Method 3)	CNN	0.98	0.98

Similarly to us, Zhang et al. (2016) proposed a novel cyberbullying detection with a pronunciation based convolutional neural network (PCNN) [58]. Since they used fromspring datasets as well, we highlight the comparison results of their work to ours in table 19. The comparisons of the results clearly show that both CNN and PCNN models by [58] yield max 96.8% accuracy and 56% F1 scores. However, our CNN models trained on expanded Fromspring datasets using proposed methods 1,2 and 3 yield 2.2% higher Accuracy and 43% higher F1 sores compared to previous work.

4. NEGATED DATASET EFFECT ON CYBERBULLYING

4.1. Overview

Negation exists in all forms of social languages,, and it is delivered to change the meaning of a speech's parts. It is a complex phenomenon that interrelates with many other features of the language. Moreover, the direct sense, negated statements often convey a hidden positive connotation. This part of the thesis explores the importance of both scopes of negation detection for cyberbullying and the effect of negated datasets use in cyberbullying detection.

Negation is reasonably well-understood and defined in language rules (grammar); the right ways to express a negation are expressed and documented—however, not much work has been done to identify it automatically. At first glance, nullification might appear easy to deal with. Therefore, it is common to think that the challenges could be reduced to determine negative polarity items, their scope, and reverse its polarity. Actually, it is much more problematic. Negation plays a remarkable role in understanding text and poses considerable challenges. Negation interacts with many other phenomena and is used for so many different purposes that an in-depth analysis needed. The followings are some issues found when dealing with negation. Detecting the scope of negation itself is challenging: "All tigers do not eat grass" means that the tiger does not eat grass. Another example, all the hate speeches are not cyberbullying (so out of all hate speeches, some are cyberbullying, and some are not). Besides, two negatives may cancel each other out; however, in language that is not the practical cases: "They are not unhappy", does not mean that they are happy; it means that they are not entirely unhappy, but they may not be satisfied either. Some negated sentences carry an absolute positive sense. For example, "tiger do not eat grass" implies that the tiger eats something other than grass. Otherwise, the speaker would have said, "tiger do not eat."

The above examples show the complexity of negation in the language meaning, which is similarly valid for cyberbullying context. For instance, "I hate you," and "I do not hate you" sentences are different and alter the meaning in terms of cyberbullying. This motivates the current work, aiming to contribute to the lack of scalability and significant bias observed in non-hate speech detection. For this purpose, we investigate a particular refinement of textual posts through reshaping the negation connectives in the post. This is motivated by the fact that cyberbullying can substantially be turned up or down through a simple introduction or removal of the corresponding negation token. Therefore, it is fascinating to evaluate the extent to which the negation connective can influence the performance of the cyberbullying detection algorithm.

This part is structured as follows: In the next two sections, 4.1.1 and 4.1.2 (Related Work and Negation in Natural Language), we described related works and examples about negation in NLP. In section 4.2 (Methodology), we expressed our approach to modeling and possible negation datasets creation algorithms. In Sect. 4.3 we evaluated our approach in experiments with non-negated datasets and discussed their results. Finally, we concluded by pointing out potential directions for future work in Sect. 6 and 7.

4.1.1. Related Work

Negation has broadly studied outside of computational linguistics. We have seen examples of how it is usually the most straightforward unary operator, and it reverses the truth value. One of Horn's (1989) primary work on this area presents the main thoughts in belief and psychology [59]. We follow his notion in the next two topics. Two fundamental laws given by Aristotle are the Law of Excluded Middle (LEM): This law states that in every case we must either affirm or deny), and the Law of Contradiction (LC): states, it is impossible to be and not be at the same time). LEM is not always relevant to statements concerning negation of scalar values (e.g., one can deny feeling the heat and not feeling the heat). Philosophers also recognized that a negative comment could have hidden positive sense, e.g. "King is not well" ultimately states that King is alive. Psychology has studied the constructs, practice, and cognitive processing of negation. They note that negated statements are not on similar status with a positive statement; they are diverse and subordinate kind of statements. There is an indication that children gain negation later in life than the power to communication. Psychology also affirms the constant thought that humans usually communicate in favorable terms and reserve negation mostly to describe unique or unanticipated situations.

Scholars have found negation a highly complicated phenomenon. The Cambridge Grammar of the English Language (Huddleston and Pullum 2002) applies over 60 pages to negation, polarity items (e.g., already, any), reporting verbal (e.g., he doesn't agree), non-verbal (e.g., Not all of them agree), and multiple negation [60]. Among others, negation communicates with quantifiers and anaphora [61]. For example, (1) Some of the trainees passed the test. They must have worked hard; (2) Not all the learners failed the test. They must have studied carefully. Negation also influences reasoning, analyzes the way several languages state and form negative elements as well as the definition of more than one negative element[62]. Within ordinary language processing tasks, negation has gain popularity, particularly in emotion analysis (Wilson, Wiebe, and Hoffmann 2009) and the biomedical domain [63]. The Negation and Speculation in NLP Workshop (Morante and Sporleder 2010)[64] worked on targeting negation and speculation.

The CoNLL-2010 Shared Task (Farkas et al. 2010) [65] targeted the detection of scope and their negation. Councill, McDonald, and Velikovich(2010) created their corpus to detect explicit negations and their range in a supervised manner [66]. Using the BioScope corpus, Morante and Daelemans (2009) [64] propose a monitored field detector offer syntactic rules to disclose scopes. Wiegand et al. (2010) [27] study the role of the opposite in emotion analysis. Some statements in NLP deal indirectly with negation. Amongst many others, van Munster(1988) examines contradiction for machine translation, Rose et al. (2003) [67] for text classification and Bos and Markert (2005) [68] for recognizing entailments. A work by (Abderrouaf and Oussalah 2029)[32] represents an algorithm for negation detection similar to ours where they have worked on Wikipedia dataset and Online Hate Speech Detection. As far as we are involved, we have not find many corpus that has been developed artificially for solely cyberbullying with negation detection.

4.1.2. Negation in Natural Language

In this part, we follow Huddleston and Pullum (2002) [60] to represent the properties of negation in common language with a focus on the English language. Unlike positive statements, negation is marked by words (e.g., not, no, never) or affixes (e.g., -n't, un-). Negation can communicate with other words in extraordinary ways. For example, negated sentences use various connective adjuncts similarly positive clauses: neither, nor instead of either and or. The negatively oriented polarity sentence includes words beginning with any- (anybody, anymore, anytime, etc.), the logical units- (much, till, at all, etc.), and the modal auxiliaries- (need and dare)[60]. Negation in verbs normally entails an auxiliary; if none is present, the auxiliary "do" is inserted (He eats rice vs. He didn't eat rice). We can recognize four differences for negation :

Verbal vs. Non-verbal

Verbal if the label of negation is grammatically connected with the verb (e.g., He did not eat anything at all); non-verbal if it is connected with a dependent of the verb (e.g., he ate nothing at all).

Analytics vs. Synthetic

If the negation is indicated by words whose sole syntactic function is to identify negation is represent as Analytic (e.g., Jhon did not go). However, if the words have some other functions as well, it would be Synthetic (e.g., Nobody went to the office). This example, nobody considers the negation and operates the role of AGENT.

Clausal vs. Subclausal

Clausal if the negation produces a negative clause (She didn't have a significant income); otherwise, subclausal (She had a not negligible income).

Ordinary vs. Metalinguistic

A negation is common if it means that something is not the case, e.g.,

1. She didn't have dinner with my man: he couldn't do it. On the other hand, a negation is metalinguistic if it does not oppose the truth but rather reformulates a statement, e.g.,

2. She didn't have dinner with your man: she had lunch with your father. Note that in (i) the lunch never took place, whereas in (ii) a lunch did take place.

4.1.3. Negation Handling

Negated concepts and certainty conditions which encoded within the system; thus, it enables them to distinguish between negated/uncertain concepts and factual information crucial in information retrieval. Classification of negation from free post documents is very challenging. Indeed, the existence of negation like connectives in

natural language processing (e.g., no, not, none, xx-less, among others, many context-related negated wordings) are still difficult to recognize in line with Fregean injection, which indicates, negation may happen anyplace in a sentence without making the thought undoubtedly negative.

Several text processing methods utilize hand-crafted rule-based negation/uncertainty detection modules. This involves, for instance, NegEx proposed by Chapman et al. (2001) [69] that identifies negative scope. Elkin et al. (2005) [70] used a list of negation words and a list of negation scope-ending words to define negated narratives and their range. This introduces affixal negation constructs, (e.g., either concepts with the prefixes un-, in-, dis-, a-, an-non-, im-, il-, ir-, or the suffix -less). For example, dishonesty can say, not honest. Huang and Lowe (2007) [71] achieved a hybrid method to automated negation detection by linking regular expressions matching with grammatical parsing. In this respect, negations are sorted based on syntactic classes and are located in parse trees.

Wilson et al.(2009) suggested a matching learning polarity classifier is trained with a set of negation features obtained from a list of hint words and a small window around the text[63]. On the other hand, WordNet (Miller, 1995)[72] and thesauri, such as Roget's, already provided a collection of lexical negations. In WordNet, antonymy is defined as a lexical relation between individual lexemes that have precise opposite meanings (rather than between concepts, i.e., all the members of a synset). These "direct antonym" pairs (e.g., good: bad or ugly: beautiful) are psychologically salient and have a strong associative bond between them resulting from their frequent co-occurrence [73]. "Indirect antonyms," then, result from similarity relationships determined for the members of these direct antonym pairs. For example, "moist" and "humid" are categorized as semantically related to wet and are, therefore, indirect antonyms of the lexeme would be "dry."

In our negation handling, we have used NegEx proposed by Chapman et al. (2001) [69], Elkin et al. (2005) [70] negation words list, along with PoS tagging and synonym antonym. Negation handling algorithm detailed in section 4.2.1. (Negated Dataset Creation).

4.2. Methodology

The overall methodology for effects of negated data construction includes a four-stage process same as section-3: base datasets collection; negated datasets creation (algorithm), feature engineering and classification; finally, testing and validation with initial datasets. Testing and training datasets description described in section 2.2 (Datasets Description). In the second phase, a negated datasets construction using our python scripts. The classification and feature engineering use the same as previously described in section 2.5 (Classification Architecture) and 2.4 (Feature Engineering). The final result achieved by the negation process has been duly evaluated with initial base results for both Askfm and Formspring dataset.

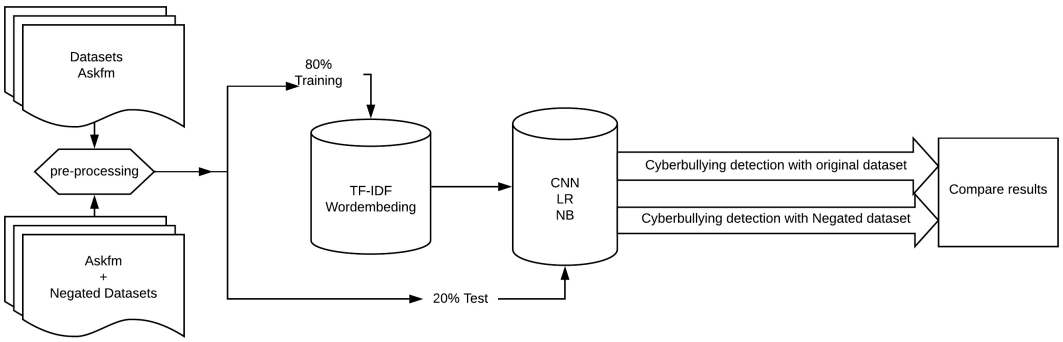

Figure 12. A general synoptic of comparing results of cyberbullying detection using Original and Negated datasets.

4.2.1. Negated Dataset Creation

Algorithm 2. Generate new list of Negation sentences.

Input : Load a sentence and initiate a new list sentence negation.
Output: Generate new list of negated sentences

```
for each sentence Identify Negated Findings by Using NegEx do
    Perform Part of Speech Tagging on the word tokenized of the sentence.
    for each word in POS tags do
        if If there is negative connective in sentence and the word belongs to
          verb or adjective forms AND antonyms exists then
            Replace the verb / adjective by its antonym, if it exists. Change the
              label of sentence.
        end
        elseIf Remove the negation and restore sentence, change the label of
          sentence.
        else Ignore negation of the sentence. Do nothing.
    end
end
```

One way to generate a non-cyberbullying and cyberbullying speech is to do the negation of the datasets. Building the negation of a sentence is not that simple, since the difficulty arises when it is a long sentence, and yet it is harder when it comes to non-formal speech (social media posts). To have a negation of a sentence, we have built a python code (algorithm 2) to do this task. The algorithm of it is as follows:

Load sentences and initiate a new lists of sentence negations. First perform negation findings by using NegEx. If there is negation contains then perform PoS Tagging on the word. Every time check if the word belongs to one of the verb forms or adjective forms, then perform either adding antonym instead of it, or add negation before it (with different forms of negations and stemming if it is a verb), or pass the word doing nothing, or remove the negation from the word.

For example, the sentence, "Alex does not like Steve Jobs" negation detection depicted in the Figure 13. Based on our algorithm, it will first be analyzed by NegEx

and will find the negation part "not Like" in the sentence. It will then perform the PoS tagging and check if the "like" word has any antonym by using python library WordNet with NLTK. Since "like" word has antonyms, it will be replaced by "hate," and the result would be, "Alex does not hate Steve job." If NLTK library fails to produce antonym, then the negation part will be removed. In that case, the output result would be "Alex does like Steve Jobs."

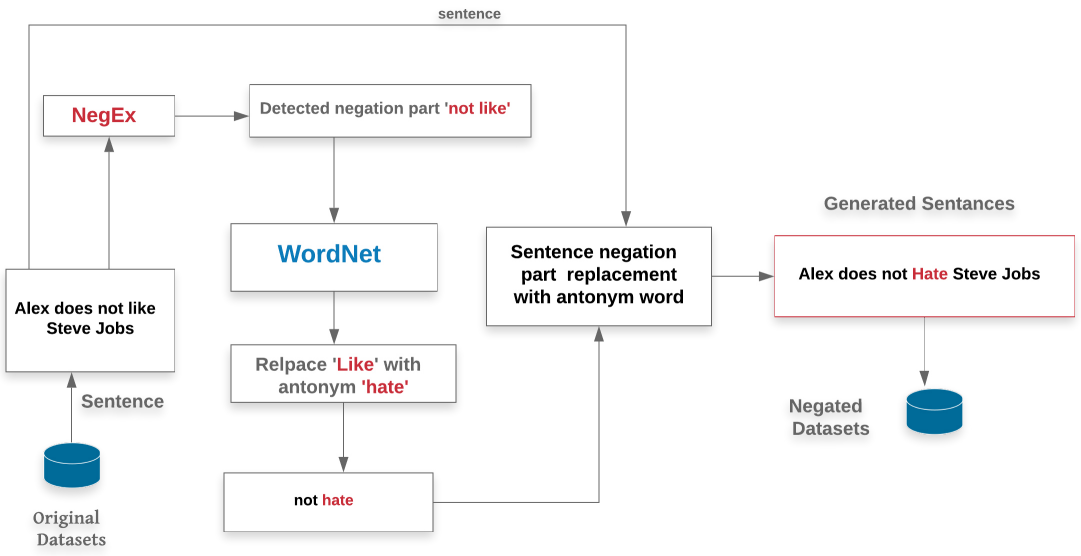

Figure 13. A general synoptic of negated sentence creation by using NegEx and WordNet.

The code of the algorithm is better seen in the provided github link in the negation Class [9]. After having the code ready to start, we took the prepared and clean datasets and pushed them into the code. This results in new datasets of the csv form containing over 1k of well-negated sentences. We take an example of the sentence before and after negation performed: simple and more complex sentences.

One can observe the negation's quality in the second sentence, how the built code could negate the present simple, add to it "do not" before, and yet stem the verb, Table 20. Another example, sentence "you are not gay" contains negation word and a bad word; therefor, our algorithm only removes the negation part "not" to make it non-negative sentence and replace the labeling non-cyber bullying to cyberbullying. This makes sense that the output sentence is well understood, and one may not detect that this is a computer-generated negation.

4.3. Result Comparison

The results highlighted in Table 21 indicate the negation effect after adding with cyberbullying datasets, and Table 22 negatioin and contcatanation with sentiment features as following:

[9] `https://github.com/saroarjahan/cyberbullying` (accessed June 03, 2020)

Table 20. Example generated negated sentences

Original Sentence	**Label**	**Generated Sentences**	**Label**
you are not gay	0	you are gay	1
go to hell	1	do not go to hell	0
dont call her bitch	0	call her bitch	1

- The results highlighted in Table 21 indicate an increasing overall performance of 2% after using negation datasets over Askfm datasets. Similar to previous results of classifications, CNN and LSTM classifiers outperformed compared to baseline classifiers. This improvement clearly shows that the useful application of negation datasets in cyberbullying.
- On the other hand, the analysis of the individual classifiers and impact of various features reveal the following: First, the accuracy and F1 performance of the classifiers show a slight increase in CNN and LASTM; however, LR with WordlevelTf-IDF and features in baseline models marginally outperforms that generated using CharacterlevelTf-IDF or N-gram (N=2,3) features.
- Third, the use of LIWC features (sentiment) in the baseline model induce increased performance 1% in terms of F1-score evaluation. The results highlighted in Table 22 have been obtained after testing the concatenation of sentiments the feature. We only reported those features and effects that yield the best overall classification results; In this case, only sentiment feature yields reasonable performance after concatenation. For instance, LR classifier's best overall performance is obtained when using a concatenation of Charecterlevel Tf-Idf and LIWC as features for the classifier.

Table 21. Classifier Accuracy & F1 scores by using Askfm and Negated 1k Datasets (best in Bold).

Classifier	Askfm		Negated 1k Datasets	
Feature Name	**Accuracy**	**F1 score**	**Accuracy**	**F1 score**
NB + WordLevel TF-IDF	88	82	89	88
NB + N-Gram TF-IDF	87	82	86	85
NB + CharLevel TF-IDF	88	83	89	88
LR + Count Vector	89	87	91	90
LR + WordLevel TF-IDF	90	88	91	91
LR + N-Gram TF-IDF	88	83	90	89
LR + Char Level TF-IDF	**91**	**89**	**92.8**	**91**
RF + WordLevel TF-IDF	88	87	91	87
RF + N-Gram TF-IDF	88	85	87	84
RF + CharLevel TF-IDF	89	87	91	88
CNN + WordEmbedding	**91**	**89**	**93**	**92**
LSTM + WordEmbeddings	90	88	91	90

Table 22. Classifier Accuracy & F1 scores after using negation datasets and concatenation of sentiment feature for Askfm datasets (best in Bold).

Classifier	Askfm		Negated 1k Datasets	
Feature Name	**Accuracy**	**F1 score**	**Accuracy**	**F1 score**
NB + CharLevel TF-IDF	88	83	89	88
NB + C.LevelTF-IDF+sentiment	88.7	83.6	89.7	88.5
LR + Char Level TF-IDF	**91**	**89**	**92.8**	**91**
LR + C.LevelTF-IDF+sentiment	**92**	**90**	**93.6**	**91.7**
RF + CharLevel TF-IDF	89	87	91	88
RF + C.LevelTF-IDF+sentiment	89.7	87.5	91.6	88.6

5. DEVELOPED TOOL

For every application, it is important to have graphical user interface (GUI) for the user. GUI can be developed by creating a desktop application, web application, mobile application, or any existing related light framework. Since this thesis is about detection of cyberbullying using social media datasets; therefore, we have chosen web UI. We have used the Django web framework for developing our web-based UI [10].

Figure 14. Developed web-application for testing usability

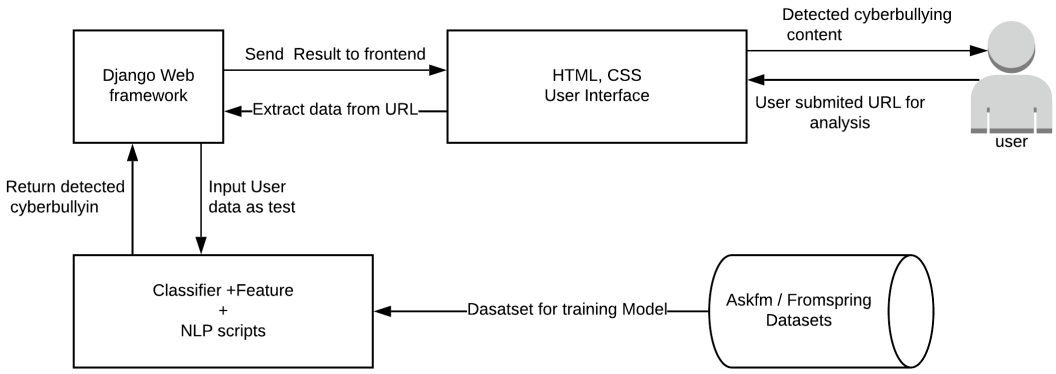

Figure 15. A general synoptic of the developed tool

Django is fast, therefore there will be no delay between backend and front-end. Furthermore, Django is developed in python platform, which allows us to integrate our NLP scripts with Django. For the front-end, we have used HTML and CSS. Our developed web application is a complete example of a working prototype that represents how this kind of application help in a real-life scenario. In GUI, we have shown classifier name, classifier accuracy, identified cyberbullying sentence, and LIWC result of positive and negative emotion of detected text. Below Figure 14

[10]https://www.djangoproject.com/

depicts the developed GUI of the project, and project source code uploaded to GitHub directory [11].

5.1. Use-Case

Our developed GUI is straightforward, where the users are able to enter the URL of the website that they want to check whether it contains any cyberbullying. Figure 15 shows general synoptic of the tool. After they submit the URL, it will be received by Django web-framework and then it will extract all the website text contents as test data. Since website text contents have many HTML tags (url, image tag, div tag...); therefore, our scripts will perform a special kind of preprocessing that will remove all unnecessary HTML tags and send a clean version of text further to use it as a test data. Meanwhile, Django backend is connected with our NLP scripts (classifier, features, text processing etc.) and training datasets. This NLP part will process input user's test data from Django and will return identified Cyberbullyings if exist in the test data. If there any cyberbullying is detected, Django will send it to the front-end for visualization.

5.2. Evaluation Methods

We used a structured interview process with the predefined System Usability Scale (SUS) questions, which are a trendy way to evaluate especially web-based tools [74]. We interviewed four participants through remote means using Zoom, an online-based teleconferencing service. Due to Covid-19, we had to continue the usability test with a few number of participants. All participants was the student of the University of Oulu and aged between 24 to 33. Participants informations were kept anonymous, and we have not provided any reward during the evaluation for avoiding biasedness. The interviews lasted between 20 to 25 minutes. When participants connected to the teleconference call on Zoom, they were first welcomed and briefed about the tools. After then, participants were given the link of the developed tools to explore the setup. And finally, we have sent them a Google evaluation form to evaluate, which contain SUS questions, and sore was between 1 (strongly disagree) to 5 (strongly agree).

The System Usability Scale Questions:

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.

[11] https://github.com/saroarjahan/Django_Online_hate_Speech_detection

6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system

Calculation of usability scores using SUS

Here is an overview of the method for the calculation of SUS score. Our users has ranked each of the ten templates questions mentioned above from 1 to 5, based on their level of agreement.

1. For each of the odd-numbered questions, we subtracted 1 from the score.
2. For each of the even-numbered questions, we subtracted their value from 5.
3. Finally, these new values were added up and multiplied by 2.5 to achieve the total SUS score as follow:

$$SUSscore = ((odd\_Q\_score - 1) + (5 - even\_Q\_score) * 2{,}5)$$

The result of all these is our system evaluation score out of 100. This is not a percentage; instead, it is a way of seeing usability of developed tools.

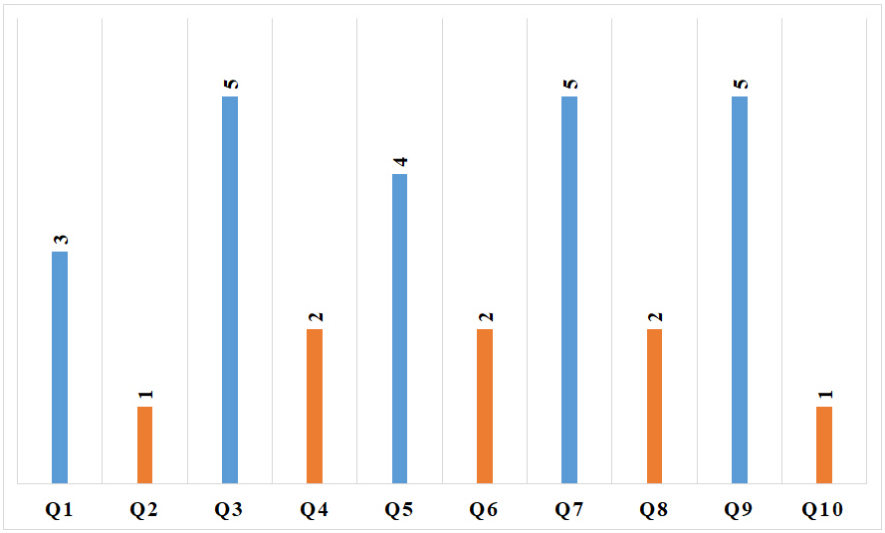

Figure 16. Average scores for ten individual SUS questions for four participants.

5.3. Evaluation Results

Figure 16 shows the average scores of ten individual SUS questions which scored by four participants. Blue color indicates the odd number results, and yellow represents even number results. Here odd-numbered results are the positive scores for the systems, which clearly shows our system is strongly recommended by most of the users, whereas even-numbered questions representing the negativity of the tool. Here we can see that most of the users strongly disagree our systems are faulty.

Figure 17 depicted the SUS individual sores for four different participants. Here we can observe that all of the user scores were between 80-85, which strongly represents our system's strong recommendation of usability.

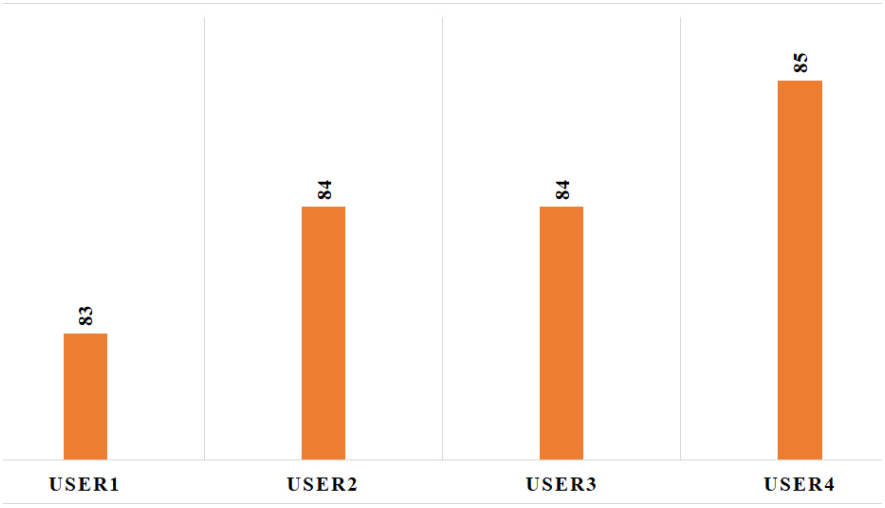

Figure 17. SUS individual scores for four participents.

The feedback was overall positive and the SUS total average score reached a result of 84. This means that the system scored Best Imaginable and users found the application as useful and will most likely recommend it to their friends [74].

6. DISCUSSION

This thesis work has yielded several important outcomes that surely can contribute to the research field of NLP and Cyberbullying Detection. Although we have achieved outstanding results, however, there is still room for further improvements. In this chapter, we will discuss what we have done so far, how we've accomplished them, and the challenges that arose during the experiment. Besides, we will also address a few limitations concerning the detection of cyberbullying and potential further improvements to enrich this area.

6.1. Literature Review

This thesis work is comprised of four different tasks (Datasets creation, Cyberbullying and categories detection, Extended datasets effect, and Negated datasets effect on cyberbullying) and all these tasks are related to Cyberbullying Detection and Improvement. We went through a number of state-of-the-art literature reviews firstly to understand the concepts of Cyberbullying and Traditional Bullying; after that, we went forward to understand the Automation Detection Technique for Cyberbullying. We have found numerous useful research works done in the last couple of years, especially for Cyberbullying Automation Detection. Most of the works are based on data collection from Social Media (SM), preprocessing of the datasets, feature engineering, and modeling. All the literature reviews helped us to understand the usefulness of deep-learning in NLP; therefore, we have used CNN and LSTM in our project along with four (4) other baseline classifiers. The literature reviews also helped us to understand the dataset collection and dataset annotation schema. We also have studied some literature reviews based on Dataset Augmentation and Negated Dataset creation; regrettably, we did not find many prominent work to be mentioned, especially in the field of cyberbullying dataset extension and negation detection.

6.2. Datasets Collection

We have collected two cyberbullying datasets- Askfm and Formspring. Since Formspring was publicly available and already was annotated, we did not face any difficulties while working with it. We have created our own dataset- Askfm; creating a dataset was a tedious process and time-consuming too, especially when it comes to annotating/ labeling the dataset. To create the dataset Askfm, we have used a website crawling technique, namely Beautifulsoup. At first, we needed to separate the native English-speaking users, which was a bit tricky to determine, but we solved this by using a VPN. Besides, it was challenging to crawl each of the profiles from top to bottom since website crawlers usually do not let us do it. We had to customize the available crawler libraries with our scripts.

Furthermore, the annotation process was extremely time-consuming. It took almost a month to manually annotate this 10k dataset. This manual annotation could be speeded up by crowdsourcing, but we did not ask for any help from crowdsourcing since we were required to ensure the possible best quality of our dataset.

6.3. Cyberbullying Detection and Improvement

This part of our thesis work is mainly a combination of four different tasks: Analysis of textual based features, Cyberbullying detection, Cyberbullying category detection, and Finding the best feature.

6.3.1. Analysis of Textual Based Feature

We used three categories of main features: Sentiment, Semantic, and Vectorization Feature. Each category has several sub-features: eleven (11) possible features under the sentiment, three (3) punctuation features, four (4) word features, and four (4) type of character features. However, we have not seen any notable increase in using semantic features as a cyberbullying detection compares to the vectorization feature. Among all of these features, 'Multiple sentiment score' outperformed the rest of the features. We have used three (3) types of TF-IDF features; between Character level and Word-level, Character level n-gram performed the best compared to word-level n-gram. During feature analysis, it was challenging to work with a vast number of features within a short time (25 features in total). Therefore, we only used the sentiment features and vectorization features throughout our experiment.

6.3.2. Cyberbullying and Category Detection

We have followed traditional ways of classification for the detection of -cyberbullying and categories of bullying. For cyberbullying detection, we have annotated datasets into two different ways- 0 and 1; however, for categories annotation we followed eight different ways. Afterwards, we pre-processed our datasets and prepared feature engineering with TF-IDF (n-gram word level and character level); finally, we used five different types of classifiers including CNN, LSTM and some baseline classifiers. We faced significant challenges while training eight (8) separate categories with CNN architecture. We have seen that deep learning (CNN) performs better than baseline classifiers, also cyberbullying detection performs much better compared to category detection.

Cyberbullying categories detection showed 10% less accuracy compare to cyberbullying post detection. These results indicated that cyberbullying category detection is tough due to training the classifier with multiple labeling simultaneously. One of the reasons could be that when we were labeling categories, the dataset was distributed to eight categories, making fewer posts labeled for each category. This lower number of data might encounter difficulties in training the models compare to only bullying detection.

6.3.3. Finding the Best Features

To find the best features, we have used an RF classifier since it has built-in functionality to find out the best features. After the analysis, we were able to determine which

features are essential, such as "you" and "f*ck" words seemed important as features. In addition, we found some less important features as well.

6.4. Datasets Extension

In this experiment, we have extended both the datasets- Askfam and Fromspring; we have proposed three (3) different methods to perform dataset augmentation: Sense disambiguation, PoS tagging, and Synonyms. All three ways yielded outstanding performance (99% accuracy). To best of our knowledge, the augmentation of dataset work was highly overlooked in NLP, especially for cyberbullying. We hope our artificially augmented dataset methods could be a significant contribution that documented for cyberbullying detection. For method-1, we have used, Lesk algorithm for sense disambiguation and WordNet for retrieving synset. However, method-2 only used PoS tagging and synonyms, and methods-3 only used synonyms. We have achieved this by using python NLTK[12] library.

After completing the Effect of the Expanded Datasets experiment, the performance increased by 5%-7%. This experiment clearly suggests the future usage of artificially extended datasets in the NLP based projects. To support our claim, we have compared our results with the results of some previously done research works, which were based on the same Fromspring dataset; our results were 4% better than the previous results.

All these methods were quite feasible to implement, except for the facts that it took a longer time to develop the scripts and to run the experiments. For example, after extending the dataset using the methods mentioned above, six (6) extended datasets were produced. Method-3 yielded 100 times larger datasets compared to base datasets; therefore, it was time-consuming to train classifiers with such a large size of datasets.

6.5. Negation Datasets Effect

We wanted to find the negation handling for cyberbullying datasets. We started with the definition of negation and reviewing other related research works. Fewer research works are available in this field, and we found the negation detection topic challenging due to the involvement of diverse complexities, including how natural language works and how people communicate in social media. In this experiment, we used datasets that have previously been used in other research works so that we can relate our results with those previous results. NegEx, PoS tagging, and antonyms replacement methodology have been used for creating negation datasets. After testing the effects of using negated datasets, it was clear that negated datasets improve the performance by 2% while compared to non negated datasets.

During the negation detection experiment, we faced difficulties dealing with the format of social media datasets since it does not follow many grammatical rules; as a result, it took much of our time to develop an algorithm that will work for every sentence for our datasets. However, we were 85% successful in targeting each of our

[12]https://www.nltk.org/

datasets sentences, and sentences which were not feasible to process simply ignored by the algorithm.

6.6. Development of GUI Tools

To develop and test a practical tool, we developed a web application. The back-end of this web application was attached to our python NLP scripts that we prepared for our thesis. This developed tool was able to check websites, online documents, SM for offensive words/comments that motivate cyberbullying. During the development process, finding the resources was challenging as we had to make the NLP scripts compatible with our system. We used the Django web framework with HTML, CSS, and JavaScript to overcome the existing challenges.

7. CONCLUSION AND FUTURE WORK

In this section of the conclusion, we provided an overview of our intentions behind this thesis work and the results that we obtained. Additionally, we addressed possible improvements that may contribute to this area.

7.1. Goals and Achievements of Work

Our primary goal for this thesis work was to improve the field of Automatic Detection of Cyberbullying, focusing on online social networks, especially on Askfm and Fromspring datasets. This primary objective branched out into two different subgoals- exploring this futuristic field of research and then implementing it to get an output with merest error. Our first task was to understand cyberbullying thoroughly, frequent targets of cyberbullying, and its consequences. Only then, we could identify and detect it successfully.

Since most of the authors collect data and classify the contents without actually making it publicly available; therefore, a common problem in any machine learning projects is the unavailability of data. Data is crucial in any research as it makes the research successful by comparing results and approaches. For this purpose, we first have collected and annotated a dataset (10k) of our own form the social network platform (ask.fm). Additionally, we used another previously labeled dataset related to cyberbullying, namely, Formspring, to compare our dataset's ground truth. We have used three categories of main features: sentiment, semantic, and vectorization. We have tested eleven (11) possible features under the sentiment feature, among which 'Multiple Sentiment Score' defeated rest of the sentiment features (76%). Because 'Multiple Sentiment Score' gave a better accuracy; we concatenated this feature with other vectorization features and have achieved an improvement of .8%, and performance improved by 1% after concatenation of three features (CharLevel + WordLevel+ Sentimen); this proves the usefulness of concatenation. For semantic features, we have carried out experiments with three (3) punctuation features, four (4) word features, and four (4) types of character features. However, we did not notice any significant improvement within using semantic features, as a cyberbullying detection compares to the vectorization feature. Among three (3) types of TF-IDF features: Character Level n-gram performed better compared to Word Level n-gram.

For cyberbullying detection, we have experimented with six (6) different types of classifiers. Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) architectures that use FastText word embedding features are contrasted with baseline algorithms constituted of Logistic Regression and Naives' Bayes classifiers. In our test, CNN with Word-Embedding yielded maximum performance with a detection accuracy of 92% for AskFm datasets and 93% for FormSpring datasets. Similar performance observed for LR with Character Lever TF-IDF features. Among all the classifiers, NB and SVM were the least performed classifiers; however, the RF classifier showed a decent performance.

Similarly, for cyberbullying categories detection, CNN and LR performed best among all other classifiers, yielding 81.2% and 80% of accuracies. Cyberbullying categories detection showed 10% less accuracy than cyberbullying-related post-

detection, which suggests that cyberbullying categories' detection is much more complicated than cyberbullying post-detection. We have found words like 'You' and 'F**k' as features that play a significant role in cyberbullying detection during the best feature selection experiment.

The second part of the thesis deals with semantic-meaning expansion using sense-disambiguation for cyberbullying datasets and compares identification using original feature engineering. The methodology is tested the same as previous datasets-Askfm and Fromspring, and six (6) artificially generated datasets. The testing results demonstrate the feasibility of the extended datasets for semantic meaning expansion, which clearly shows an outstanding performance. On the other hand, the superiority of the constructed CNN and LSTM based classifiers in the overall classification for all datasets is clearly emphasized. We have run several experiments and disambiguation/semantic expansion, to estimate the impact of the classification. Finally, we have compared the accuracy score for cyberbullying detection with some widely used classifiers before and after the expansion of datasets. This research outcome was promising and yielded 99% of the classifier's accuracy, which is a 5% improvement from the base score (93%).

Our final goal derived from a lack of work in the area of negation scope detection and its effect on cyberbullying detection. Our proposed approach advocates a classification like technique by using NegEx and POS tagging that uses a particular data design procedure for negation detection. We compared cyberbullying detection results after using negated datasets based on NegEx, and PoS tagging and antonyms replacement. After using the negated dataset, we achieved a 95% accuracy, which yielded overall accuracy improvement of 2% from the base score (93%).

In all the experiments mentioned above, the CNN classifier outperforms the rest of the classifiers, emphasizing on the usage of deep learning in NLP projects.

7.2. Future Work

Cyberbullying detection is a research field where plenty of advancements can be made. We believe that the starting point is to uniformize the definition of cyberbullying globally since only a model won't be able to induce something we, the humans, aren't fully knowledgeable about. Providing stricter rules and more powerful guidelines might be some significant steps towards homogenizing the concept, which is still unclear and different among countries.

Regarding the feature extraction and selection process, we believe we have tested several standard text-based features but did not receive promising results except for the sentiment feature. However, soon new features like- user profiling could also be tested for cyberbullying. As mentioned before, social network posts are often short, ambiguous, and contain typos and abbreviations, which sometimes make it hard to extract relevant patterns. We think user profiling techniques should be explored thoroughly, with particular attention to the network-based features that may obtain good results. We can also think of using the embedding features in baseline models; such an approach has not been conducted to ease the comparison with other related works and avoid the classifiers' computational explosions.

Furthermore, most of the works for cyberbullying are related to the detection of bully's existence only rather than the detection of categories or types of bullies. In our work, we have made an initial approach for detecting cyberbullying categories for the first time. Since our dataset was small and dealing with eight different categories required much larger dataset, we would like to continue this test with big datasets. Besides, labeling for multiple categories may place the same post in different categories. For example, a post 'I wanna post your nude' could be categorized as defamation and Threat. In the future, we would like to consider these cases and will work on developing more sophisticated labeling and classification techniques.

Moreover, this is our initial work for cyberbullying detection, and we strongly believe it paves the way for improved identification of bullying intentions on social media. The disambiguation and the semantic expansion used in this work are specific to cyberbullying datasets and can also be exploited by many other NLP based datasets. However, sometimes synonyms can alter the meaning in a particular context. We would like to develop further precise algorithms that can target proper synonyms to expand semantic meaning without changing the context. Moreover, we would like to improve the sense of disambiguation tasks, particularly related to cyberbullying topics, using deep learning.

8. ACKNOWLEDGMENT

This work is partly supported by European project YoungRes (#823701), which is gratefully acknowledged.

9. APPENDIX

9.1. Publications

We have submitted two papers in two different international conferences during this thesis work: Conference on Neural Information Processing Systems (NeurIPS), and International Conference on Computational Linguistics (COLING) under the SemEval-2020 competition. Papers title are as follow:

1. Cyberbullying detection with WordNet-based semantic expansion and word disambiguation (NeurIPS 2020: Submission 2401 undergoing full review process).

2. Md Saroar Jahan and Mourad Oussalah. 2020. Team Oulu at SemEval-2020 Task 12: Multilingual Identification of Offensive Language, Type and Target of Twitter Post Using Translated Datasets. Accepted in International Workshop on Semantic Evaluation (SemEval).[75]

10. REFERENCES

[1] Ekedahl J. & Golub K. (2004) Word sense disambiguation using wordnet and the lesk algorithm. Projektarbeten 2004 17.

[2] on Addiction N.C., at Columbia University (CASA) S.A. & of America U.S. (2003) National survey of american attitudes on substance abuse xiii: Teens and parents .

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

[4] Harper D. et al. (2001) Online etymology dictionary .

[5] Shiels M. (2003) A chat with the man behind mobiles. BBC News UK Online .

[6] Lenhart A., Ling R., Campbell S. & Purcell K. (2010) Teens and mobile phones: Text messaging explodes as teens embrace it as the centerpiece of their communication strategies with friends. Pew Internet & American Life Project .

[7] Subrahmanyam K. & Greenfield P. (2008) Online communication and adolescent relationships. The future of children , pp. 119–146.

[8] Belsey B. (2005) Cyberbullying: An emerging threat to the "always on" generation. Recuperado el 5, p. 2010.

[9] Smith P.K., Catalano R., Slee P., Morita Y., Junger-Tas J. & Olweus D. (1999) The nature of school bullying: A cross-national perspective. Psychology Press.

[10] Hinduja S. & Patchin J.W. (2014) Bullying beyond the schoolyard: Preventing and responding to cyberbullying. Corwin Press.

[11] Smith P.K., Mahdavi J., Carvalho M., Fisher S., Russell S. & Tippett N. (2008) Cyberbullying: Its nature and impact in secondary school pupils. Journal of child psychology and psychiatry 49, pp. 376–385.

[12] Willard N. (2003) Off-campus, harmful online student speech. Journal of School Violence 2, pp. 65–93.

[13] Slonje R. & Smith P.K. (2008) Cyberbullying: Another main type of bullying? Scandinavian journal of psychology 49, pp. 147–154.

[14] Menesini E. & Nocentini A. (2009) Cyberbullying definition and measurement: Some critical considerations. Zeitschrift für Psychologie/Journal of Psychology 217, pp. 230–232.

[15] Dredge R., Gleeson J. & De la Piedad Garcia X. (2014) Cyberbullying in social networking sites: An adolescent victim's perspective. Computers in human behavior 36, pp. 13–20.

[16] Kowalski R., Limber S. & Agatston P. (2008) Cyber bullying: Bullying in the digital age. malden, massachusetts. Blackwell Publishing. Li, Q.(2006). Cyber bullying in schools: a research of gender differences. School Psychology International 27, pp. 157–170.

[17] Dooley J.J., Pyżalski J. & Cross D. (2009) Cyberbullying versus face-to-face bullying: A theoretical and conceptual review. Zeitschrift für Psychologie/Journal of Psychology 217, pp. 182–188.

[18] Coyne I., Chesney T., Logan B. & Madden N. (2009) Griefing in a virtual community: An exploratory survey of second life residents. Zeitschrift für Psychologie/Journal of psychology 217, pp. 214–221.

[19] Wolak J., Mitchell K.J. & Finkelhor D. (2007) Does online harassment constitute bullying? an exploration of online harassment by known peers and online-only contacts. Journal of adolescent health 41, pp. S51–S58.

[20] Law D.M., Shapka J.D. & Olson B.F. (2010) To control or not to control? parenting behaviours and adolescent online aggression. Computers in Human Behavior 26, pp. 1651–1656.

[21] Fauman M.A. (2008) Cyber bullying: Bullying in the digital age. American Journal of Psychiatry 165, pp. 780–781.

[22] Juvonen J. & Gross E.F. (2008) Extending the school grounds?—bullying experiences in cyberspace. Journal of School health 78, pp. 496–505.

[23] Smith P.K. (2012) Cyberbullying: Challenges and opportunities for a research program—a response to olweus (2012). European Journal of Developmental Psychology 9, pp. 553–558.

[24] Pyżalski J. (2012) From cyberbullying to electronic aggression: typology of the phenomenon. Emotional and behavioural difficulties 17, pp. 305–317.

[25] Nobata C., Tetreault J., Thomas A., Mehdad Y. & Chang Y. (2016) Abusive language detection in online user content. In: Proceedings of the 25th international conference on world wide web, pp. 145–153.

[26] Burnap P. & Williams M.L. (2015) Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making. Policy & Internet 7, pp. 223–242.

[27] Wiegand M., Siegel M. & Ruppenhofer J. (2018) Overview of the germeval 2018 shared task on the identification of offensive language .

[28] Foong Y.J. & Oussalah M. (2017) Cyberbullying system detection and analysis. In: 2017 European Intelligence and Security Informatics Conference (EISIC), IEEE, pp. 40–46.

[29] Kumar R., Ojha A.K., Malmasi S. & Zampieri M. (2018) Benchmarking aggression identification in social media. In: Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018), pp. 1–11.

[30] Reynolds K., Kontostathis A. & Edwards L. (2011) Using machine learning to detect cyberbullying. In: 2011 10th International Conference on Machine learning and applications and workshops, vol. 2, IEEE, vol. 2, pp. 241–244.

[31] Malmasi S. & Zampieri M. (2017) Detecting hate speech in social media. arXiv preprint arXiv:1712.06427 .

[32] Abderrouaf C. & Oussalah M. (2019) On online hate speech detection. effects of negated data construction. In: 2019 IEEE International Conference on Big Data (Big Data), IEEE, pp. 5595–5602.

[33] Kwok I. & Wang Y. (2013) Locate the hate: Detecting tweets against blacks. In: Twenty-seventh AAAI conference on artificial intelligence.

[34] Hinduja S. & Patchin J.W. (2008) Cyberbullying: An exploratory analysis of factors related to offending and victimization. Deviant behavior 29, pp. 129–156.

[35] Nadali S., Murad M.A.A., Sharef N.M., Mustapha A. & Shojaee S. (2013) A review of cyberbullying detection: An overview. In: 2013 13th International Conference on Intellient Systems Design and Applications, IEEE, pp. 325–330.

[36] Yin D., Xue Z., Hong L., Davison B.D., Kontostathis A. & Edwards L. (2009) Detection of harassment on web 2.0. Proceedings of the Content Analysis in the WEB 2, pp. 1–7.

[37] Dinakar K., Jones B., Havasi C., Lieberman H. & Picard R. (2012) Common sense reasoning for detection, prevention, and mitigation of cyberbullying. ACM Transactions on Interactive Intelligent Systems (TiiS) 2, pp. 1–30.

[38] Dadvar M., Trieschnigg D. & de Jong F. (2014) Experts and machines against bullies: A hybrid approach to detect cyberbullies. In: Canadian Conference on Artificial Intelligence, Springer, pp. 275–281.

[39] Nahar V., Al-Maskari S., Li X. & Pang C. (2014) Semi-supervised learning for cyberbullying detection in social networks. In: Australasian Database Conference, Springer, pp. 160–171.

[40] Agrawal S. & Awekar A. (2018) Deep learning for detecting cyberbullying across multiple social media platforms. In: European Conference on Information Retrieval, Springer, pp. 141–153.

[41] ElSherief M., Nilizadeh S., Nguyen D., Vigna G. & Belding E. (2018) Peer to peer hate: Hate speech instigators and their targets. In: Twelfth International AAAI Conference on Web and Social Media.

[42] Van Rijsbergen C.J., Robertson S.E. & Porter M.F. (1980) New models in probabilistic information retrieval. British Library Research and Development Department London.

[43] Maitra P. & Sarkhel R. (2018) A k-competitive autoencoder for aggression detection in social media text. In: Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018), pp. 80–89.

[44] Frenda S., Bilal G. et al. (2018) Exploration of misogyny in spanish and english tweets. In: Third Workshop on Evaluation of Human Language Technologies for Iberian Languages (IberEval 2018), vol. 2150, Ceur Workshop Proceedings, vol. 2150, pp. 260–267.

[45] Saif H., Fernández M., He Y. & Alani H. (2014) On stopwords, filtering and data sparsity for sentiment analysis of twitter .

[46] Watanabe H., Bouazizi M. & Ohtsuki T. (2018) Hate speech on twitter: A pragmatic approach to collect hateful and offensive expressions and perform hate speech detection. IEEE Access 6, pp. 13825–13835.

[47] Hutto C.J. & Gilbert E. (2014) Vader: A parsimonious rule-based model for sentiment analysis of social media text. In: Eighth international AAAI conference on weblogs and social media.

[48] Kim Y. (2014) Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882 .

[49] van Aken B., Risch J., Krestel R. & Löser A. (2018) Challenges for toxic comment classification: An in-depth error analysis. arXiv preprint arXiv:1809.07572 .

[50] Ho T.K. (1995) Random decision forests. In: Proceedings of 3rd international conference on document analysis and recognition, vol. 1, IEEE, vol. 1, pp. 278–282.

[51] Breiman L. (2000) Some infinity theory for predictor ensembles. Tech. rep., Technical Report 579, Statistics Dept. UCB.

[52] Lesk M. (1986) Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In: Proceedings of the 5th annual international conference on Systems documentation, pp. 24–26.

[53] Slonje R., Smith P.K. & Frisén A. (2013) The nature of cyberbullying, and strategies for prevention. Computers in human behavior 29, pp. 26–32.

[54] Van Hee C., Lefever E., Verhoeven B., Mennes J., Desmet B., De Pauw G., Daelemans W. & Hoste V. (2015) Detection and fine-grained classification of cyberbullying events. In: International Conference Recent Advances in Natural Language Processing (RANLP), pp. 672–680.

[55] Warner W. & Hirschberg J. (2012) Detecting hate speech on the world wide web. In: Proceedings of the second workshop on language in social media, Association for Computational Linguistics, pp. 19–26.

[56] Djuric N., Zhou J., Morris R., Grbovic M., Radosavljevic V. & Bhamidipati N. (2015) Hate speech detection with comment embeddings. In: Proceedings of the 24th international conference on world wide web, pp. 29–30.

[57] Naskar S.K. & Bandyopadhyay S. (2007) Word sense disambiguation using extended wordnet. In: 2007 International Conference on Computing: Theory and Applications (ICCTA'07), IEEE, pp. 446–450.

[58] Zhang X., Tong J., Vishwamitra N., Whittaker E., Mazer J.P., Kowalski R., Hu H., Luo F., Macbeth J. & Dillon E. (2016) Cyberbullying detection with a pronunciation based convolutional neural network. In: 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA), IEEE, pp. 740–745.

[59] Horn L.R. & Kato Y. (2000) Negation and polarity: Syntactic and semantic perspectives. OUP Oxford.

[60] Huddleston R., Pullum G.K. et al. (2002) The cambridge grammar of english. Language. Cambridge: Cambridge University Press 1, p. 23.

[61] Hintikka J. (2002) Hyperclassical logic (aka if logic) and its implications for logical theory. Bulletin of Symbolic Logic 8, pp. 404–423.

[62] Dowty D. (1994) The role of negative polarity and concord marking in natural language reasoning. In: Semantics and Linguistic Theory, vol. 4, vol. 4, pp. 114–144.

[63] Wilson T., Wiebe J. & Hoffmann P. (2009) Recognizing contextual polarity: An exploration of features for phrase-level sentiment analysis. Computational linguistics 35, pp. 399–433.

[64] Morante R. & Daelemans W. (2009) Learning the scope of hedge cues in biomedical texts. In: Proceedings of the BioNLP 2009 Workshop, pp. 28–36.

[65] Vincze V., Szarvas G., Farkas R., Móra G. & Csirik J. (2008) The bioscope corpus: biomedical texts annotated for uncertainty, negation and their scopes. BMC bioinformatics 9, pp. 1–9.

[66] Councill I.G., Mcdonald R. & Velikovich L. Velikovich l: What's great and what's not: learning to classify the scope of negation for improved sentiment analysis. In: In Proceedings of the Workshop on Negation and Speculation in Natural Language Processing, Citeseer.

[67] Rosé C., Roque A., Bhembe D. & Vanlehn K. (2003) A hybrid text classification approach for analysis of student essays. In: Proceedings of the HLT-NAACL 03 workshop on building educational applications using natural language processing, pp. 68–75.

[68] Bos J. & Markert K. (2005) Recognising textual entailment with logical inference. In: Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, pp. 628–635.

[69] Chapman W.W., Bridewell W., Hanbury P., Cooper G.F. & Buchanan B.G. (2001) A simple algorithm for identifying negated findings and diseases in discharge summaries. Journal of biomedical informatics 34, pp. 301–310.

[70] Elkin P.L., Brown S.H., Bauer B.A., Husser C.S., Carruth W., Bergstrom L.R. & Wahner-Roedler D.L. (2005) A controlled trial of automated classification of negation from clinical notes. BMC medical informatics and decision making 5, p. 13.

[71] Huang Y. & Lowe H.J. (2007) A novel hybrid approach to automated negation detection in clinical radiology reports. Journal of the American medical informatics association 14, pp. 304–311.

[72] Miller G.A. (1995) Wordnet: a lexical database for english. Communications of the ACM 38, pp. 39–41.

[73] Fellbaum C. (1998) A semantic network of english: the mother of all wordnets. In: EuroWordNet: A multilingual database with lexical semantic networks, Springer, pp. 137–148.

[74] Bangor A., Kortum P. & Miller J. (2009) Determining what individual sus scores mean: Adding an adjective rating scale. Journal of usability studies 4, pp. 114–123.

[75] Jahan M.S. & Oussalah M. (2020) Team oulu at semeval-2020 task 12: Multilingual identification of offensive language, type and target of twitter post using translated datasets. In: Proceedings of the International Workshop on Semantic Evaluation (SemEval).