Identification and Classification of Sexual Predatory Behavior in Online Chat-Room Environments

Norbert Eke
UBC Okanagan
Kelowna, BC, Canada
norberteke@protonmail.com

Abdallah Mohamed
UBC Okanagan
Kelowna, BC, Canada

Jeffrey Andrews
UBC Okanagan
Kelowna, BC, Canada

{abdallah.mohamed, jeff.andrews}@ubc.ca

Abstract

According to the *Crimes Against Children Research Center*, one in five U.S. teenagers who regularly use the Internet have received an unwanted sexual solicitation via the web. There is an increasing danger in online environments such as chat-rooms, where predatory behaviour is more and more frequent, creating an unsafe environment for minors. This project aims to design an approach for online communities to enhance their member's safety by detecting malicious conversations of sexual nature. This project joins the powers of computational linguistics with statistical machine learning to decipher the insight lying in conversations, then make predictions on whether or not a specific conversation should be flagged for containing sexual predatory behaviour. The contribution of this novel approach is 2-fold: firstly, the approach is able to capture the contextual details by putting an emphasis on insight that lies within the conversation, and secondly it contains a 2 stage classification system, which is highly flexible and customizable for detecting and classifying other malicious textual data.

1 Introduction

Members of online communities face more danger than ever before, with a significant increase in the number of aggressive sexual predators present online (Wolak et al., 2008). Some statistics say "most first encounters between offenders and victims (76%) happened in online chat rooms" (Wolak et al., 2004). According to Wolak et al. (2008), online sexual predators "do not appear to be stalking unsuspecting victims, but rather con-tinuing to seek youth who are susceptible to seduction". Now more than ever there is a need for better intelligent systems that are capable of accurately detecting sexual predator's dangerous behavior online.

1.1 Previous Work

A variety of research has been done on malicious and dangerous behavior detection and classification based on textual analysis. The detection of online bullying has been previously addressed by numerous researchers like (Nandhini and Sheeba, 2015), and (Nalini and Sheela, 2015), who considered analyzing textual data from social networking websites. Other types of dangerous behavior detection include crime pattern detection (Nath, 2006), harmful language detection (Munezero et al., 2013), deception detection (Fuller et al., 2011), depression detection (Wang et al., 2013), and sexual predator detection (Inches and Crestani, 2012). The majority of these approaches apply feature extraction and text mining techniques, alongside topic modeling or learning algorithms, before finishing it off with text classifiers predicting for new textual data.

Sexual predator identification became a popular research topic in 2012, when Inches and Crestani created labeled data-sets for a sexual predator identification competition[1]. The two main tasks of the competition included the identification of predators within all users and identification of parts of predator conversations which are the most distinctive of the predator bad behavior. Several approaches have been submitted to this competition in the past. (Morris and Hirst, 2012) used SVM classification[2] with a bag-of-words model, alongside a manually constructed "blacklist" of words. (Vilariño et al., 2012) used a Multino-

---

[1] http://pan.webis.de/clef12/pan12-web/author-identification.html
[2] (Cortes and Vapnik, 1995)

mial Naive Bayes[3] model for predator classification alongside an information retrieval based approach. (Brooke and Hirst, 2012) looked at clustering paragraphs using Latent Semantic Indexing[4] to find paragraphs containing sexual predatory behaviour, while (Bogdanova et al., 2012) used language based feature extraction techniques with a Naive Bayes classifier to detect predators. Villatoro-Tello et al. applied a two step approach to detect misbehaving users by focusing on "suspicious conversations identification" and "victim from predator disclosure" (Villatoro-Tello et al., 2012). Finally, (Kang et al., 2012) applied k-NN classification[5] alongside an information retrieval based approach to detect "abnormal chat users".

The common element in almost all approaches is a machine learning model acting as a text classifier. There has been extensive research done on which learning algorithms could be best applied to text classification. (Lilleberg et al., 2015) showed that SVM classifiers applied with weighted Word2Vec[6] models work well for text classification tasks. In (Bijalwan et al., 2014)'s work one could observe how k-NN classifier outperformed Naive Bayes and Term-Graph models for the task of text classification. However, (Selvi et al., 2017)'s work suggest that the Rocchio and Random Forest Algorithms[7] work better than fuzzy relevance clustering, Multi-label k-NN or Naive-Bayes Algorithms for text categorization. Finally, (Shafiabady et al., 2016) proposed a new way of performing text classification by applying an unsupervised clustering approach to train SVM classifiers, which resulted in reducing the high dimensional nature of the textual data.

1.2 Contribution and Significance

The contribution of this project is 2-fold. Firstly, a new approach is proposed to sexual predator identification by putting an emphasis on the insight that lies within the contextual details of a conversation. Research is focused on analyzing the entire conversation and deciding if it contains sexual predatory behaviour, as opposed to previous approaches, which focused on either identifying the predators among all users in the different con-

versations or identifying the parts of the conversations which are the most distinctive of the predator behaviour. What makes this approach unique is the emphasis on contextual details. Secondly, a new two stage classification system is introduced to detect potential predatory behaviour and classify them into 3 groups of potential levels of maliciousness. The experimental process of coming up with the classification system yielded performance results from 8 different classification models, which is also part of the contribution.

The significance of this project is 3-fold. Firstly, this approach significantly helps online communities to enhance their member's safety by detecting malicious conversations of sexual nature. Secondly, the algorithm designed in this project is significant, since it furthers research conducted in the area of sexual predator detection. Thirdly, the two stage classification system is a highly flexible method, therefore future research can be focused on customizing this approach to other types of dangerous behavior detection mentioned in the previous section.

2 Research Questions and Objectives

The main goal of this research project is to design an approach that can detect and classify textual data as containing sexual predatory or non-predatory behaviour. Applying such an approach to online chat-room conversations would have to take into consideration the messy and difficult to interpret nature of the text. Textual data coming from online environments could be expected to be full of misspelled words, slang, internet acronyms, inappropriate language and broken grammar being used. One of the major tasks in this project is reconstruction of the linguistic context of words and phrases being used in a conversation. Modern computational linguistics use complex deep learning language models to capture the semantic similarity between words and concepts, thus mapping the "meaning of words and concepts" to high dimensional word vector spaces.

This research project aims to join the powers of computational linguistics with statistical machine learning to decipher the insight lying in conversations, then make predictions on whether or not a specific conversation should be flagged for containing sexual predatory behaviour. Such a powerful approach could be applied in online gaming chat-room environments, where children could be

---

[3]Multinomial Naive Bayes classifier: (McCallum et al., 1998)
[4]Latent Semantic Indexing: (Deerwester et al., 1990)
[5]k-NN classification: (Ripley, 2007)
[6]Word2Vec model: (Mikolov et al., 2013a)
[7]Random Forest: (Breiman, 2001)

disposed to being manipulated or preyed on by sexual predators. This project's sole purpose is to design an approach that could help online communities keep their members safer using an automated chat-room filter, which goes through conversations and classifies them into groups based on the uncertainty of detected malicious content.

This research project initially aimed to answer the following research questions:

Q1: How to make modern computational language techniques and models adapt to the difficult to interpret nature of online conversations?

Q2: How to extract semantic details' from conversations, such that conversations containing malicious intent could be detected?

Q3: What kind of classification system and what statistical machine learning models can make a difference and predict whether or not a conversation contains sexual predatory behavior?

3 Algorithm Proposed

The input of this algorithm is parsed as a CSV file containing all conversation data. This file should have individual columns for conversation-id, conversation line number, author/ author-id for the line, text content in that line, and a conversation label (being a binary value of 0 or 1, 0 meaning the conversation contains no predatory behavior, 1 meaning predatory behavior).

3.1 Data Manipulation and Text Pre-processing

Given a CSV file containing all conversation data, one needs to parse each conversation to combine all lines within a conversation into one textual observation, then pre-process the text, and attach the appropriate labels to each conversation. For text pre-processing a number of text cleaning techniques have been considered, like removal of extra white spaces and HTML tags or links, also removal of numerical characters. All characters have been converted to lowercase and the auto-correct[8] (McCallum, 2016) library's spelling corrector was also used. Conversations containing less than 3 words would be removed. The pre-processed text output is a text file, with each line containing a conversation label and the entire conversation.

---

[8]https://github.com/phatpiglet/autocorrect

3.2 Vector Representation of Words

In order to create quantitative attributes about the content of conversations, the use of vector representation of words was needed. One of the most famous deep learning models that allows the computation of vector representations of words is Google's Word2Vec.

Word2Vec is a deep learning model takes a textual data as input and produces high dimensional vector representations of words as output. In a Google blog post the algorithm is described as "it constructs a vocabulary from the training text data and then learns vector representation of words. The resulting word vector file can be used as features in many natural language processing and machine learning applications" (Mikolov et al., 2013c). These word vectors are also called word embeddings, and they are the output of the Word2Vec model. The model is trained to reconstruct the linguistic context of words by placing the vectors of words used in the same context close to each other. Word embeddings are also useful, since they are "good at capturing syntactic and semantic regularities in language" (Mikolov et al., 2013d), by placing the vectors of similar words closer to each other. Figure 5 offers a visual explanation of how Word2Vec works and how it is used.
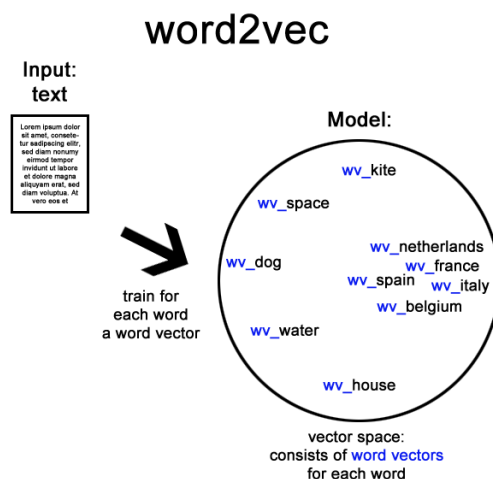

Figure 1: Detailed Overview of the Deep Learning Model Called Word2Vec. Image credits go to (L, 2015).

More details about the model architecture of Word2Vec can be found in Mikolov et al's paper (Mikolov et al., 2013a).

If dimensionality reduction would be performed

on word embeddings, these semantic regularities be visualized in so called low dimension word vector spaces. A good example of such vector space can be seen in Figure 2.
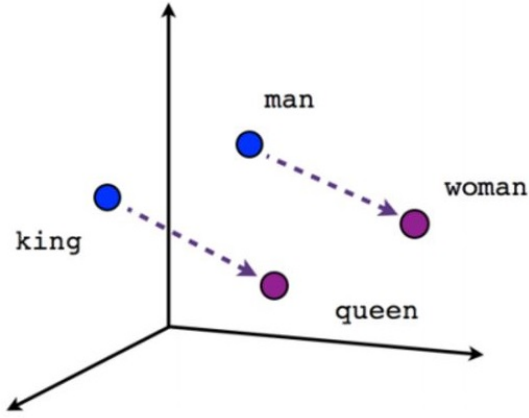

Figure 2: Photo Credit to (NSS, 2017). Three dimensional vector space, which contains the vectors 'Man', 'Woman', 'King', 'Queen'. 'Man' and 'Woman' are related to each other the same way as 'King' and 'Queen' are related to each other, since the vectors in between them describe the male-female relationship.

In terms of the algorithm proposed, these word embeddings were created by taking all pre-processed words used in all available conversations and train a large Word2Vec model, which converts each unique word into a high dimensional vector. Each unique word became part of a training text corpus that was used as input for the Word2Vec model. In order to train the Word2Vec model, Gensim[9] (Řehůřek and Sojka, 2010) python library's implementation of Word2Vec was used. Gensim's implementation requires the text corpus and size of word vectors as parameters. $size = 400$, $workers = 20$ and a text file containing a pre-processed conversation for each line were the parameters given for the trained model. $size = 400$ was chosen on an experimental basis, as 100, 200, 300, 400 and 500 were considered, but word vectors of size 400 appeared to work best with the classifications models discussed later. The Word2Vec model created a vector representation of each unique word used somewhere within a conversation, thus a conversation can be represented as set of n word vectors, where n is the number of words present in that specific conversation.

---

[9]https://radimrehurek.com/gensim/

3.3 Feature Extraction using Word Embedding Aggregation

The most important aspect of this algorithm is putting the emphasis on contextual details within the conversation. More specifically, the emphasis is on the insight found in the contextual details of a conversation. Having a 400 dimensional word vector represent each word and n word vectors represent a conversation provides the opportunity to dig deep into the contextual details of conversations and potentially find some insight about the content of the conversation.

Using the word vectors coming from the Word2Vec model is a good starting point for learning the context of a conversation, but a better approach includes an extra step, which is feature extraction. Traditionally, feature extraction is done with the help of dimensional reduction algorithms such as Principle Component Analysis, or variable selection methods such as LASSO. Word embeddings are used to extract contextual details from textual data, therefore a word vector specific technique called "word embedding aggregation" (De Boom et al., 2016) was used to extract features. De Boom et al. noted that aggregating word embeddings through a mean, max, min function is one of the easiest and most widely used techniques to derive sentence embeddings.

In the data manipulation section it was mentioned that each conversation is being concatenated into one very long sentence, which seemed like an odd choice for data manipulation. The real purpose of conversation concatenation is to set up conversations to be aggregated into "conversation embeddings", using Word2Vec embeddings and De Boom et al.'s word embedding aggregated sentence embeddings. Based on De Boom et al.'s results, the algorithm makes use of coordinate-wise min and max functions applied to each word embedding within a conversation. This approach works well, assuming that "conversation embeddings" behave the same way as sentence embeddings. If the vectors for the n words in the conversation are $v_1, v_2, \ldots, v_n \in \mathbb{R}_d$, then the computation of $\min(v_1, v_2, \ldots, v_n)$ and $\max(v_1, v_2, \ldots, v_n)$ are needed. What this means is that the coordinate-wise minimum vector and maximum vector of n word vectors within a conversation become 2 separate feature vectors.

Finally, as suggested in De Boom et al.'s work, the concatenation of these two feature vectors re-

sults in obtaining a coordinate-wise min-max feature vector for feature extraction purposes. In this project, the word embeddings are 400 dimensional, so the concatenation of coordinate-wise minimum and maximum vectors results in an 800 dimensional feature vector.

3.4 Two Stage Classification System

After performing feature extraction using word embedding aggregation, the 800 dimensional feature vectors can be fed into various classification models. These machine learning classification models will separate conversations that contain predatory behaviour from the ones that do not contain. Various binary classification models were considered: Linear Discriminant Analysis (Ripley, 2007), Support Vector Machine Classifier (Fan et al., 2008), LASSO and Ridge Classifier (Friedman et al., 2010), Random Forest Classifier (Breiman, 2001), Bagging Classifier (Breiman, 1996), Generalized Boosted Classifier (Ridgeway et al., 2006), AdaBoost Classifier (Freund et al., 1996) and (Hastie et al., 2009), kNN Classifier (Ripley, 2007) and Multinomial Naive Bayes Classifier (McCallum et al., 1998).

After fitting numerous classification models, the results indicated that no model is able to minimize false positives and false negatives in the same time. Some models are better at minimizing one over the other, but no models can do it both, and still be predict accurately. Therefore, based on these observations a solution was to design two stage classification system, where two different classifiers would be trained.

3.4.1 First Stage Classifier

The first classifier reduces the number of false negatives (predatory conversations labelled as non-predatory), minimizing Type II errors. This first stage classifier is trained to predicts a conversation as containing or not containing predatory behaviour. Using the first stage classifier, the predicted groups can be interpreted the following ways:

1. Conversations predicted by the first stage classifier as containing non-predatory behaviour: these are conversations that most likely do not contain predatory behaviour.

2. Conversations predicted by the first stage classifier as containing predatory behaviour: these are conversations that need to be looked at again, by a secondary classifier for filtering out false positives.

A comparison of different first stage classification models' performance can be found in the Results section, but it is worth noting that the Linear Discriminant Analysis Classifier was chosen as the first stage classifier, purely based on cross validated error rate and precision measurements.

3.4.2 Second Stage Classifier

The second stage classifier filters through all conversations labelled by the first stage classifier as containing possible predatory behaviour, reducing the number of false positives (non-predatory conversations labelled as predatory), minimizing Type I errors. Using the second stage classifier, the predicted groups can be interpreted the following ways:

1. Conversations predicted by the second stage classifier as containing non-predatory behaviour: these are conversations that possibly could contain predatory behaviour

2. Conversations predicted by the second stage classifier as containing predatory behaviour: these are conversations that most likely contain predatory behaviour

A comparison of different second stage classifier's performance can be found in the Results section, but it is worth noting that the AdaBoost Classifier (Freund et al., 1996) was chosen as the second stage classifier, purely based on cross validated error rate, precision and recall measurements.

Overall, the 2 stage classification system generates 3 groups, in increasing danger levels: conversations most likely not containing predatory behaviour, conversations possibly containing predatory behaviour and conversations most likely containing predatory behaviour. This resulted from 2 groups found by the first stage classifier, then the second stage classifier breaks down one of the first classifier's groups into 2 separate groups. This results in a hierarchical relationship between the groups.

3.5 Algorithm Overview

1. Parse each conversation to combine all lines in each conversation into one textual observation, pre-process the text for each conversation, and attach the labels to it

2. Take all pre-processed words used in the whole dataset of conversations and train a 400 dimensional Word2Vec model, which converts each unique word into a 400 dimensional vector.

3. For each conversation aggregate a min and a max feature vector by taking the min/max of each dimension from all words present in a specific conversation. This will result in a 400 dimensional min feature vector and a 400 dimensional max feature vector for each conversation.

4. To obtain an 800 dimensional conversation feature vector, for each conversation concatenate the min and max feature vectors.

5. Perform Linear Discriminant Analysis on the whole dataset of conversation feature vectors, to classify conversations as predatory vs. no-predatory behaviour (First stage classifier).

6. Take all conversations that have been labelled to contain predatory behaviour and feed them into an AdaBoost model to filter out conversations that contain non-predatory behaviour, but have been labeled as containing predatory behaviour (Second stage classifier).

7. Obtain 3 different groups of conversation, differentiated by their danger levels:

   - Group A = conversations most likely do not contain predatory behaviour: Linear Discriminant Analysis model predicted the conversation to contain non-predatory behaviour
   - Group B = conversations possibly could contain predatory behaviour: AdaBoost model predicted the conversation to contain non-predatory behaviour, but Linear Discriminant model predicted it to contain predatory behavior
   - Group C = conversations most likely contain predatory behaviour: AdaBoost model predicted the conversation to contain predatory behaviour

4 Case Study Results and Discussion

4.1 Sexual Predatory Conversation Identification Task

A controlled case study was conducted on a Sexual Predatory Conversation Identification task, similar to the Sexual Predator Identification tasks (Inches and Crestani, 2012). The data-set used for this controlled case study is the test set coming from Inches and Crestani's work. The test set was chosen over the training set, since the two set's ground truth labels are different, and only the test set's labels matched the controlled case study's needs.

This controlled case study differs from Inches and Crestani's Sexual Predator Identification tasks. Its aim is to focus on analyzing a whole conversation and deciding if it contains predatory behaviour, instead of identifying the predators among all users in different conversations or identifying the part of the conversations which are the most distinctive of predatory behaviour.

Inches and Crestani's test set provides an xml file, containing 155128 conversations, alongside author and line meta-data. A ground-truth label file is also given, containing conversations id's and lines id's of those lines considered suspicious (of a perverted behavior) in a particular conversation. Given this data, the conversation xml file was processed into a csv file, with columns containing conversation id, line number, author, text and each row representing a line within a conversation. To get the data in the format described at the beginning of section 3, the conversion of predatory line labels to conversation level labels was needed. In order to do so, each conversation's line level labels were checked, and a new label of 0 (non-predatory) or 1 (predatory) was assigned if any of the conversation's lines contained predatory behaviour.

After these initial data processing steps, the whole algorithm described in section 3 was executed. The sample size of the conversations dataset is 155128 observations, which is a large sample size. In order to speed up the computational process, a sample of 100000 conversations was taken. In this random sample 99450 conversations had the label 0, meaning no predatory behaviour has been identified within the conversation, and 550 conversations had the label 1, suggesting that predatory behaviour is present within the conversation. A seed number of 2017 was used in order to create reproducible results. The final results of this controlled case study were the classification results obtained from the 2 stage classification system. K-fold cross validation was used to assess the performance of various classification models on the feature vectors extracted from each conver-

sation. $k = 10$ was chosen for the number of folds in the cross validation process.

4.2 First Stage Classification Results

In section 3.4 a list of potential classification models were provided. In the first stage classification process LASSO, Linear Discriminant Analysis, Support Vector Machine, Random Forest, Bagging, Generalized boosted models have been tested out. The Random Forest model consistently outperformed the Bagging model, and since both of them as similar, the Bagging model was dropped.

The two groups within the data-set are massively unbalanced, thus measuring the error rate of classifiers would be a mistake. Instead, recall and precision measurement for the predatory behavior labels, and F-scores are taken into consideration (Sokolova and Lapalme, 2009). F-scores, recall and precision for predatory labels is calculated as per Table 2 in Sokolova and Lapalme's work. In statistical analysis of binary classification the F-score is a harmonic mean of precision and recall and it has a parameter, β. This parameter is just a constant, that places more emphasis on either the precision or recall. For the purposes of this binary classification tasks a β of value 1 was chosen, which makes the F-scores F1-scores, and it equally weights recall and precision, while punishing extreme recall or precision values. Table 1 contains the average precision and recall measurements from each fold within the the cross validation process, also the overall F1-score for each model. All results are color coded based on cross-comparisons done between each model's average recall, precision and F1 scores. All values are percentage rates, and lower values within the table are red, while higher values are represented with the green color cells within the table.

As previously established, the first stage classifier needs to reduce the number of false negatives (predatory conversations labelled as non-predatory), therefore the model with the highest average precision value is the best first stage classifier. Random Forest has a large recall average value of 0.8241, but its average precision is only 0.2982, thus it is not an appropriate first stage classifier. Its F1-score is under 0.5 which makes it an ineffective classifier for the overall classification task. Nonetheless, the Random Forest model is good at detecting non-predatory conversations

Classification Model	Average Recall	Average Precision	F1 Score
LDA	0.5223	0.9144	0.6648
SVM	0.7664	0.6585	0.7084
Random Forest	0.8241	0.2982	0.4379
LASSO	0.6739	0.6492	0.6613
Gr. Boosting Machine	0.8646	0.3018	0.4474

Table 1: Average Recall and Precision, and Overall F1-Score from the top 5 First Stage Classification Models.

labelled as predatory. The Support Vector Machine (SVM) classifier has a lower average precision value (0.6585) than its average recall value (0.7664), therefore it does not minimize the number of false negatives, and it can not be used as a first stage classifier. Its F1-score is 0.7084, which is acceptable, but using this SVM classifier will result in balancing out recall with precision, without focusing on minimizing either, therefore this model is not perfect for providing the best results. The LASSO model's performance is very similar to SVM's case, having an F1-score of 0.6613. LASSO has almost equal precision (0.6492) and recall (0.6739) values, having the same problem as the SVM model, not minimizing the number of false negatives, thus not being a good first stage classifier. Also worth noting that the Gradient Boosting Machine model's performance is really similar to the performance of the Random Forest model. Both models are good at detecting non-predatory conversations labelled as predatory, average recall being 0.8646, but they both have really poor average precision performance (0.3018), thus they are ineffective at detecting predatory conversations accurately. Finally, Linear Discriminant Analysis (LDA) minimizes the number of false negatives, having the largest average precision value at 0.9144, therefore it was chosen as the First Stage Classifier. The LDA classifier is very good at identifying conversations that contain predatory behavior, but its very low recall rate of 0.5223 shows that this model can not solve all the problems at once. The low recall rate created a concern over the large number of false positives, which triggered the addition of a second stage classifier. The second stage classifier should be focused on filtering out false positives, while not

creating a large number of false negatives. Figure 3 below shows the confusion matrix, after predicting for the left-out subset of observations during the cross validation process for the LDA model.

Cross Validated LDA

Figure 3: Confusion Matrix from the Cross Validated LDA model results.

As shown in Figure 3, the LDA model correctly classifies 503 out of 550 predatory conversation, out of a total of 100000 conversations. This model minimizes the number of undetected predatory conversations, which makes this model so effective, with a large average precision value. However if one looks at the F1-score, it is noticeably low, only 0.6648, which is understandable, since the average recall is only 0.5223. It can be concluded that the LDA model in only good at precision, and it is ineffective at detecting non-predatory conversations labelled as predatory, where it mis-classifies conversations about half of the times. Models like Random Forest and SVM are good at this this specific task, so a second stage classifier in addition to the LDA model would help.

4.3 Second Stage Classification Results

For the Second Stage Classification process the following classifiers have been trained: LASSO, Ridge Classifier, Naive Bayes Classifier, k-NN Classifier, Linear Discriminant Analysis, Support Vector Machine Classifier, Random Forest Classifier, Bagging Classifier and AdaBoost Classifier.

The second stage classifier's emphasis is on filtering through all conversations labelled by the

Classification Model	Average Recall	Average Precision	F1 Score
SVM	0.6934	0.8246	0.7533
Naive Bayes	0.5858	0.9285	0.7184
LASSO	0.7442	0.7711	0.7574
AdaBoost	0.7767	0.8091	0.7926
k-NN	0.6279	0.8966	0.7385

Table 2: Second stage classification process, with cross validated results from the top 5 models, and their average precision, recall measurements, alongside F1-scores.

first stage classifier (LDA) as containing predatory behaviour, reducing the number of false positives (non-predatory conversations labelled as predatory), thus minimizing Type I error. Since the second stage classifier only filters through flagged conversations, there is no need to classify all 100000 conversations, only those labelled by the LDA model as predatory (961 conversations, from Figure 3).

Table 2 contains the 5 best models built for the second stage classification process. The performance results from these 5 models are cross validated and average precision and recall measurements from each fold within the cross validation process are returned, alongside the overall F1-score for each model. All results are color coded based on cross-comparisons done between each model's average recall, precision and F1 scores.

The five best classifiers for the second stage classification process were SVM, Naive Bayes, LASSO, k-NN, and AdaBoost. Looking at the F1-scores, all 5 models have F1-values between 0.7 and 0.8, therefore all 5 models are competitive at filtering out the mistakes that the LDA model does in the first stage classification. The Naive Bayes and k-NN classifier's performance is quite similar, by having similarly large average precision values, 0.9285 for Naive Bayes, and 0.8966 for the k-NN model, but they both lack performance in average recall, where the values are quite low, 0.5858 and 0.6279 respectively. Overall, both have over 0.7 F1-scores, but the conclusion is that both Naive Bayes and k-NN models are effective at precision, but not at recall, and that leaves quite room for plenty of mis-classifications. A good second stage classifier would need to be really effective at both precision and recall, since if the chosen classifier would be optimized to minimize just one of pre-

cision or recall, a third stage classifier would be needed. Through observing the SVM model, it has a 0.7533 F1-score, which is larger than the previous 2 models, and even the average precision is over 0.8. but the average recall value is just shy of 0.7. Unfortunately the SVM model's performance has more emphasis on precision, therefore choosing such a model as the second stage classifier would result in having the same problem as with the Naive Bayes and k-NN classifier models. Looking at the LASSO model's performance, it has an F1-score of 0.7574, which is better than all previous second stage classifier models. Its average recall is 0.7442, and precision is 0.7711, which makes it an effective and recall and precision-wise balanced model. This LASSO model would be a good choice for the second stage classifier, but there is one problem: the AdaBoost model outperforms the LASSO model. AdaBoost has an average recall of 0.7767 and average precision of 0.8091, making its F1-score 0.7926, just shy of 0.80. This model performed the best at filtering out mis-classifications that the first stage classifier made, and it achieves equally large and balanced performance on both recall and precision, making the second stage classifier a complete system for accurate classification, without needing a third classifier. Figure 4 below shows the confusion matrix, after predicting for the left-out subset of observations during the cross validation process for the AdaBoost second stage classifier model.
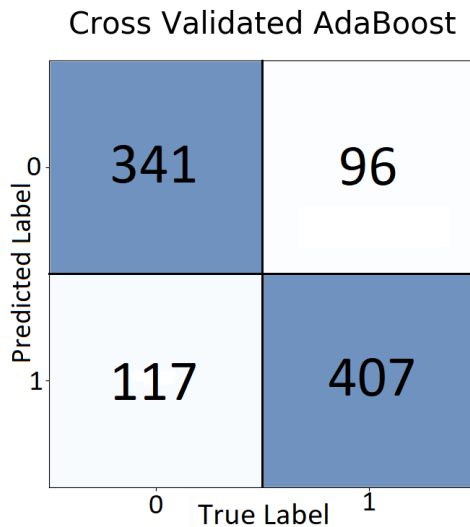

Cross Validated AdaBoost

Figure 4: AdaBoost Second stage Classification Result's Confusion Matrix

As shown in Figure 4, the AdaBoost model cor-

System of Classifiers	Recall	Precision	F1 Score
LDA → SVM	0.6928	0.7545	0.7224
LDA → Naive Bayes	0.5859	0.8491	0.6934
LDA → AdaBoost	0.7767	0.74	0.7579
LDA → LASSO	0.7433	0.7055	0.7239
LDA → k-NN	0.6273	0.82	0.7108

Table 3: Recall, precision and F1 scores from First Stage classifier combined with each possible second stage classifier.

rectly classifies 407 out of 503 predatory conversation. It also only mis-classifies 117 out of 458 non-predatory conversations. This AdaBoost model is as effective as it can be at minimizing errors made on both Type I and Type II errors, having large values for both average precision and recall. It can be concluded that the AdaBoost model is effective at filtering out mis-classifications made by the LDA first stage classifier. The combination of LDA as first stage classifier and AdaBoost as the second stage classifier answers one of the research questions asking "what kind of classification system and what statistical machine learning models can make a difference and predict whether or not a conversation contains sexual predatory behavior?".

4.4 System-wide Classification Results

In order to re-assure that LDA and AdaBoost are the best 2 models for the two stage classification system, Table 3 shows the Recall, Precision and F1-scores for LDA, the first stage classifier, combined with each possible second stage classifier. These values are obtained by fitting, then predicting using the first stage classifier, then using the second stage classifier to filter all observations that the first stage classifier flagged as potential predatory conversations.

The First stage classifier, LDA is the same model combined with different second stage classifiers. One can see how LDA → SVM, LDA → Naive Bayes and LDA → k-Nearest Neighbors has a lower recall value, but a much larger precision value. On the other hand, LDA → AdaBoost and LDA → LASSO have a much more balanced recall-precision ratio, since both of their values are closer to each other, meaning that both

AdaBoost and LASSO are good at avoiding both Type I and Type II errors. By observing the F1 scores, then LDA → AdaBoost outperforms every other combination of first stage-second stage classifier pairs, by having an F1-score of 0.7579. LDA → AdaBoost's largest F1-score is the main reason why the final 2 stage classification system is composed of the interaction of these 2 models. Figure 5 shows the overall classification system's confusion matrix, after both models have been applied, and their predictions have been merged.

Figure 5: Overall Classification system: LDA → Adaboost

It is worth noting that the results from Figure 5 are only useful for calculating the final recall, precision and F1-scores for the overall two stage classifier system. The algorithm's final output is a 3 group classification of conversations based on each group's uncertainty level on maliciousness, which can be seen in Table 4. The 3 groups can be interpreted the following ways:

- Group A: conversations most likely do not contain predatory behaviour

- Group B: conversations possibly could contain predatory behaviour

- Group C: conversations most likely contain predatory behaviour

Group A comes from the first stage classifier's upper half of the confusion matrix, where the Linear Discriminant model predicted the conversations to contain non-predatory behaviour.

Group	Non-predatory Conversations	Predatory Conversations
A	98992	47
B	341	96
C	117	407

Table 4: 3 group classification of conversations, based on uncertainty levels of maliciousness

Group B comes from the second stage classifier's upper half of the confusion matrix, where the Linear Discriminant model predicted the conversations to contain predatory behavior, but the AdaBoost model filtered the conversations by predicting them not to contain predatory behaviour.

Group C comes from the second stage classifier's lower half of the confusion matrix, where both the Linear Discriminant and AdaBoost model predicted the conversations to contain predatory behaviour.

Classifying conversations in 3 different groups based on the uncertainty levels of their maliciousness is a good idea, since it makes the output of the classification system more flexible. In the same time, Groups A, B and C show how the decision whether or not a conversation contains predatory behaviour is not always binary, since there could be some grey areas in the interpretation of the content of the conversation.

4.5 The Messy Nature of the Textual Data

One of the key aspects of the algorithm designed is taking into consideration the nature of data being analyzed. It is worth emphasizing that online chat-room conversations are filled with misspelled words, slang, internet acronyms, inappropriate language, broken grammar, short, messy and unstructured textual data. The most challenging aspect of this project is finding a way to interpret and analyze textual data. These linguistic challenges have been addressed in 3 different ways. Firstly, the careful selection of the adequate text cleaning techniques is considered. Secondly, special emphasis is put on the insight that lies within the contextual details of a conversation. Thirdly, a domain specific feature extraction technique is being used to extract the essential details from each conversation. These three key components of the algorithm answer the first research question, which asked "how to make modern computational language techniques and models adapt

to the difficult to interpret nature of online conversations?".

4.5.1 Advantages of Proper Text Cleaning

The text cleaning techniques used were whitespace, HTML tag, hyperlink and numeric character removal, lowercase conversion, autocorrect. Surprisingly, extensive white spaces, HTML tags and hyperlinks show up in Inches and Crestani's sexual predatory conversations dataset quite often, and since Word2Vec is not built to extract meaning from random HTML tags and hyperlinks, their removal from the conversations was obvious. When using the Word2Vec model, it is considered standard practice to remove numerical values, since their interpretation depends on domain, country of origin and other contextual details that often impossible to retain from the conversation. The lowercase conversion of all textual data is done for the sole purpose of not making discrepancies between uppercase and lowercase spellings of the same word. Conversations containing less than 3 words have been discarded, since no meaningful interpretation of such short conversation could be made. Arguably the most important text cleaning tool used is (McCallum, 2016)'s library called autocorrect, which is a spelling corrector. This tool is necessary for the as accurate as possible interpretation of broken language and misspelled words or expressions.

4.5.2 Insight in Contextual Details

When it comes to the Sexual Predator Identification task, previous approaches included the identification of predators among all users in different conversations or identification of parts of conversations which are most distinctive of the predator behaviour. In order to come up with a unique approach, the algorithm is centered around detecting insight that lies within the contextual details of a conversation. More specifically, with the help of vector representation of words, and customized feature extraction, the vector representation of a whole conversation is obtained. These "conversation feature vectors" are composed of each conversation's contextual details detected by the Word2Vec model, then carefully selected and aggregated by a feature extraction process. The "conversation feature vectors" obtained in such a manner are the essential input for classification models, which decide whether or not a conversation contains predatory behaviour. These con-

cepts answer the research question formulated as "how to extract semantic details from conversations, such that conversations containing malicious intent could be detected?". It is also worth noting that this approach considers the whole conversation as one large textual observation, without looking at discrepancies between each individual line in the conversation. To do so, the original labels from Inches and Crestani's dataset need to be parsed. The original labels are predatory line labels, which means that within each conversation, each line is labeled as predatory or non-predatory. The end goal of the label parsing process is to create conversation level labels, where each conversation contains a predatory/non-predatory label. This can be easily obtained by checking whether or not a predatory line is present within a conversation.

4.5.3 Domain Specific Feature Extraction

The feature extraction technique used in the algorithm is specific to only vector space representation of textual data. De Boom et al. stated that their technique was designed specifically for shorter text, that is messy, noisy and has a sparse vocabulary. They also noted that "traditional textual representations, such as tf-idf (Sparck Jones, 1972), have difficulty grasping the semantic meaning of such [messy and noisy] texts" (De Boom et al., 2016). De Boom et al.'s work suggests that a simple feature extraction technique that seems to work reasonably well is to compute the vector embedding for each word in the conversation, then aggregate them using the coordinate-wise mean, min, max, or a combination of them. It is worth noting not all of De Boom et al.'s work has been used in this project, only their work on coordinate-wise word embedding aggregation functions. The benefits of word embedding aggregation functions are enormous, since coordinate-wise min and max functions applied to each word within a conversation tend to detect extreme values within the vector embedding, thus detecting unusual, or extreme meaning within the conversation. Furthermore, one could speculate that these extreme semantic values within the conversations could mean extreme behaviour within conversations, such as malicious, predatory intent. This observation can not be proven, but incorporating the detection of extreme coordinate-wise values within vector embeddings is a proper way to extract features from word embeddings.

4.6 A Closer Look at the Vector Representation of Words Model Setup

In terms of knowledge representation, the benefits of the Word2Vec model towards the algorithm proposed are essential. Word2Vec was chosen to be part of the algorithm as the one and only vector representation of words model, because it "reconstructs the linguistic context of words and ...captures syntactic and semantic regularities in the language" (Mikolov et al., 2013d). Early in the research process it was not clear whether or not a pre-trained or a newly-trained Word2Vec model should be used. To address this concept, a couple of simple experiments have been conducted using a few famous pre-trained Word2Vec and Doc2Vec (Le and Mikolov, 2014) models. Doc2Vec is a similar model to Word2Vec, and it was used to evaluate whether or not the vector embeddings created by pre-trained models are useful enough, when applied to Inches and Crestani's sexual predatory conversations dataset. The pre-trained models include Google's 300 dimensional Word2Vec model (Mikolov et al., 2013b) and a couple of different pre-trained Doc2Vec models coming from (Lau and Baldwin, 2016)'s work, who experimented with document embedding generation. The experiments involved training a few classification models on the output of the above mentioned pre-trained models, and check if any correct conversation labels could be predicted from the vector embeddings. The pre-trained models could not provide any useful embeddings, since no classification model could make any accurate prediction based on the pre-trained vector embeddings. The simple and intuitive explanation for the above observation is that usually pre-trained models have been trained on specific data (e.g. all Wikipedia articles or news articles), but Inches and Crestani's sexual predatory conversations dataset does not necessarily match the same textual characteristics. In order to interpret the content of the conversation dataset, a Word2Vec model needs to be trained on the messy nature of the data described at the beginning of this section, therefore a self trained Word2Vec model was used during this research project, with the training data being Inches and Crestani's sexual predatory conversations dataset (Inches and Crestani, 2012).

Similar experiments have been conducted on the appropriate number of dimensions for the vectors in the Word2Vec model and the word embedding aggregation function used for feature extraction. The considered options were $dimensionSize = 200, 300, 400$ or 500, while the aggregation functions for feature extraction suggested by De Boom et al. are coordinate-wise minimum, maximum, mean, or a combination of them, like concatenation of a coordinate-wise minimum and maximum vector. Coordinate-wise mean would get rid of extreme values within the vectors, which is undesirable for detection of malicious textual content, thus it was not considered. Numerous experiments have been conducted with various combinations of model dimension size and feature extraction function. To be more precise, 3 to 5 classification models have been trained for each of the following pairs of vector aggregation function - Word2Vec model size: Min-200, Max-200, MinMax-200, Min-300, Max-300, MinMax-300, Min-400, Max-400, MinMax-400, Min-500, Max-500, MinMax-500. As previously mentioned in the algorithm section, the best results on these experimental model pairs have been achieved by the 400 dimensional model, with the concatenated coordinate-wise minimum and maximum feature vectors, or just shortly the MinMax-400 model.

4.7 A Closer Look at the Classification System

4.7.1 First Stage Classifier

By studying the first stage classifier, one would probably like to understand why a Linear Discriminant model does so much better than most other predictive models. Without going too much into depth about what a Linear Discriminant model is and how it works, one just needs to understand how does the model make decide if an observation x_i belongs to class $k = 0$ or $k = 1$, where x_i is an n dimensional vector, and class k is just a group within the data. Equation (1) (Tibshirani et al., 2013) shows the calculation of the discriminant function, $\hat{\delta}_k$, which decides to what group does an observation belong to. There are multiple parameters for the discriminant function: $\hat{\mu}_k$ is the mean of the group k, while $\hat{\sigma}_k$ is the variance of the group k, and $\hat{\pi}_k$ is the prior class membership probability of a group k.

$$\hat{\delta}_k(x_i) = x_i \cdot \frac{\hat{\mu}_k}{\hat{\sigma}^2} - \frac{\hat{\mu}_k^2}{2\sigma^2} + log(\hat{\pi}_k) \quad (1)$$

Firstly, LDA uses some estimation method to estimate the mean $\hat{\mu}_k$ and variance $\hat{\sigma}_k$ of a group

k, then it requires to know or estimate the prior class membership probability, $\hat{\pi}_k$. After the parameters have been estimated, the LDA classifier plugs the estimates for $\hat{\mu}_k$, $\hat{\sigma}_k$ and $\hat{\pi}_k$ into equation (1), and assigns an observation $X = x_i$ to the class for which $\hat{\delta}_k$ is the largest. One of the key parameters that can influence the discriminant function's decision is $\hat{\pi}_k$, the prior class membership probability of a group k. During training of the LDA model, the training data influences this parameter $\hat{\pi}_k$ by adjusting it to prior class weights of the training data, which is known, since the training data has labels. For the first stage classifier, the LDA model was trained on 100000 observations. Out of this training set 99450 observations belong to the non-predatory behaviour group ($k = 0$), while 550 observations belong to the predatory behavior group ($k = 1$), thus $\hat{\pi}_0 = 0.99450$, while $\hat{\pi}_1$ = 0.00550. These $\hat{\pi}_k$ values adjust the weighting on the probability that a new observation belongs to a class k, thus they had a massive influence on the performance and accuracy of the LDA model.

As LDA became First Stage Classifier, its performance was very good at identifying conversations that contain predatory behavior, but it was creating a concernedly large percentage of false positives, which in the end triggered the fitting of a second stage classifier. About half of the conversations flagged as containing predatory behaviour actually were non-predatory conversations being classified as predatory conversations. This means that too many "innocent conversations were mislabeled, which affected the classification system's overall trustworthiness. The classification system needed a filtering model, which focused on lowering the overall amount of false positives, and it was called the second stage classifier.

4.7.2 Second Stage Classifier

In the second stage classification AdaBoost performs well, better than similar tree based methods like Random Forest or Bagging, since its algorithm is a similar, but improved version of Bagging. Boosting works in a sequential manner, as each tree within the AdaBoost model is fitted on random subsets of the original training set, without the use of Bootstrapping, then finally the iterative models are added up to create a strong classifier. Furthermore, since (Freund et al., 1996) and (Hastie et al., 2009)'s AdaBoost algorithm includes iterative feedback to the sequence of models fitted using the weights of each observation being updated by the error rate, this provides an extra accuracy to the AdaBoost model fitted as the second stage classifier.

The LASSO model's performance is accurate as both a first and second stage classifier. It can be noted that the LASSO model's performance ranked in the top 5 most accurate models for both first and second stage classification process. Unfortunately, both times it was outperformed by other models, but nonetheless it is a great model, which is good minimizing both types of errors to some level of accuracy. With F1-scores of 0.6613 and 0.7574 for the first and respectively second stage classifier, one must wonder why such a model is worth taking into consideration for classification tasks. The explanation is simple: LASSO is supervised machine learning method, which is also known as a shrinkage and variable selection method for linear regression models. In this project LASSO is applied on the "conversation feature vectors", which are 800 dimensional. One could only speculate which of those 800 dimensions are useful predictors for the Sexual Predatory Conversation Identification task, but the LASSO model can actually do so. By carefully choosing λ, the model's penalty term, the model applies constraints on the original coefficients, which end up shrinking the coefficients of useless predictors to zero. Penalizing the useless predictors results in selecting the most important variables associated with the response variable. The advantage of using LASSO for this classification task is great variable selection, which provides greater prediction accuracy and better model interpretability.

4.8 Model Trade-off and Dimensionality Reduction

A few observations worth mentioning are the trade-offs of using different models for second stage classification. A decision needs to be made on what to prioritize more: filtering out more "non-predatory" conversations from type I errors or putting back more "predatory" conversations into type II errors. For the sake of the argument, let's assume that the two stage classification system gets implemented by a company who process an online product which contains online chat-room. The moderators of the chat-room could choose to adjust the model used for the second stage classifier, based on their priority of mini-

mizing either type I or type II errors. Using the SVM, Naive Bayes or k-NN classifier in the second stage would result in having less type II errors over more type I errors. On the other hand, using either the LASSO or AdaBoost model in the second stage balances out the first stage classifier's large number of type I errors, having a more less equal amount of type I and type II errors overall.

Lastly, there was an attempt on applying dimensionality reduction algorithms like Classical Multidimensional Scaling (Borg and Groenen, 2005), Principle Component Analysis and Factor Analysis (Jolliffe, 1986) on the "conversation feature vectors" to reduce some of the 800 dimensions. The goal would have been to apply the classification system on the reduced feature vectors, but no real benefit was observed, since all classification models performed significantly worse after performing dimensionality reduction.

5 Future Works

There are 3 main areas, where improvements could be made in the future: the representation of words in vector space, the feature extractor and the classification system.

Firstly, Facebook's efficient learning of word representations and sentence classifications, Fast-Text (Joulin et al., 2016) could be used to improve the quality of the word vector representation. Alternatively other vector space models like Doc2Vec (Le and Mikolov, 2014), Lda2Vec(Moody, 2016), ConceptNet Numberbatch (Speer et al., 2017) or Sense2Vec (Trask et al., 2015) could be experimented with. Secondly, the entire version of De Boom's algorithm (De Boom et al., 2016) could be used to create a better feature extractor by applying representation learning concepts combined with weighted word embedding aggregation. Thirdly, a deep learning approach could be considered for the classification system, by favouring prediction accuracy over model interpretability.

6 Conclusion

This research project was proven to be the perfect combination of computational linguistics and statistical machine learning in order to find insight in the data. The main goal of the project was to design an approach that could detect and classify online conversations as either containing or not containing sexual predatory behaviour within the conversation. Throughout this project 100000 messy and unstructured online chat-room conversations were looked at, with the hopes of detecting a few needles in the haystack, 550 malicious conversations. With the help of deep learning language models, innovative feature extraction techniques, variety of statistical machine learning models, and exploration of contextual details a new, innovative and unique approach was born. The proposed algorithm uses models like Word2Vec, Linear Discriminant Analysis and AdaBoost to detect potential predatory behaviour with accuracy measured as a F1-score of 0.7579. The algorithm's two stage classification system creates a 3 group classification of conversations based on their uncertainty levels of maliciousness. One of the advantages of such classification system is that the second stage classifier could be easily switched to a different model to customize the system's priorities by minimizing either type I or type II errors. In the end, the algorithm aims to enhance children's safety in online environments by detecting malicious behaviour in online conversations.

References

Bijalwan, V., Kumar, V., Kumari, P., and Pascual, J. (2014). Knn based machine learning approach for text and document mining. *International Journal of Database Theory and Application*, 7(1):61–70.

Bogdanova, D., Rosso, P., and Solorio, T. (2012). On the impact of sentiment and emotion based features in detecting online sexual predators. In *Proceedings of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis*, pages 110–118. Association for Computational Linguistics.

Borg, I. and Groenen, P. J. (2005). *Modern multidimensional scaling: Theory and applications*. Springer Science & Business Media.

Breiman, L. (1996). Bagging predictors. *Machine learning*, 24(2):123–140.

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

Brooke, J. and Hirst, G. (2012). Paragraph clustering for intrinsic plagiarism detection using a stylistic vector space model with extrinsic features. In *CLEF (Online Working Notes/Labs/Workshop)*.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3):273–297.

De Boom, C., Van Canneyt, S., Demeester, T., and Dhoedt, B. (2016). Representation learning for very short texts using weighted word embedding aggregation. *Pattern Recognition Letters*, 80:150–156.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391.

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., and Lin, C.-J. (2008). Liblinear: A library for large linear classification. *Journal of machine learning research*, 9(Aug):1871–1874.

Freund, Y., Schapire, R. E., et al. (1996). Experiments with a new boosting algorithm. In *Icml*, volume 96, pages 148–156. Bari, Italy.

Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1):1.

Fuller, C. M., Biros, D. P., and Delen, D. (2011). An investigation of data and text mining methods for real world deception detection. *Expert Systems with Applications*, 38(7):8392–8398.

Hastie, T., Rosset, S., Zhu, J., and Zou, H. (2009). Multi-class adaboost. *Statistics and its Interface*, 2(3):349–360.

Inches, G. and Crestani, F. (2012). Overview of the international sexual predator identification competition at pan-2012. In *CLEF (Online working notes/labs/workshop)*, volume 30.

Jolliffe, I. T. (1986). Principal component analysis and factor analysis. In *Principal component analysis*, pages 115–128. Springer.

Joulin, A., Grave, E., Bojanowski, P., and Mikolov, T. (2016). Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Kang, I.-S., Kim, C.-K., Kang, S.-J., and Na, S.-H. (2012). Ir-based k-nearest neighbor approach for identifying abnormal chat users. In *CLEF (Online Working Notes/Labs/Workshop)*. Citeseer.

L, K. (2015). Graphic representations of word2vec and doc2vec.

Lau, J. H. and Baldwin, T. (2016). An empirical evaluation of doc2vec with practical insights into document embedding generation. *arXiv preprint arXiv:1607.05368*.

Le, Q. and Mikolov, T. (2014). Distributed representations of sentences and documents. In *International Conference on Machine Learning*, pages 1188–1196.

Lilleberg, J., Zhu, Y., and Zhang, Y. (2015). Support vector machines and word2vec for text classification with semantic features. In *Cognitive Informatics & Cognitive Computing (ICCI* CC), 2015 IEEE 14th International Conference on*, pages 136–140. IEEE.

McCallum, A., Nigam, K., et al. (1998). A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, volume 752, pages 41–48. Citeseer.

McCallum, J. (2016). Spelling corrector python 3 library.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013b). Googlenews 300 dimensional word vectors. Statistical Model 12. Retrieved October 6, 2017.

Mikolov, T., Sutskever, I., and Le, Q. (2013c). Tool for computing continuous distributed representations of words. Google open source blog, Google Knowledge.

Mikolov, T., Yih, W.-t., and Zweig, G. (2013d). Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751.

Moody, C. E. (2016). Mixing dirichlet topic models and word embeddings to make lda2vec. *arXiv preprint arXiv:1605.02019*.

Morris, C. and Hirst, G. (2012). Identifying sexual predators by svm classification with lexical and behavioral features. In *CLEF (Online Working Notes/Labs/Workshop)*, volume 12, page 29.

Munezero, M., Mozgovoy, M., Kakkonen, T., Klyuev, V., and Sutinen, E. (2013). Antisocial behavior corpus for harmful language detection. In *Computer Science and Information Systems (FedCSIS), 2013 Federated Conference on*, pages 261–265. IEEE.

Nalini, K. and Sheela, L. J. (2015). Classification of tweets using text classifier to detect cyber bullying. In *Emerging ICT for Bridging the Future-Proceedings of the 49th Annual Convention of the Computer Society of India CSI Volume 2*, pages 637–645. Springer.

Nandhini, B. S. and Sheeba, J. (2015). Online social network bullying detection using intelligence techniques. *Procedia Computer Science*, 45:485–492.

Nath, S. V. (2006). Crime pattern detection using data mining. In *Web intelligence and intelligent agent technology workshops, 2006. wi-iat 2006 workshops. 2006 ieee/wic/acm international conference on*, pages 41–44. IEEE.

NSS (2017). An intuitive understanding of word embeddings: From count vectors to word2vec. Retrieved October 6.

Řehůřek, R. and Sojka, P. (2010). Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. ELRA.

Ridgeway, G. et al. (2006). gbm: Generalized boosted regression models. *R package version*, 1(3):55.

Ripley, B. D. (2007). *Pattern recognition and neural networks*. Cambridge university press.

Selvi, S. T., Karthikeyan, P., Vincent, A., Abinaya, V., Neeraja, G., and Deepika, R. (2017). Text categorization using rocchio algorithm and random forest algorithm. In *Advanced Computing (ICoAC), 2016 Eighth International Conference on*, pages 7–12. IEEE.

Shafiabady, N., Lee, L. H., Rajkumar, R., Kallimani, V., Akram, N. A., and Isa, D. (2016). Using unsupervised clustering approach to train the support vector machine for text classification. *Neurocomputing*, 211:4–10.

Sokolova, M. and Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4):427–437.

Sparck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21.

Speer, R., Chin, J., and Havasi, C. (2017). Conceptnet 5.5: An open multilingual graph of general knowledge. In *AAAI*, pages 4444–4451.

Tibshirani, R., James, G., Witten, D., and Hastie, T. (2013). An introduction to statistical learning-with applications in r.

Trask, A., Michalak, P., and Liu, J. (2015). sense2vec-a fast and accurate method for word sense disambiguation in neural word embeddings. *arXiv preprint arXiv:1511.06388*.

Vilariño, D., Castillo, E., Pinto, D., Olmos, I., and León, S. (2012). Information retrieval and classification based approaches for the sexual predator identification.

Villatoro-Tello, E., Juárez-González, A., Escalante, H. J., Montes-y Gómez, M., and Pineda, L. V. (2012). A two-step approach for effective detection of misbehaving users in chats. In *CLEF (Online Working Notes/Labs/Workshop)*.

Wang, X., Zhang, C., Ji, Y., Sun, L., Wu, L., and Bao, Z. (2013). A depression detection model based on sentiment analysis in micro-blog social network. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 201–213. Springer.

Wolak, J., Finkelhor, D., and Mitchell, K. (2004). Internet-initiated sex crimes against minors: Implications for prevention based on findings from a national study. *Journal of Adolescent Health*, 35(5):424–e11.

Wolak, J., Finkelhor, D., Mitchell, K. J., and Ybarra, M. L. (2008). Online" predators" and their victims: myths, realities, and implications for prevention and treatment. *American psychologist*, 63(2):111.