Identifying Cyber Predators by Using Sentiment Analysis and Recurrent Neural Networks

Dan Liu

A Thesis
In The Department
of
Computer Science and Software Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of
Master of Science (Computer Science) at
Concordia University
Montréal, Québec, Canada

April 2018

Concordia University

School of Graduate Studies

This is to certify that the thesis prepared

By: **Dan Liu**

Entitled: **Identifying Cyber Predators by Using Sentiment Analysis and Recurrent Neural Networks**

and submitted in partial fulfillment of the requirements for the degree of

Master of Science (Computer Science)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the Final Examining Committee:

_____ Chair
Dr. Abbas Javadtalab

_____ Examiner
Dr. Leila Kosseim

_____ Examiner
Dr. Tiberiu Popa

_____ Supervisor
Dr. Olga Ormandjieva

_____ Co-supervisor
Dr. Ching Y. Suen

Approved by _____
Chair of Department or Graduate Program Director

_____ 2018

_____
Amir Asif, Dean
Faculty of Engineering and Computer Science

Abstract

Identifying Cyber Predators by Using Sentiment Analysis and Recurrent Neural Networks

Dan Liu

Recurrent Neural Network with Long Short-Term Memory cells (LSTM-RNN) have impressive ability in sequence data processing, particularly language model building and text classification. This research proposes the combination of sentiment analysis, sentence vectors, and LSTM-RNN as a novel way for cyber Sexual Predator Identification (SPI). There are two tasks in SPI. The first one is identifying sexual predators among chats. The second one is highlighting specific sexual predators' lines in chats. Our research focuses on the first task.

An LSTM-RNN language model is applied to generate sentence vectors which are the last hidden states in the language model. Sentence vectors are fed into the LSTM-RNN classifier, so as to capture suspicious conversations. Hidden state makes a breakthrough in the generation of unseen sentence vectors i.e., the system can score a sentence never seen before in the training data. Fasttext is used to filter the contents of conversations and generate a sentiment score to the purpose of identifying potential predators. IMDB sentiment review task is introduced to provide an intuitive measurement of the combined method. The model identified 206 predators out of 254. The experiment achieved a record-breaking F-0.5 score of 0.9555, higher than the top-ranked result in the SPI competition.

Acknowledgments

I would like to thank my supervisors, Dr. Ching Yee Suen and Dr. Olga Ormandjieva, for their guidance and encouragement. I admire their academic achievements and am truly grateful for their valuable comments on this thesis. I would like to thank fellow researchers in the academia and industry for their contribution and enlightenment. My appreciation also goes to the Natural Sciences and Engineering Research Council of Canada, for supporting this project.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

The greater popularity of social networks gives rise to cyber-criminal activities conducted by sexual predators. On Wikipedia, those who commit sex crimes, such as rape or child sexual abuse, are commonly referred to as "sexual predators". During 2000 to 2006, there was about 381% (Figure 1.1) increase in arrests of cyber sexual predators who accosted undercover investigators servicing as a prostitute in the US [1]. Moreover, according FBI's data [2], there were 750,000 child predators online in 2009. In this context, PAN (Plagiarism analysis, Authorship identification, and Near-duplicate detection) lab initiated the Sexual Predator Identification (SPI) Task in 2012 [3]. The PAN lab collects and shares an overwhelming amount of online chats, inside which there are predators, to facilitate research in predator behaviors. The dataset includes only a few conversations that are initiated by sexual predators, many conversations where people talking about sex and general topics. There are 142 predators for training and the purpose is finding 254 predators in about 200,000 online chats (Figure 1.2).

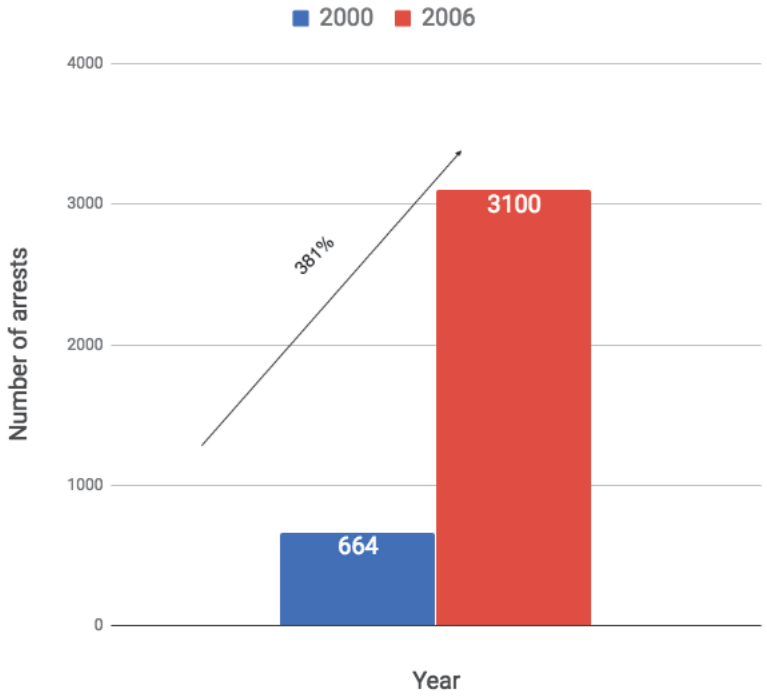

Figure 1.1: Online predator arrests from 2000 to 2006 [1].

	PJ perverted-justice.com	krjin krijnhoetmer.nl/irc-logs	irclog irclog.org	omegle omegle.inportb.com
#conversations	11350	50510	28501	267261
#conv. length ≤150	9076	48569	21896	265747
(% all)	(80%)	(96%)	(77%)	(99%)
Training set				
#conv. length≤150	2723	14571	6569	43064
” and exactly 2 user	984	2420	1146	41067
(% training)	(36%)	(17%)	(17%)	(95%)
unique (perverted) users	291 (142)	2660	10613	84131
Testing set				
#conv. length≤150	5321	33998	15327	100482
” and exactly 2 user	1887	5648	2673	95648
(% testing)	(35%)	(17%)	(17%)	(95%)
unique (perverted) users	440 (254)	4358	17788	196130

Figure 1.2: Properties of the PAN-2012 dataset [3].

1.1 Problem Definition

There are two separate tasks in SPI, namely, identification of sexual predators among chats and highlighting specific sexual predators' lines in chats. The research reported in this thesis focuses on the first task. As [3] indicated, the first step is to find out which conversations are suspicious, then identify which conversations belong to which author. [4], [5], and [6] use the similar method as [3] to identify the predators.

The organizers set the goal for the SPI task as creating a large and realistic dataset. The side effects of realistic data are high noise level, unbalanced training samples, and various lengths of conversations. More specifically, there are many general and sex-related conversations, while among them only a few involve sexual predators. Furthermore, there are many chat abbreviations and cyber slangs in conversations, such as "ur" for "your", "yr" for "year", "sorryyyy" for "sorry", to name a few. Such expressions are crucial to and should be considered in feature selection. Therefore, traditional machine learning methods cannot achieve satisfying performance unless truncating data with numerous rules. Even if n-gram is used, with hundreds of thousands of conversations, the noise will generate extreme sparsity, and the performance will be weakened consequently [7].

However, very complex and specific rules were applied to remove noise or to extract features. Especially, in [4], only about 10% of the samples remained for training and testing. Such removal could influence the generalization ability of the classifier. Manual rules for features extraction in [8]–[10] will reduce the performance because only samples that match the rules can be classified. A neural network language model approach can overcome the above-mentioned problems.

The second classifier is about predator identification. Support Vector Machines (SVM) [5], [8], Naive Bayes [11], [12] and other classical machine learning approaches [5], [7], [9], [13] were introduced. According to the official rank [3], those approaches with greater-than-90% precision had a lower recall (fewer than 80%).

1.2 Hypothesis and Research Approach

1.2.1 Hypothesis

The hypothesis of our work is:

- Predators always want to launch attacks by using the same pattern and always ask questions with attacking intention. In this way, the sentiment score of predator must be higher than the victim's.

1.2.2 Research Approach

A common strategy for SPI is the use of two classifiers. The first classifier will detect suspicious conversations which can be seen as positive (with predators) or negative (without predators) [4]–[6]. Long short-term memory recurrent neural networks [14] (LSTM-RNN) sentence vectors are introduced to solve the above-mentioned noise and performance problems. Different from n-grams, sentence vector can capture sentence features more efficiently and compress the size of the input data, as the classifier will only take sentences, instead of words, as features. Meanwhile, LSTM-RNN classifier can also be used for suspicious conversation detection (SCD) since it is good at learning long-term dependencies in time series data.

The work involves three types of neural networks. The LSTM-RNN-based language model, which is used to express the relation inside a sentence. The last hidden state of the LSTM-RNN of each sentence will be used as sentence vectors. A two-layer LSTM-RNN classifier, which is used to find suspicious conversations by learning the dependencies among sentences in a conversation. Each sentence in a conversation will be regarded as a single input and fed into the classifier. Following the detection of suspicious conversations, a Fasttext-based sentiment score model is introduced to identify sexual predators.

1.2.3 Overview of System Architecture

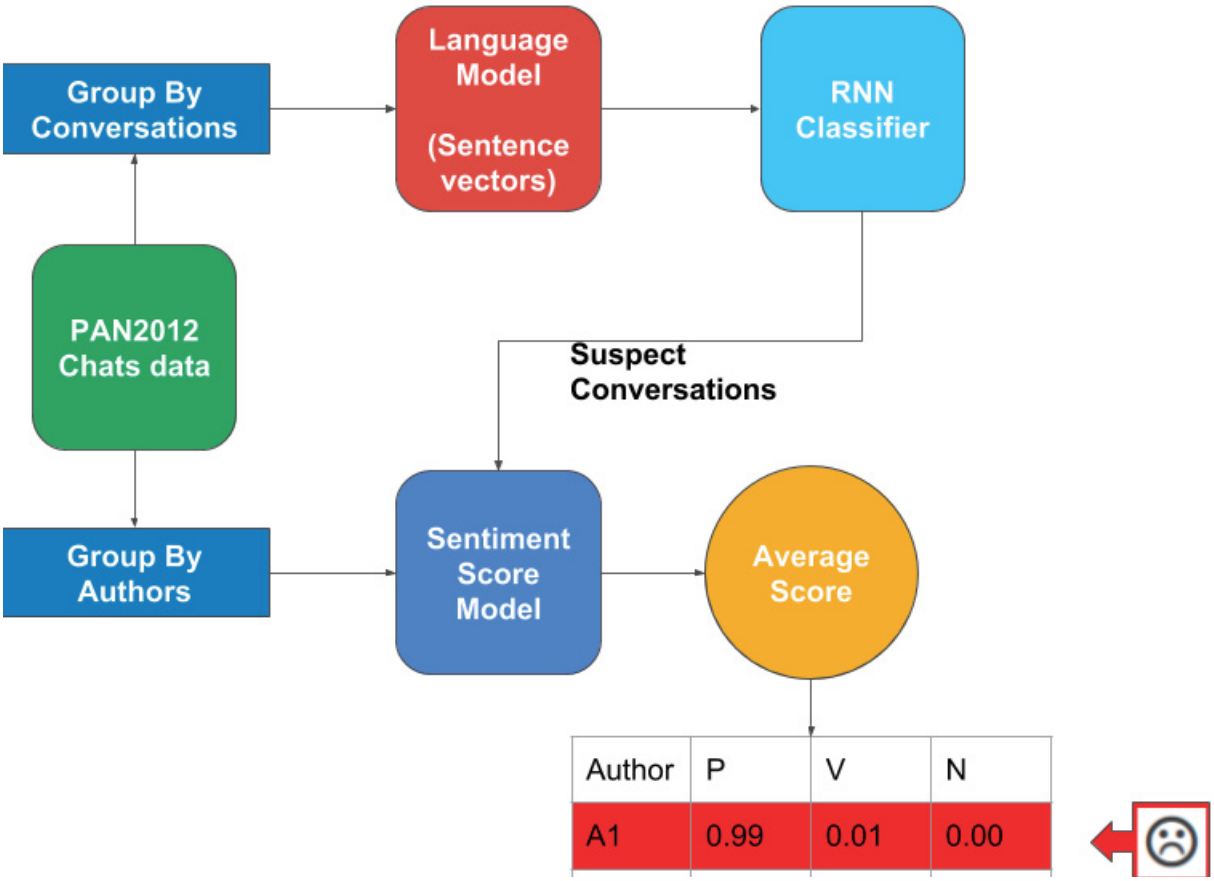

Figure 1.3: System structure of identifying predators.

The system includes two parts i.e., the conversation classification part and predator identification part (Figure 1.3).

Firstly, the LSTM-RNN-based language model, which is used to express the relation inside a sentence, is proposed. The training purpose of this step is to minimize the perplexity of the model. In information theory, perplexity measures the prediction ability of the model. A lower perplexity indicates stronger prediction capabilities. RNN learns knowledge based on the weights obtaining from previous inputs, which classical neural networks are unable to do. It contains loops and can store information in the form of weights and states. However, one of the shortcomings of an RNN is the limited capacity of handling long-term

dependencies, which is very common in real life, for example, learning the dependencies in very long sentences. In this context, LSTM-RNN, first introduced by Hochreiter and Schmidhuber [14], has become popular in recent years. It designs a gate strategy to regulate the cell states by controlling the weights passing through. In this way, LSTM-RNN can learn long-term dependencies from training samples, which is very useful in context-based datasets. An LSTM-RNN language model has two classical applications. The first one is learning the distribution of probabilities of words within training samples. The second one is learning the word representations (or word embeddings). Word embeddings showed its power in many natural language processing applications in recent years. In our work, the last hidden state of LSTM-RNN language model of each sentence will be used as sentence vectors. With a number of sentence vectors, the conversation can be presented in a highly compressed way. This is the key method for training a classifier in large dataset without much performance loss. The model has the advantage of generating new vector for the sentence that has never seen before. However, its disadvantage is that there must be a dictionary to store the mapping between a sentence and its vector.

Secondly, a two-layer LSTM-RNN classifier is used to find suspicious conversations by learning the dependencies among sentences in a conversation. As the conversation has been compressed via the approach above, the training speed of LSTM-RNN classifier is phenomenal. A typical predator conversation, depending on different attacking stages [4], usually includes age information, parents' information, wearing information, and images information of victim. The nature of the LSTM-RNN determines the ability of finding those relations among different stages, as those stages are time-sequence-based. In this way, the sentences sharing similar keywords of a stage-specific topic in predator conversations will be captured by the LSTM-RNN classifier. The sentence vector is a dense representation which compresses the relations in the format of weights into a single vector. It can be used to obtain the similarity of sentences, by calculating the cosine distance of the sentence vector. Each sentence in a conversation will be regarded as a single input and fed into the LSTM-RNN classifier. There are two substructures involved in the training procedure. The first one learns and expresses the dependencies among words in sentences. The second one learns features among the sentences in conversions. Each substructure is fulfilled by a

LSTM-RNN model. After passing through two LSTM-RNN models, the information in the conversation is compressed significantly. Because of the compression, the training speed is also increased.

Lastly, following the detection of suspicious conversations, the conversations are regrouped by participants. For regular users, regrouping will not change the sentiment score significantly. The same participant in different contexts will generate completely different types of sentence and topics of conversion. The sentiment features will also be blurred. On the contrary, according to our sample analysis, sexual predators usually tend to launch attack on different victims with similar patterns i.e., asking for privacy information directly. By regrouping the conversations of participants, the context information will be weakened and broken. However the patterns of predators will be exposed under the spotlight of random topics generated by regular users. As the transformation will split conversations into small parts, while there are massive irrelevant short sentences involved in the training of classifier, the patterns of sexual predators become obvious. In this situation, a very shallow and fast-training neural network is needed to do the scoring work. A Fasttext-based sentiment score model is introduced to identify sexual predators by scoring the authors. The output value of Fasttext is taken as score. As the conversations are split into different groups by author, the sentiment scores in different conversation should be averaged. Our results indicate that, even though the victims are sometimes assigned a very positive score, the sexual predators always get a much higher score. This is because the previous LSTM-RNN classifier has already detected the suspicious conversations. In this way, the sexual predators and victims can be recognized very precisely via sentiment score.

1.3 Contributions

The experiment achieved an F-0.5 score of 0.9555 on SPI. Finally, 206 out of 254 predators were identified by the intersection of two classifiers with zero error, which exceeded the best result [4] of the official ranking (203 out of 254 with 3 misclassifications). The contributions of this thesis are three-fold (Figure 1.4), namely:

- LSTM-RNN is introduced to generate sentence vectors especially for sentences never seen before (The words of the sentences must be in the training vocabulary).
- Internet Movie Database (IMDB) sentiment analysis dataset [15] is used to test the performance of sentence vectors model.
- Sentiment score is introduced to improve the performance of sexual predators identification.

Aspect	Highlights
PAN2012	1. The experiment achieves a record-breaking accuracy which higher than the top-ranked result. 2. Applying three different type of neural networks together to get the best result.
IMDB	1. IMDB sentiment analysis via sentence vectors. 2. Comparing the performance of sentence vectors with other methods.
Neural Network and Language Model	1. Using LSTM-RNN Language Model to generate sentence vectors. 2. Applying sentence vectors as features to identify suspect chats.
Sentiment Analysis	Using FastText as sentiment score model to measure the attacking intention.

Figure 1.4: Highlights of our work.

1.4 Overview of The Thesis

The related work and different approaches are summarized in the Literature Review chapter. In this section, the highlight of their performance, classifiers, and processing rules are compared. The analysis of advantages and disadvantages of their methodology are listed.

Our approach is explained in the Methodology chapter. It describes the details and background of data processing, neural network model structure, and sentiment score model. It also proposes the novel procedure of sexual predators detection and how each component function. The detail of the sentence vectors - the key feature to accelerate the training procedure - is introduced.

The experimental work conducted in this research is described in the Experimental chapter. It includes dataset description, the performance and comparison of each classifier and the configuration of the neural network. The IMDB sentiment analysis work is introduced to measure the novel method we applied in SPI competition.

The experimental results of this research is described in the Results chapter. The results of training language model, SCD classifier and sentiment score model are detailed in this section. It also compares the performance of the sentence vector in both IMDB and SPI work.

Finally, the conclusion chapter summarizes the thesis and proposes potential research pathways in the future. The literature review of our work is in the next chapter.

Chapter 2

Literature Review

2.1 Pros and Cons of Common Methodologies

Most of the competitors at the PAN-2012 took common strategy to use two classifiers for suspicious predator identification. At first, the classifier will be trained to classify suspicious conversations to see if there is a predator involved [4]–[6]. Then, another classifier for predator identification will be applied based on previous classification results. Our work also involves part of this combined approach. Different from our work, other researchers usually design a group of complex and specific strategies to remove noise or to extract features. The introduction of manual rules will simplify the problem, reduce the total training samples to accelerate the learning speed. Unlike deep learning, sort of black box, rule-based feature extraction methods will classify samples in a human-understandable way. On the contrary, this kind of method applies too many human opinions, i.e., high level dimension of features, into classification work. The generalization ability of the classifier may be limited if too many human summarized features are included.

As most researchers in SPI used traditional machine learning method, the rule-based pre-processing methods are quite useful. With the introduction of n-grams, the side effect is the large number of meaningless features, i.e., noises. By using the TF-IDF weighting strategy, the noises within the samples are reduced.

However, manual rules for features extraction in [8]–[10] may reduce the generalization ability because only the sample that matches the rules can be classified. Neural network language model approach can solve these problems. However, even though the neural network was applied in [4], only about 10% of the samples remained for training and testing. Such removal could influence the generalization ability of the classifier. The truncation rules bring many unclassifiable samples i.e., 140,000 in 150,000. Therefore, the advantages of neural network are restricted. In other words, that model cannot classify the samples excluded by rules.

2.2 Lexical Features

Customized corpora, for example, special terms or n-gram are used. Researchers built the features by assigning short number an age label "young", "adult", or "old" [5] and try to specify the gender of the conversation's participant based on rules. The n-gram and Maximum-Entropy is popular feature extraction methods as well. Specifically, 5-gram was used in classification work [7] that requires huge computation resources and a long period for training and classifying. For the lexical feature of n-gram which n is fewer than three in [4], [5], [8], [9], as there is a large number of cyber slangs, the training corpus is likely to be submerged by irrelevant noise.

2.3 Behavioural Features

Another category of features is behavioural features. Behavioural features are the actions of a participant of a conversation. The researcher in [13] referred to 51 hand-coded patterns to describe the conversation. For example, questions regarding the family of victims, privacy questions, and meeting requests. Some researchers summarized sexual predators actions systematically [4], [8], [9]. Those types of features are usually based on the result of training data analysis, e.g., length of conversations, psychological features, and attacking stages. The psychological extraction method categorizes words in general that reflect the underlying

psychology characteristics. For the attacking stage features, researchers categorize common attacking procedure of predators and take the stages as training materials to train the language model.

2.4 Neural Network Language Model

Neural network language model was introduced in [16]. After that a series of derived version [17]–[19], i.e., word embeddings, sentence embeddings etc. is widely applied in Natural Language Processing (NLP) work. It shows very strong ability in NLP tasks in terms of sentence embedding or document embedding. Because it is hard for those language models to represent sentences never seen before, the new sentence which is not in the training dataset will be a problem. Unlike various of embeddings, [19] proposed a more compatible method with sequence to sequence model to vectorize sentences based on the preceding and next sentences.

2.5 Sentiment Score

There are two main types of sentiment analysis tasks which are online service rating [20] and movie review [15]. In the entire IMDB sentiment analysis dataset, movies with fewer than 30 reviews or with neutral ratings are not included. A negative review has a score lower than 4 out of 10, and a positive review has a score higher than 7 out of 10.

For the score model, Severyn and Moschitti [21] used the distance to the margin of SVM as sentiment score, the larger the distance is, the more positive or negative it will be (Figure 2.1). Another popular score model takes the output of the neural network as score, i.e., 0 means very negative and 1 means very positive (Figure 2.2). Deep neural network is also a popular area for sentiment analysis. Hong and Fang [22] compared the performance of different neural network models and traditional machine learning methods on IMDB sentiment datasets, such as convolutional neural network (CNN), naive Bayes

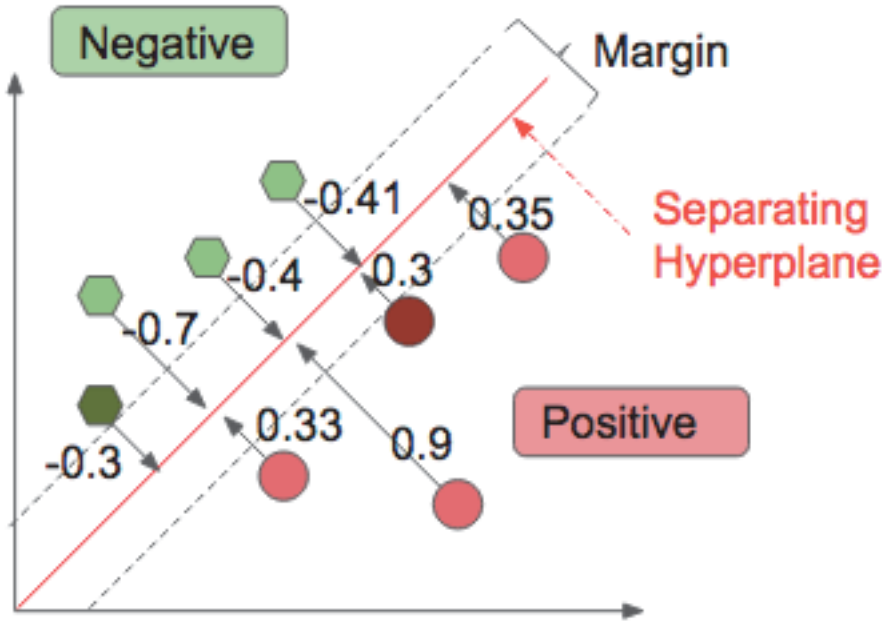

Figure 2.1: SVM as sentiment score model.

SVM (NBSVM) [23], and LSTM-RNN.

Although very deep neural networks have strong capability on NLP tasks, the training cost cannot be neglected. Fasttext [24] is a neural network with structure which is good at processing sentiment analysis tasks. It combines bag of words, word-embedding, and average pooling with fully-connected layers. This very simple structure brings impressive accuracy and speed (Figure 2.3, Figure 2.4). With such advantages, it can be sentiment score model for SPI work. In the next chapter, the methodologies used in our work are introduced.

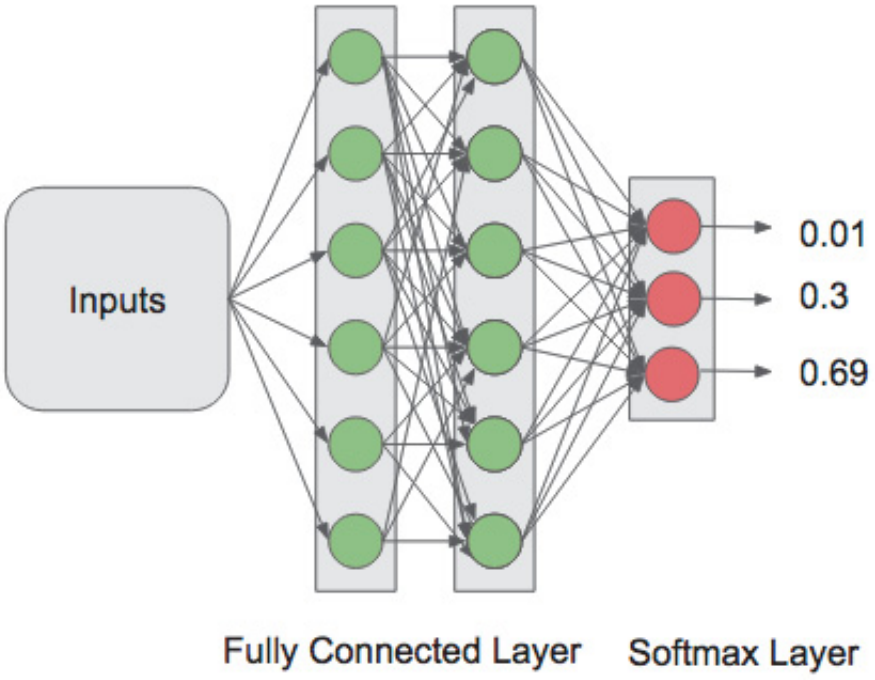

Figure 2.2: Neural Network score model. The output values of Softmax layer are scores.

Model	AG	Sogou	DBP	Yelp P.	Yelp F.	Yah. A.	Amz. F.	Amz. P.
BoW (Zhang et al., 2015)	88.8	92.9	96.6	92.2	58.0	68.9	54.6	90.4
ngrams (Zhang et al., 2015)	92.0	97.1	98.6	95.6	56.3	68.5	54.3	92.0
ngrams TFIDF (Zhang et al., 2015)	92.4	97.2	98.7	95.4	54.8	68.5	52.4	91.5
char-CNN (Zhang and LeCun, 2015)	87.2	95.1	98.3	94.7	62.0	71.2	59.5	94.5
char-CRNN (Xiao and Cho, 2016)	91.4	95.2	98.6	94.5	61.8	71.7	59.2	94.1
VDCNN (Conneau et al., 2016)	91.3	96.8	98.7	95.7	64.7	73.4	63.0	95.7
`fastText`, $h = 10$	91.5	93.9	98.1	93.8	60.4	72.0	55.8	91.2
`fastText`, $h = 10$, bigram	92.5	96.8	98.6	95.7	63.9	72.3	60.2	94.6

Figure 2.3: Performance of Fasttext on sentiment datasets [24].

	Zhang and LeCun (2015)		Conneau et al. (2016)			`fastText`
	small char-CNN	big char-CNN	depth=9	depth=17	depth=29	$h = 10$, bigram
AG	1h	3h	24m	37m	51m	1s
Sogou	-	-	25m	41m	56m	7s
DBpedia	2h	5h	27m	44m	1h	2s
Yelp P.	-	-	28m	43m	1h09	3s
Yelp F.	-	-	29m	45m	1h12	4s
Yah. A.	8h	1d	1h	1h33	2h	5s
Amz. F.	2d	5d	2h45	4h20	7h	9s
Amz. P.	2d	5d	2h45	4h25	7h	10s

Figure 2.4: Training time of Fasttext on sentiment datasets [24].

Chapter 3

Methodology

3.1 Overview

The approach involves three types of neural networks, i.e., LSTM-RNN language model, LSTM-RNN classifier, and Fasttext sentiment score model. LSTM-RNN language model is used to model sentences within conversations. Its hidden states will be used to represent the sentences. The LSTM-RNN classifier is applied to learn the relation among sentences. The Fasttext classifier is the score model for finding predators. An extra sentiment score model named Fasttext is introduced. The purpose of doing so is based on the hypothesis that the participants of conversations talk to each other with emotion and personal feelings. Then the sentence they generate in conversation can be rated by using sentiment score. In this way, the popular sentiment score methodologies can be applied to rate the sexual predators and regular users by assign them a score from 0 (regular users) to 1 (sexual predators). In other words, taking the sentences generated by different participants as training data is a useful augmentation method. It will reuse the same data in different dimension. The potential features that represent sentiment will be concentrated by regrouping. The classifiers can learn features from conversations scope and participants scope.

3.2 Processing

3.2.1 Role of Processing

The processing part is introduced to make sure that the language model and sentiment score model can use the data with low noise level. The terms, sentences, even conversations that match the filter rules will be removed or replaced. Table 3.1 shows the number of samples before and after filtering

Table 3.1: Attributes of PAN-2012 dataset

	Training		Test	
Type	Original	Filtered	Original	Filtered
Positive	2016	1088	3684	1880
Negative	64911	52854	151210	123229
Non-predators	97547	97291	218488	217997
Predators	142	138	254	215

3.2.2 Processing Strategies

The PAN-2012 dataset [3] contains a great number of chat abbreviations, cyber slangs, and emoticons etc., which will increase the perplexity of language model. The length of conversations varies from one to another. Keywords replacement is a popular preprocessing method in NLP work. To predict the next word more accurately, noise removal is indispensable. However, to secure the generalization ability of the model, the removal methods should keep as much raw information as possible to truly reflect the actual environments. Some words or abbreviations may convey important information, for example, yrs means years, ur means your, etc., therefore recovery of these abbreviations is also necessary. Such removal and replace strategies will also reduce the average length of sentences. All the strategies for processing are listed below:

- Replace all numbers by the symbol 00NUM. According to our sample analysis, the predators are likely to ask the age of victims to find their targets. More specifically,

the conversation participated by sexual predators and victims usually includes age information in number form. However, it is hard to define young or old by providing age ranges. Unlike previous researchers who split numbers into different age groups [5], our strategy is designed to keep raw information as much as possible. As the neural network usually learns the features in its own way, it is important to keep low-level features for learning. On the other hand, there are many regular conversations including numbers. If each number is assigned an age group label, it will generate noises and thus interfere with the classifier.

- Replace all words longer than 30 characters by symbol 00LW. For convenience, PAN replaced author's nickname by using user ID i.e., a long hash code. This replacement generated some hash codes inside the conversations. There are many free style long words, such as sorrrrrr….y, and randomly typed terms in PAN suspicious conversation samples. It is because those conversations are collected from the Internet. The long meaningless words will also increase the perplexity of the language model. On the other hand, the word-frequency of long words is very low. Although it is hard to learn features from them.

- Replace the URL with symbol 00URL in the data. Usually the URL consists of meaningful words. But the data were collected from the Internet. The form of URLs such as http://, ftp://, and file:// are quite common in normal conversions. It is necessary to replace URLs with a single symbol to obtain stable samples with low noise level.

- Remove all non-ascii chars. Since participants of conversations are not all native English speakers and there is no restriction on chat charset, there are many non-ascii chars in the samples. Therefore, removing those chars will also reduce the noise. At the same time, the size of corpus will also be reduced.

- Remove all emoticons. Unlike other researchers using emoticons as manual features to indicate the emotion or sentiment in conversations, based on our data analysis work, it is unlikely for sexual predators to use emoticons to conduct sexual abuse behaviors. On the other hand, it is hard to tell the difference between normal punctuations and

emoticons without introducing complex rules.

- Recover popular yet unofficial abbreviations. The authors tend to use cyber slangs in their conversations. For example, u, r, and ur etc., which obviously can be learnt by neural networks as features. In particular, it is necessary to recover the abbreviations like pics, cam, and yrs, which are common in conversations involving sexual predators.
- Words with term-frequency of less than 10 are removed as noise and the remaining words are sorted by term frequency–inverse document frequency (TF-IDF) weights.

3.3 Recurrent Neural Network

3.3.1 Applications in Our Work

Recurrent neural network (RNN) is used as a classifier and language model in our work. There are two types of learning methods i.e., supervised, and unsupervised. The supervised RNN model is the classifier which identifies suspicious conversations. Unsupervised RNN model, or neural language model, is used to learn the features inside conversations and represent the sentences. The sentences are firstly fed into the unsupervised model to get dense representations. After that, with the dense representations, the supervised RNN classifier learns the features of predators' conversations (Figure 3.1).

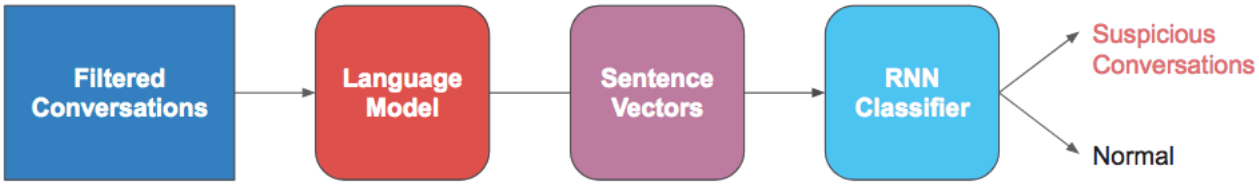

Figure 3.1: Two RNNs work together to identify suspect conversations.

3.3.2 Overview of RNNs

An RNN is a neural network model that processes elements of a sequence one by one and learns the dependencies among previous inputs. In another word, it memorizes the

information that has been processed. It is an artificial neural network that makes use of sequential information, such as long text, times series data e.g., acoustic data, stocks etc.

Simple fully connected neural networks take all inputs, independent of one another. However, for most tasks, it is not good enough. If you are going to predict the price of stocks, you need to know its historical performance and the trend of the market. In our research topic, if you are going to predict a chat content, you need to know previous conversations. Recurrent means occurring repeatedly i.e., each single element of a sequence is processed by the same neural cells of RNNs. Therefore, the activation outputs of the RNNs are dependent on previous inputs. RNNs can learn knowledge arbitrarily in length of sequences. However, usually it is only able to roll back for a few steps.

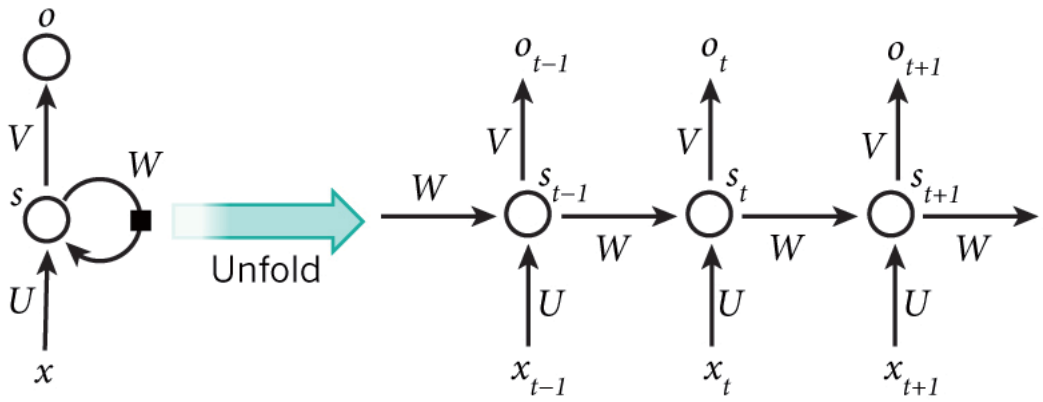

Figure 3.2: Structure of unfolded RNN [25].

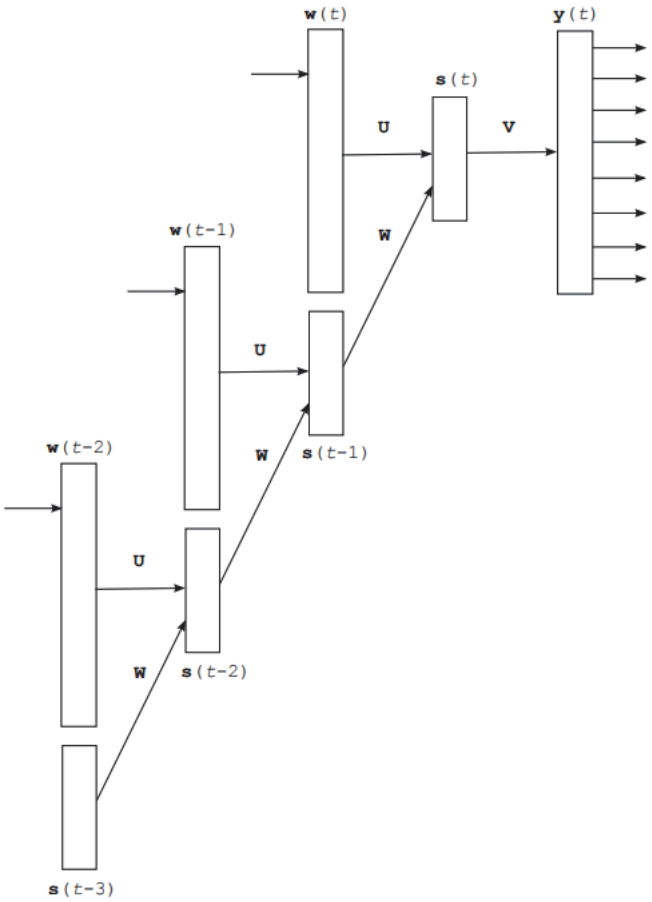

Figure 3.3: Inputs are fed into the RNN with different time steps [26].

3.3.3 Mathematics in RNN

Figure 3.2 shows an RNN unfolded into a simple fully connected neural network. For example (Figure 3.3), there is a sentence of three words. It is a three-layer unrolled neural network. Every single word is the input of each layer. There are three type of layers, i.e., Input X, Hidden S, and Output O. The input of the RNN at time step t is $x_t \in R^n$ and the hidden state is $s_t \in R^m$.

$$s_t = f(Wx_t + Us_{(t-1)}) \tag{1}$$

s_t is calculated by (1), based on the previous hidden state $s_{(t-1)}$ and the current input step x_t where the function$f$ is a nonlinear function i.e., *tangent* or $ReLU$. The y_t is the output

at time step t. It is a vector of probabilities corresponding to each word in the vocabulary. To predict the next word in a sentence, the index with maximum probability in the vector will be the word index in the vocabulary. The output is given by (2).

$$o_t = Softmax(Vs_t) \tag{2}$$

Unlike a fully connected deep neural network, where there is no repeated layer for processing different inputs, the RNN uses the same weights throughout the entire learning procedure. The reason is that the same neural cell processes different inputs of the sequence. It will reduce the number of neural cells inside the network. There are outputs for each time step of RNN, however, for sentiment analysis or classification work, only the last output is needed while for language model each output is needed. The very important thing of RNN is the hidden state, as it captures useful information of the sequence.

3.3.4 Gradient Vanishing and Exploding

RNN is featured by its ability to capture dependencies in sequences and share the same parameters (U, V, W) throughout all steps. Theoretically, an RNN learns through all previous time steps, however, due to vanishing gradients problem [27], it is hard to capture long-term dependencies. More specifically, it cannot learn information from a long time ago. For a simple example, "*The fork is on the table.*". It will not need much information from previous context to predict the "*table*". But for a longer sentence, "*I don't like cheese. I will not eat Pizza.*", The "eat" indicates that the prediction will be a food name. The context of "*cheese*" is very important for predicting. The vanishing gradient problem will limit the ability for learning through long time.

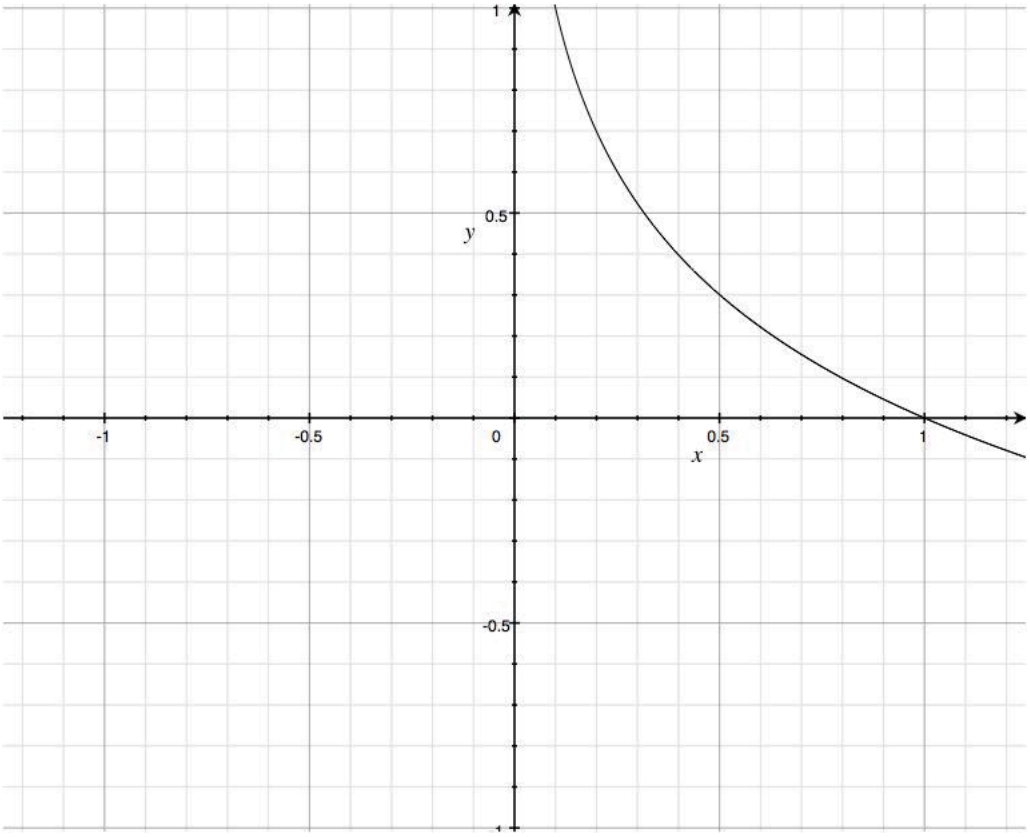

Figure 3.4: Cross entropy loss.

The cross entropy loss is given by: $E_t(o_t, \hat{o}_t) = -o_t \log \hat{o}_t$ where o_t is the ground truth at time step t, and $\hat{o}_t$ is the prediction result. In Figure 3.4, i.e., $y = -1 * log(x)$, let y be the cross-entropy loss and let x be the prediction. When the prediction is near 1 i.e., the ground truth, the loss is getting lower. Otherwise, the loss will be infinity. The goal of training is getting the minimum loss. It can be done by calculating the gradient of the loss. The gradient of loss E is $\frac{\partial E}{\partial U}$, where $\frac{\partial E}{\partial U} = \sum_t \frac{\partial E_t}{\partial U}$ and

$$\frac{\partial E_t}{\partial U} = \sum_{k=0}^{t} \frac{\partial E_t}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial s_t} \left(\prod_{j=k+1}^{t} \frac{\partial s_j}{\partial s_{j-1}} \right) \frac{\partial s_k}{\partial W} \tag{3}$$

The output of sigmoid activation function (Equation 1) is mapped into a range between 0 and 1 (Figure 3.5). The derivative of E_t is equation 3. According to the paper [28], as the values $\frac{\partial s_j}{\partial s_{j-1}}$ (from 0 to 1) are multiplied by each other at each time step i.e., $\left(\prod_{j=k+1}^{t} \frac{\partial s_j}{\partial s_{j-1}} \right)$, the derivative from long time ago is very easy to become zero. On the other hand, other

types of activation functions and network parameters will lead to exploding gradients if the values are large. This is the problem that limits RNN to learn from long steps. Although this problem also occurs in other feed-forward neural networks, it is more problematic in RNN as the depth of RNN is very deep compared with others. Using LSTM can prevent gradient problem.

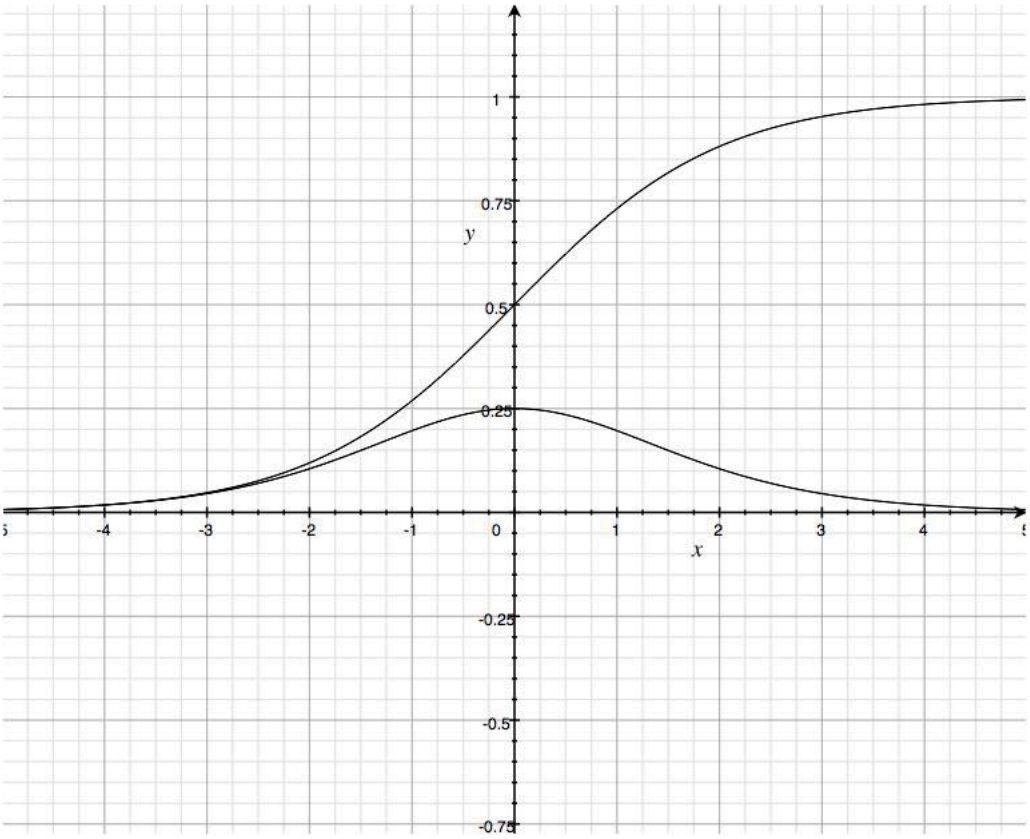

Figure 3.5: Sigmoid function and its derivative.

3.4 Long Short-Term Memory RNN

LSTM (Long Short-Term Memory) is a special cell structure that is able to deal with information from a long time ago. Similar to RNNs, it takes in the previous hidden state and the current step as input, then outputs a new hidden state. The errors of a neural network can backpropagate to unlimited numbers of unrolled layers i.e., time steps. LSTM-RNN works well when there are long delays among events. It proposes a gating mechanism

to avoid vanishing gradients problem. More specifically, a new state c_t is introduced to calculate hidden state s_t (Figure 3.6).

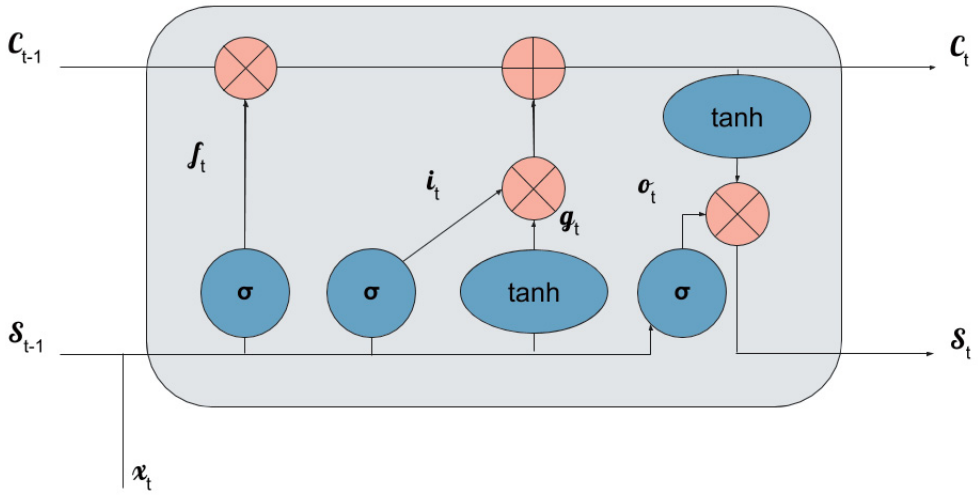

Figure 3.6: LSTM-RNN cell. The σ is sigmoid function.

c_t, and s_t are calculated as below:

$$i_t = \sigma(x_t U^i + s_{(t-1)} W^i) \tag{4}$$

$$f_t = \sigma(x_t U^f + s_{(t-1)} W^f) \tag{5}$$

$$o_t = \sigma(x_t U^o + s_{(t-1)} W^o) \tag{6}$$

$$g_t = tanh(x_t U^g + s_{(t-1)} W^g) \tag{7}$$

$$c_t = f_t * c_{(t-1)} + i_t * g_t \tag{8}$$

$$s_t = o_t * tanh(c_t) \tag{9}$$

where i_t, f_t and o_t are input, forget gate and output gate respectively. The σ is sigmoid function. The g_t is the hidden state like s_t in RNN being used to compute the new s_t. They have the exact same functions, but different parameter matrices. All the gates process the same shape of matrices i.e., the shape of the hidden state. Those functions are named gate

because the sigmoid function maps the values of the inputs to outputs which are from 0 to 1. The outputs multiply the states vector to calculate how much of that state vector will pass through. The forget gate decides how much of the previous state C_{t-1} will be kept. The input gate decides how much of the new weights for the input x_t and S_{t-1} will pass through. The output gate decides how much of the new state will be outputted to the next time step (or another neural layer).

Forget gate decides what information is going to be dropped. This decision is made by a function f_t. It takes S_{t1} and x_t as input and outputs a value between 0 and 1. It will keep all of the C_{t-1} when the output is 1 and will completely drop it when the output is 0.

Input gate decides what new information is going to be kept. To do this, there are two functions involved. A *sigmoid* function decides which values will be updated. Then, a *tangent* function generates a matrix with new values, i.e., g_t (7) which can be added to the state C_{t-1}. Next, these two states will be combined to create a new value C_t so as to update the state C_{t-1}. The old state C_{t-1} is multiplied by f_t, which learns what will be forgot. The remaining $f_t * C_{t-1}$ then adds $i_t g_t$ to get the new candidate values (8). Those values indicate how much information will be used to update state C_{t-1}.

Output gate decides what is going to be outputted. This output is based on the cell state processed by the forget gate and the input gate. The output is going to be a filtered state. Firstly, the *sigmoid* function decides how much of the cell states (x_t, S_{t-1}) will be outputted as O_t. Next, the processed state C_t will be inputted into a tangent function to get values between -1 and 1 and multiply O_t (9). Finally, the new cell state C_t and new output S_t are generated for the next step.

3.5 Neural Language Model

3.5.1 Role of Neural Language Model

To identify the predators or the suspicious conversations, it is necessary to express the probabilistic distribution of sentences precisely in a vector space. The language model learns the features inside sentences. The features in our work are sentence vectors, which is a middle state of the neural language model. With those sentence vectors, another LSTM-RNN classifier is trained to distinguish if a predator is involved in the conversation.

3.5.2 Overview of Neural Language Model

The goal of a neural language model is to compute the probability of a sentence i.e., to fit a model that assigns probabilities to a sentence. It does so by predicting the next words in a text based on a history of previous words. The creation of word embedding, text generation, and text classification are a few typical applications of neural language models [16] [22]. A neural language model measures the similarity of two words based on the likelihood of them appearing together in a real text. It provides an indicator of grammatical and semantic correctness. Neural language model can generate new texts based on the information it learns from real-world texts. For example, a neural language model trained on Shakespeare can generate text that resembles Shakespeare's style. It also can generate source code or Latex [29].

A popular application of neural network language models is the creation of word embeddings to represent words. The neural network takes words from a text corpus as input and maps them to vector space. Word embeddings represent words in vector space where similar words are mapped near each other. It assumes that words appearing in the same contexts share similar meanings. Word embeddings were originally introduced by Bengio, et al. [16]. By using word embeddings, analogies between words can be represented by the difference of vectors. For example, E(w) means the embedding of word w, "E(King) - E(Man) + E(Woman)" generates a vector that is very close to "E(Queen)" [17]. In this case, the word

embedding learns the representation of gender. In this way, semantic information can be stored.

The neural language model can also be used for text classification purpose[16]. Text is a sequence of words. The state of embedding layer of a neural language model is a dictionary containing the vector representation of words. Word embeddings are unique vectors that can be added and subtracted [17]. Therefore, the vector of sentence is the average value of its word vectors obtained from the embedding layer. Later on, the representations of sentences can be fed into either traditional classifier e.g., SVM, regression and Naive Bayes, or deep neural networks such as RNNs and CNNs. More specifically, the state of hidden layers in RNN language model can also be a representation of the sentence. This is because RNN keeps previous information in the hidden state.

The classical structure of a language model includes an embedding layer, hidden layers, and a softmax layer. The embedding layer is the layer that contains word embeddings which is a lookup table to output the representation of words. The weights of the first layer are the word embeddings. The hidden layers produce the outputs mapped from the input, e.g. a fully-connected layer using sigmoid function to process the word embeddings of previous words. The Softmax layer is the last layer that outputs the probability distribution of words in text corpus (Figure 3.7).

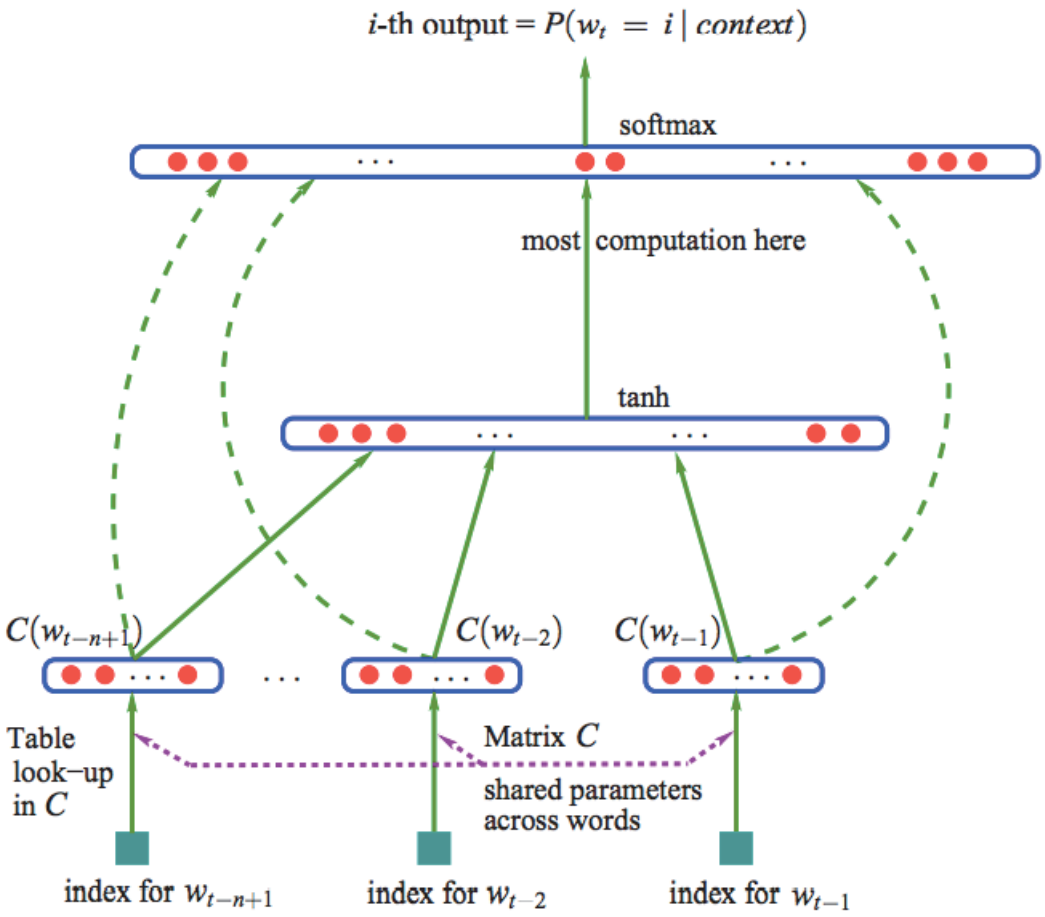

Figure 3.7: Neural language model [16].

3.5.3 Measure of Neural Language Model

A neural network language model takes a word sequence $W = [w_1, \ldots, w_t], w_t \in V$ where V is the vocabulary set as input and learns to predict the probability $p(w_1, , w_t)$ of the next word w_{t+1} by applying the softmax activation function at the output layer.

$$p(w_1, \ldots, w_t) = Softmax(s_t^\top e_{w_t})$$

where $e \in E[e_{w_1}, \ldots, e_{w_t}]$ [16]. The maximum log-likelihood principle is applied to maximize the probability of the word w_t when given the contexts w_1 to w_{t-1}. The training

of neural network uses the back-propagation algorithm over time to maximize the log-likelihood (10) of training data. The input is mapped to vectors e_{w_t} in vector space within the neural network.

$$L(\theta) = \sum_{t} logP(w_{t-n+1}, \ldots, w_{t-1}) \tag{10}$$

Learning the probability distribution over a text corpus is the key feature of language models. Normally, language models are often evaluated by using perplexity, a cross-entropy based method. Perplexity is the possibility of how many words can be selected after being given previous words. The lower perplexity, the better the language models. The size of the vocabulary will influence the perplexity as the totality of potential words will limit the selection of the next word. This method of measuring a language model is developed from information theory [30]. A text corpus is a discrete information source that generates a sequence of words $w_1, w_2, \ldots, w_n$ from a vocabulary set. The dependent probability of word w_n relies on previous words $w_1, w_2, \ldots, w_{n-1}$. The entropy H represents the amount of non-redundant information (11) in the corpus. When given a large text corpus, H can be approximated by $\hat{H}$ (12). The language model is an information source which owns the entropy of H. The entropy related method can be used to measure the performance of a language model. Perplexity, PP, is defined by (13) which is equivalent to (14). $p(w_1, w_2, \ldots, w_m)$ is the probability of the word sequence $(w_1, w_2, \ldots, w_m)$ estimated by a language model.

$$H = -\lim_{m\to\infty} \frac{1}{m} \sum_{w=1}^{m} (p(w_1, \ldots, w_m) log_2 p(w_1, \ldots, w_m)) \tag{11}$$

$$\hat{H} = -\frac{1}{m} log_2 p(w_1, \ldots, w_m) \tag{12}$$

$$PP = 2^{\hat{H}} \tag{13}$$

$$PP = \hat{P}(w_1, \ldots, w_m)^{\frac{1}{m}} \tag{14}$$

3.5.4 LSTM-RNN to Language Model

LSTM-RNN language model can be obtained by replacing the hidden layer with LSTM-RNN layers in language model (Figure 3.8). At the beginning, the hidden state h_1 is initialized with zero. In the first time-step, the input to the LSTM-RNN language model is w_1. The hidden state vector is updated at the same time and passed to the next step. In the second time step, the input is w_2, and the output is $p(w_2|w_1)$. The s_2 is the hidden state that contains information from previous step. At each time-step, LSTM-RNN learns the probability of the words in the vocabulary. The output layer of the LSTM-RNN language model is the softmax layer which returns a vector. The output is a group of probabilities of each word in the vocabulary when given w_t.

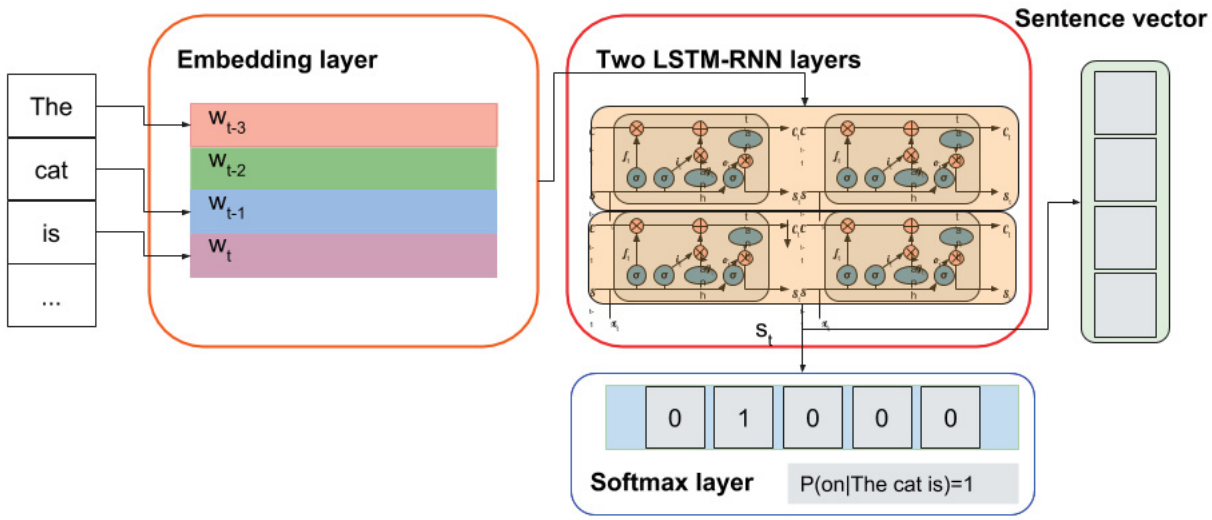

Figure 3.8: LSTM-RNN language model.

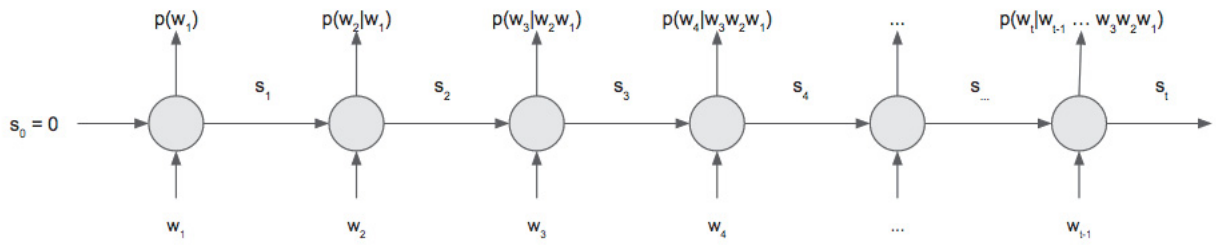

Figure 3.9: Unfolded LSTM-RNN in language model

3.6 Sentence Vectors

3.6.1 Overview of Sentence Vectors

In our work, the sentence vector is fed into the RNN classifier to find suspicious conversation involving predators. A sentence vector is a by-product of LSTM-RNN language model. It is the hidden state of the last layer of the language model and is used to represent the sentence. It takes a sequence of words as input and the hidden state of the last word in the sequence as the sentence vector.

The sentence vectors (Figure 3.10, Figure 3.11) are inspired by Seq2Seq model (Figure 3.12 [31]) which has been successfully applied in neural machine translation. In traditional phrase-based translation systems, the source sentences and target sentences are broken into small chunks. This leads to information loss. It is not like human translation process.Humans read the whole sentence, interpret its meaning, and generate the translation [33]. The Seq2seq structure is a encoder-decoder architecture similar to auto-encoder, which can compress information [32]. Seq2seq processes the source sentence by using the RNN model in the encoder part to build meaning outputs, a vector space representation of source sentence. The outputs then are fed into the second RNN model in the decoder. The output of the decoder is the target sentence (Figure 3.13). In our work, we simplified the structure of the seq2seq model by just taking the hidden state from encoder RNN as

memory to represent the sentence.

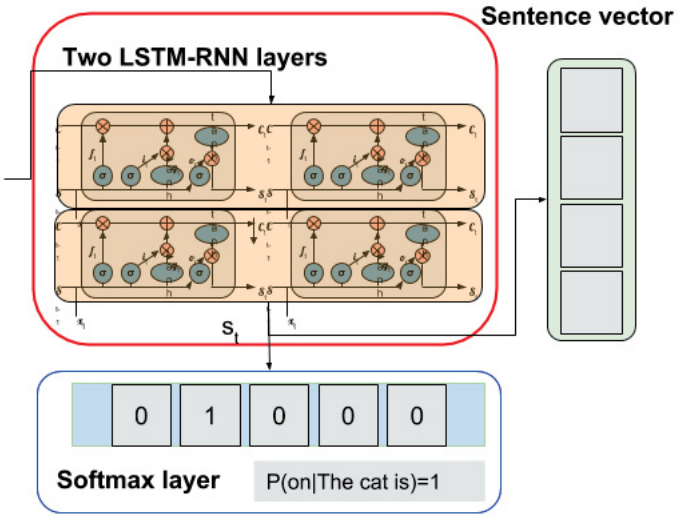

Figure 3.10: Inside of the sentence vector.

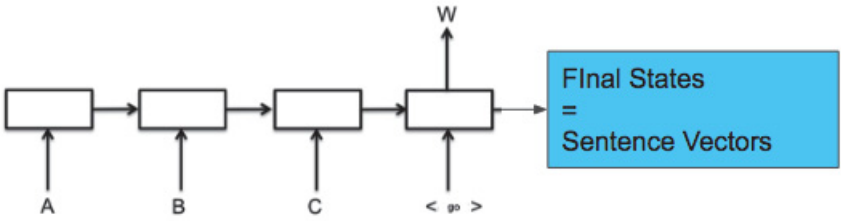

Figure 3.11: Sentence vector.

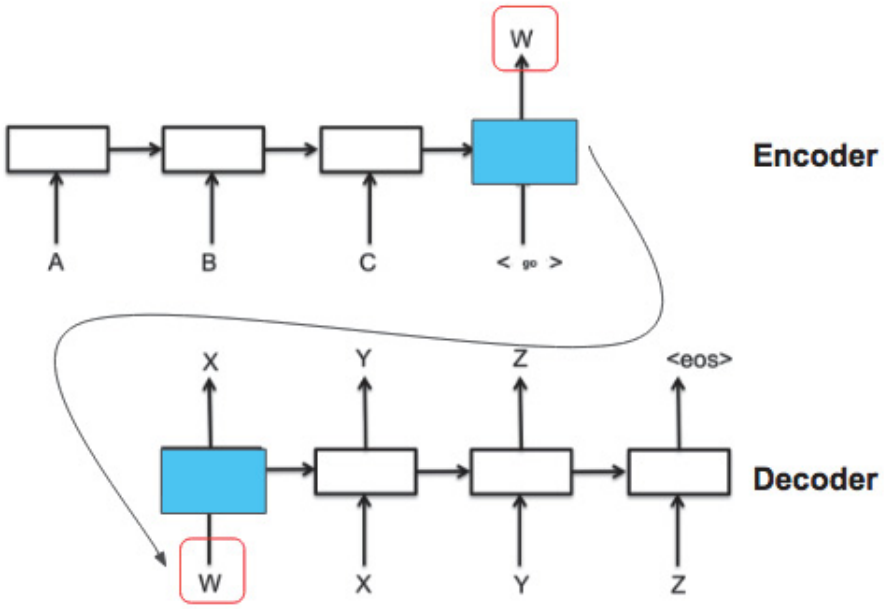

Figure 3.12: Seq2seq model in neural machine translation.

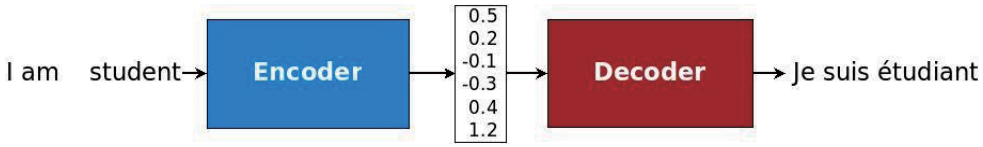

Figure 3.13: Encoder-decoder architecture. An encoder converts a source sentence into a "meaning" vector which is passed through a decoder to produce a translation [33].

LSTM-RNN neural network language model is composed of three layers. The first layer is the embedding layer. The embedding layer represents words as dense vectors. The second layer is LSTM-RNN model which learns the dependencies among the words in the sentences. The final layer is the Softmax layer, a multinomial logistic regression layer used to solve multi-class prediction problems. The hidden layers store the information of the sentence (Figure 3.8). Specifically, the last time when step hidden state in LSTM-RNN language model st is used to represent the input sequence $\{w_1, \ldots, w_t\}$. The purpose is to minimize prediction errors. Similar sentences will activate the same neurons in the last layer.

3.6.2 Pros and Cons of Sentence Vectors

Compared with using the average word embeddings as sentence vector, this representation reduces the length of inputs and captures the dependencies among the sequences of words. Besides, the average method may cause confusion in the case of longer sentences. A longer sentence increases the risk of conflict. This is because the addition or subtraction will keep the high level meaning e.g., gender information mentioned above [17] rather than details. Word embedding method cannot drop useless information in the sentence either. It is the gate mechanism that will help the neural network to drop useless information. Using RNN language model to compress words sequence into sentence vectors will increase the learning speed and get a more embedded expression. Moreover, LSTM-RNN based sentence vectors have the advantages of being able to capture the dependency and compress the size of conversations. The performance of this method on short sentence is good. There are many short sentences, unrestricted terms, typos, and cyber slangs in PAN-2012 dataset. Therefore, it is hard to apply traditional language model to this task. In addition, there is no sufficient extra materials for building language model for this task. Although the twitter dataset includes many typos and cyber slangs, it is not a conversation-based dataset and thus is unable to reflect contextual information, such as chats.

However, this type of sentence vector performs poorly on IMDB dataset as it is unable to express long sentences. It brings the accuracy to 83.2%, comparable to Skip-thoughts (82.5%, [34]). For longer sentences, Self-Adaptive Hierarchical Sentence Model [35] may perform better (inference based on [36]).

3.7 Conversation Classification

3.7.1 Motivation

The goal of the PAN-2012 competition was to identify predators. To that end, the first thing needed is to find suspicious conversations and feed them into a sentiment score model. The

training and test dataset in PAN-2012 are unbalanced. There are only a few true positives (predators) and many false positives (victims or regular users) (Table 4.1). To find the predators inside the conversations, we need to find suspicious conversations. Therefore, the behaviour features are very important. According to our analysis, the behaviour of predators can be decomposed into three stages in general i.e., three-stage features.

Table 3.2: The samples' distribution of PAN-2012.

	Training		Test		Ratio (Training:Test)
Type	Original	Filtered	Original	Filtered	
Positive Chats	2016	1088	3684	1880	≈1:2
Negative Chats	64911	52854	151210	123229	≈1:2
Ratio (Positive:Negative)		≈1:48		≈1:65	
Predators	142	138	254	215	≈1:1.5
Non-predators	97547	97291	218488	217997	≈1:2
Ratio (Predators:Non-predators)		≈1:705		≈1:1013	

(1) The first stage is privacy information related. For example, the predators will ask for the age of the victims. Other age related questions will also be asked by the predators for information collection purpose. With such information, the predators will move on to the next stage.

(2) The second stage is asking for the information of the victims' parents. Predators know their behaviour is illegal. They try to induce victims to hide the existence of their conversations from their parents. Later, the topics related to time e.g., week, day, schedule are very common. Also, there are the topics on family and household. The predators always try to get as much information about victims' parents as possible e.g., if their parents are at home or if they are aware of the conversations. For most of the situations, human review of stage one and stage two can confirm if a participant is a predator. Nevertheless, the topics related to age and family still belong to general topic.

(3) In the third stage, the predators are ready to launch the attack. For example, the predators may ask question about the presence of a camera for a video chat or try

to obtain pictures from victims. Some of the predators even attempt to persuade the victims to have an offline meeting. This stage reveals the key features of predators.

The decomposition provides an overview of the predators' behaviour. For the language model to learn the features of the three different stages, it requires a method that keeps as much information as possible. The method of averaging word embeddings of a sequence of words will cause information loss as the extreme values in the vector will be removed. Since RNN can keep the context information, the sentence vector is a good choice for representing the features inside conversation. With sentence vectors, the conversations can be represented. LSTM-RNN, which learns long period, is a good model for conversation classification work, as online chats are based on time.

3.7.2 Features and Assumptions

After analyzing the dataset, we observed five features related to conversation classification (Table 3.3):

(1) Sex related topics in both training and test data. Those conversations on sexual topics are easily confusable with true positive samples, i.e., conversations involving predators.

(2) There are topics related to parents in the dataset. Such topics are not so confusing as the first type.

(3) People also talk about cyber camera, photos, and videos. Those terms also appear in a typical predator's conversation.

(4) Conversations including age, number, and other age-related contents are very common.

(5) Apart from those predators related features, the training and test sample include many general topic conversations. The side-effect is that the conversation classifier will learn features from general topics rather than predators' conversations.

With the features of conversations above, the assumption is "Conversation contains all three

Table 3.3: The features of the dataset of PAN-2012.

	Total Sent.	Dad	Mom	She	Pic	Sex	Cam
Normal Chats	775805	509	1064	855	4117	3267	2511
Predator Chats	79178	439	809	560	666	470	332
Percentage in Normal		0.07%	0.14%	0.11%	0.53%	0.42%	0.32%
Percentage in Predator		0.55%	1.02%	0.71%	0.84%	0.59%	0.42%
Ratio (Predator:Normal)		8.45	7.45	6.42	1.59	1.41	1.3

stages is likely to involve predators. On the contrary, any conversation only includes part of the three stages is normal." The classifier should distinguish the three-stage features from single-stage feature. To support this hypothesis, a classifier with context relation learning ability is necessary.

3.7.3 Contributions

In our work, LSTM-RNN sentence vector model is introduced to represent sentences. The LSTM-RNN classifier, which is good at learning in contexts (time sequence features), is used to identify suspicious conversations. Unlike a bag-of-words, sentence vector compresses the information of a sequence of words. As the size of training data is reduced, the vector space model can increase the training speed. In addition, the sentence vector keeps more context information than bag-of-words does. LSTM-RNN is good at learning through time. It is better than convolutional neural network (CNN), which is good at spatial learning. In previous work, there were two typical neural networks related approaches i.e., fully connected neural network [4] and CNN [37]. They used bag-of-words model as inputs of the neural network. The disadvantage of bag-of-words model is that it generates redundant words group and therefore will increase the size of vocabulary. CNN demonstrates impressive ability in the image processing. It is hard to train a fully connected neural network on real image dataset as the input size of an image is very large. One of the purposes of using CNN is to reduce the connections among hidden neurons in the neural network so as to accelerate the training procedure. Another purpose is to capture features in very small scale. CNN is inspired by cat's visual cortex [38] [39] and it is good at capturing 2D

or 3D features in small area. Based on the previous research [37], CNN did not get ideal classification results.

3.7.4 Workflow

Regarding suspicious conversation detection, LSTM-RNN has strong ability to learn the long-term dependencies among time steps, which means it can capture relations among sentences that contain the features of predators. There are three steps in the identification of suspicious conversations.

First, the filtered conversations without punctuation are used for LSTM-RNN language model training. The goal of this step is reducing the perplexity of the language model i.e., to increase the prediction accuracy of the model. Secondly, sentence vectors are generated for all conversations. Each word of sentences is fed into well-trained LSTM-RNN language model. When the last word in a sentence is processed, the hidden state of last layer is the sentence vector. Lastly, after getting sentence vectors, conversations with sentence vectors are input into a three-layer LSTM-RNN-based model and the latter will learn the context features (Figure 3.14). Considering different number of sentences in conversations (from 1 to more than 500 sentences), those extra-long conversations will be padded by zeros and then split into parts, each with an equal length of 100 words (an experience-based value). This strategy will prevent underfitting in LSTM-RNN model when processing long conversations as there are only a few of them. The features are well-distributed in suspicious conversation. A predator is very likely to carry out criminal activities during the entire conversation.

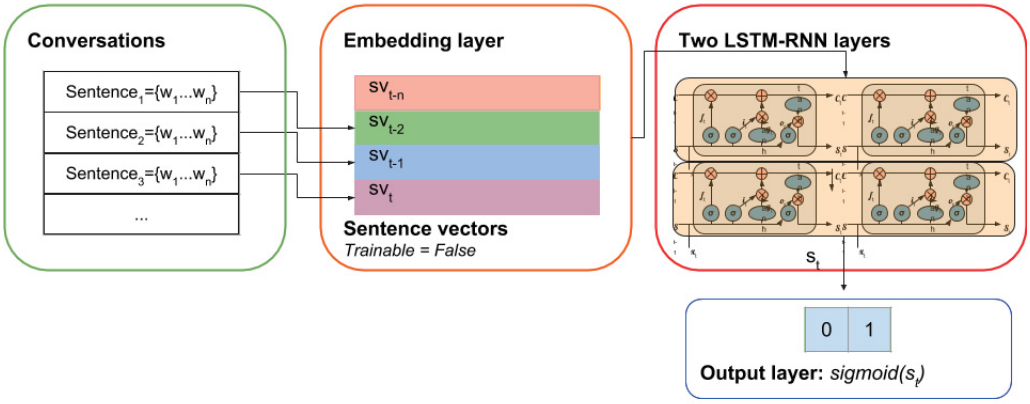

Figure 3.14: LSTM-RNN classifier.

3.8 Participant Classification

3.8.1 Motivation

The goal of the participant classification is to find the predator from suspicious conversations based on the results from conversation classifier. The suspicious conversation classifier is a filter that narrows down the range of predators identification. It helps the sentiment score model to work on suspicious conversations involving predators only. The participant classifier is influenced by the degree of precision of the conversation classifier. Data analysis shows that a predator usually attacks the same victim more than once or attacks more than one victim. To achieve the goal of identifying predators, the following steps are carried out.

(1) Re-group the conversations by participants, i.e., split each conversation by participants. This can help the participant classifier to focus on predators' behaviours only.

(2) Considering the re-grouping will disrupt the time dependencies inside the conversations, a new neural network model named Fasttext is introduced. Fasttext is a type of neural network that averages the features among input words. It is an ideal model for sentence classification.

(3) The attacking behaviours of one predator may distribute across different conversations. Some conversations provide only a few information, for example, when one of the participants is offline or is too busy to respond more. However, the same participants may start another conversation in another time and provide a large amount of information. To get an objective rating of the participants, the sentiment score assigned to them must be averaged. This strategy can overcome the disadvantage of unbalanced information.

(4) The predators are assigned a score of 1, and the victims are assigned a score of 0. This score is the same as classification result from the Softmax layer of Fasttext model.

3.8.2 Features and Assumptions

The features of participants are listed below.

(1) Predators and victims are not paired. One predator may try to contact different victims or communicate with same victim in different conversations (over different time). Some of the conversations launched by predators may not contain any useful information indicating a potential attack, while some conversations may reveal a potential attack but there is only one participant (only predator in the conversation).

(2) Regular users, victims and predators are separated into three categories. The reason of such categorization is that regular users talk about general topics while predators conversations meet the three-stage features.

The key difference between predators and regular users (regular users who talk about sex-related topics) is their behavioral pattern. Many behaviour and psychology related topics

were reviewed, including: Linguistic Inquiry and Word Count (LIWC), the Knowledge-Based Conversation Filter (KBF), etc. More specifically, the KBF module builds 51 hand-coded patterns. LIWC is a psycholinguistics method that counts the number of words. LIWC is based on the assumption that the ways in which people talk and write provide clues of their emotion and cognition. Therefore, LIWC can be used to analyze their behaviour. However, those patterns are hand-coded, while the deep learning method always learns pattern by itself. Based on the above-mentioned features, the assumptions are:

(1) Predators always want to launch attacks on targets.

(2) The average sentiment score of predators must be higher than victims.

(3) The predators always ask victims questions with attacking intention.

(4) The predators, victims, and regular users play different roles in their conversations. There are significant differences among their chat contents.

(5) The victims' chats content may deceive the classifier as sometimes they are answering the questions posed by predators. However, after averaging their scores, the score of victims is unlikely to be higher than that of the predators.

3.8.3 Contributions

In this part of work, the Fasttext model is introduced to generate the sentiment score for rating participants. There are three types of the Fasttext model. As a result, there are three scores for each type of participants i.e., predator score, victim score and normal score. These scores are the classification results from the outputs of Softmax layer. The difference between other researchers' work and ours is that we averaged the score by participants. This method overcomes the problem of confusion between predators and victims. Averaged score kept the sentiment information from conversations and reduced the impact of extreme situation, for example, conversations where only predators are involved or victim act like a predator unintentionally. Previous researcher [4] introduced binary classifier to find predators. However, this method only takes predators and victims as training sample

without using normal conversations. The advantage of doing so is that the noise from normal conversations is eliminated. But the trained classifier might not be able to work with normal conversations in test samples.

3.8.4 Workflow

Different from LSTM-RNN, which is good at capturing time series features, Fasttext is a very shallow neural network capable of global feature extraction. Once the suspicious conversations are identified by LSTM-RNN model, the Fasttext-based classifier can identify sexual predators among the participants. In addition, conversations only involving predators will not be deleted because they may contain useful features. In order to improve the accuracy of sexual predators identification, sentiment score is assigned. There are three types of scores, i.e., P, V, N, to categorize participants into predators, victims, and normal users. The output of Softmax layer is the most ideal score model. A participant with a higher score in a certain score type among the three will be classified into that group. It is unlikely for two predators to appear in the same conversation, therefore a participant with the highest score in P category will be identified as the predator. As there might be conversations initiated by the same participant at different occasions, the sentiment score of the same participants is averaged (Figure 3.15). The experimental setup of our research is described in the next chapter.

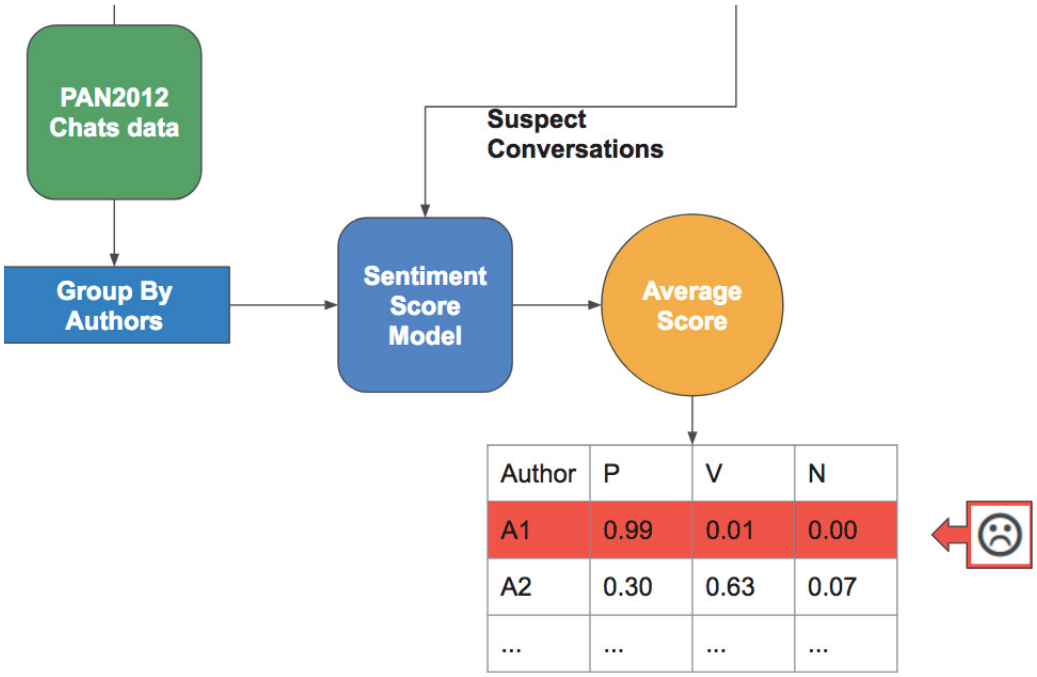

Figure 3.15: Workflow of participant classification.

Chapter 4

Experiments

4.1 Overview of the System

Detecting sexual predators involves the following steps (Figure 4.1):

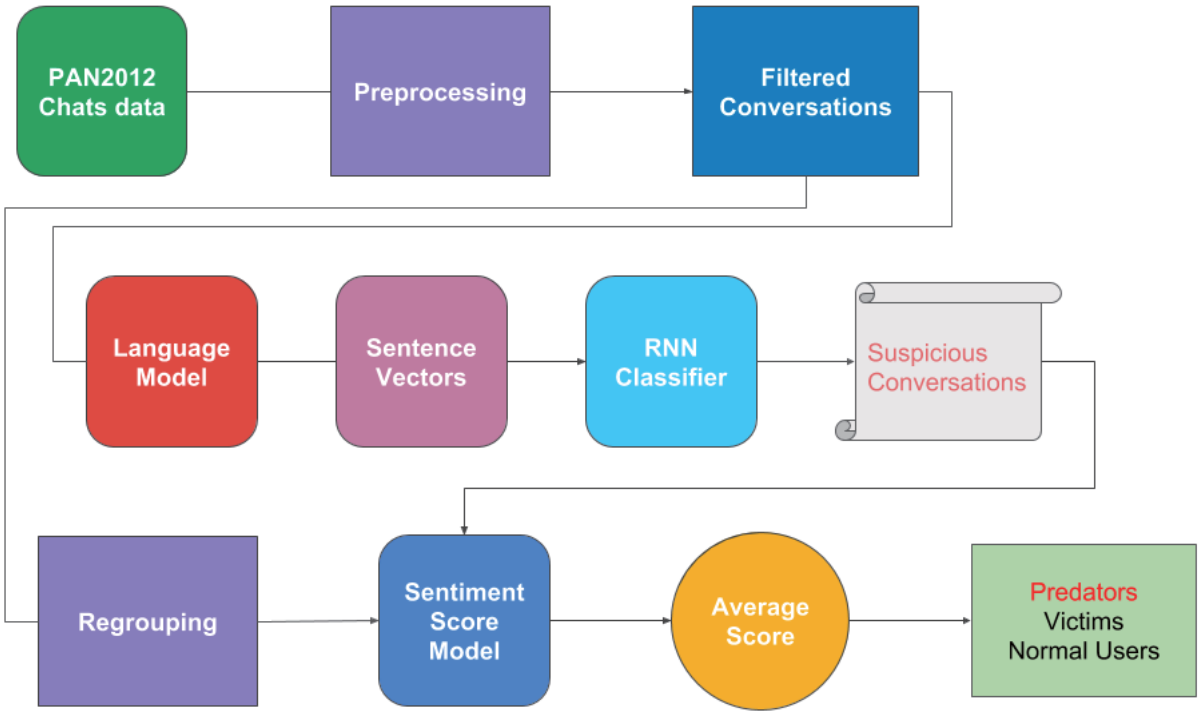

Figure 4.1: Detailed steps of predator identification.

(1) Preprocessing. All the conversations in the training dataset and test dataset are

preprocessed by the rules mentioned in Processing Strategies. Each conversation takes one line in the file.

(2) Identifying suspicious conversations. The filtered conversations in the training dataset are fed into LSTM-RNN Language Model. The LSTM-RNN-based language model is trained for expressing the relation inside sentences. With a number of sentence vectors, the conversation can be presented in a highly compressed way. After the training procedure, each sentence in each conversation (in both training and test dataset) is converted to sentence vector. Each conversation is represented by a group of sentence vectors. The processed conversations in training dataset are used to train a LSTM-RNN conversation classifier. After that, the conversations in the test dataset are fed into the conversation classifier. Finally, the suspicious conversations are outputted for next step.

(3) Identifying predators. The preprocessed conversations in training dataset are re-grouped by participants. After that, the conversations are fed into Fasttext model. The Fasttext based classifier is used to score the participants. Finally, all participants in conversations of test dataset are scored by their sentences of the chats.

As a novel method—sentence vectors—is introduced in our work, an objective evaluation is needed. Therefore, the IMDB sentiment review dataset is used to evaluate the performance of the model, i.e., sentence vectors methodology. There are 25,000 reviews in the dataset. Half is positive, and the other half is negative. The IMDB movies reviews are very popular. It is easy to compare our result with others.

(1) The IMDB movie reviews are filtered by the rules mentioned in Processing Strategies. Each review takes one line in the file.

(2) The filtered reviews are used to train the LSTM-RNN neural language model. The input data is from training samples of IMDB dataset.

(3) Each review is converted to a group of sentence vectors via language model. After that, a LSTM-RNN classifier is trained to measure if a review is positive or negative (Figure 4.2).

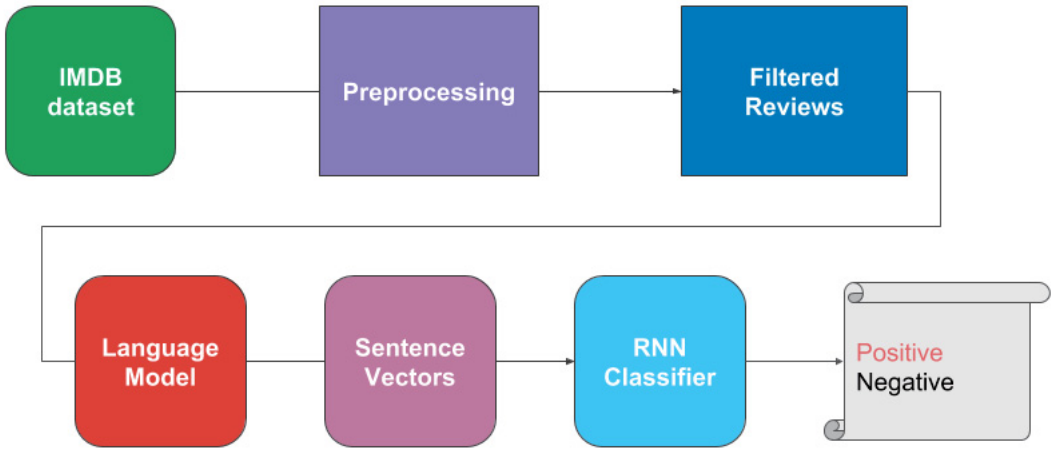

Figure 4.2: Overview of IMDB sentiment analysis.

4.2 Performance Criteria

For the performance indicators, the criteria from [1] are referred. The author of [3] took Precision (P)(16), Recall (R)(17) and F measure (18) from standard information retrieval as measurements where β is 0.5. True positive (TP) means the number of identified predators in the dataset. False positive (FP) means the number of non-predators that are identified as predators in the dataset. True negative (TN) means the number of non-predators that are identified as non-predators in the dataset. False negative (FN) means the number of predators that are identified as non-predators. The accuracy is defined by (15).

		Prediction	
		No	Yes
Ground Truth	No	TN	FP
	Yes	FN	TP

Figure 4.3: Definition of TN, TP, FP, and FN.

$$Accuracy(A) = \frac{(TP + TN)}{(TP + FP + TN + FN)} \tag{15}$$

$$Precision(P) = \frac{(number\ of\ relevant\ items\ retrieved)}{(number\ of\ retrieved\ items)} \tag{16}$$

$$Recall(R) = \frac{(number\ of\ relevant\ items\ retrieved)}{(number\ of\ relevant\ items)} \tag{17}$$

$$F_\beta = (1 + \beta^2) \cdot \frac{P \cdot R}{\beta^2 P + R} \tag{18}$$

The "retrieved items" means the ids of the participants that are identified as predators. In "relevant items retrieved", "relevant" means the total number of true positive and true negative results. Considering the real-world situations, the designer of PAN-2012 [3] hopes to provide as many suspicious predators as possible. They choose both F-1 and F-0.5 as F measurement. The F-1 measurement sets β =1, which means the contributions of P and R are the same. Since they want to find more suspicious predators, the P is more important than R. Therefore, based on (18), the β is assigned with 0.5.

The official rank of PAN-2012 is used to compare the performance. The IMDB sentiment review dataset is used to measure the performance of sentence vectors.

4.3 Dataset

The performance of the models is evaluated on two datasets, PAN-2012 dataset and Stanford Large Movie Review Dataset (IMDB sentiment review dataset) [15]. Performance will be measured against existing publications on sentiment classification tasks. In this chapter, the property of the datasets will be summarized, covering statistical features, lexical features, and behavioral features.

4.3.1 PAN-2012 Dataset

Overview

PAN-2012 dataset is provided in the context of Sexual Predator Identification (SPI) Task in 2012 initiated by PAN (Plagiarism analysis, authorship identification, and near-duplicate detection) lab. There are two datasets in PAN-2012. There are over 200,000 short sentences (online chats) and only 1,088 true positive samples for learning. The number of true negative samples (non-predators) is large. More specifically, it means that there are only 142 chat users available for training. The goal is to find the 254 chat users from more than 200,000 users.

Data Format

The file format of PAN-2012 dataset is Extensible Markup Language (XML). It is a file format being used to store information with standard ASCII text. PAN-2012 provided two files for training: one that includes all conversations and the other that contains the ids of predators (Figure 4.4). The test dataset shares the same structure with training datasets. Each conversation contains the author id, the message time and the message. To train the conversation classifier, as there is no label for conversations, the conversations contain predator is labeled as "positive". Each message in a conversation in the XML file is filtered based on the strategies above. All of the conversations are in one file and each of them takes one single line.

```
<conversation id="2c1892c998f3e223d57c28da3b856169">
  <message line="1">
    <author>edb259c0e0038f38bb200bc20c8cbf7e</author>
    <time>04:43</time>
    <text>we're not at the mall</text>
  </message>
  <message line="2">
    <author>edb259c0e0038f38bb200bc20c8cbf7e</author>
    <time>04:45</time>
    <text>hmm</text>
  </message>
  <message line="3">
    <author>edb259c0e0038f38bb200bc20c8cbf7e</author>
    <time>04:45</time>
    <text>you're gone</text>
  </message>
</conversation>
```

Figure 4.4: A conversation sample in PAN-2012 training dataset.

Statistical Features

The training and test dataset share similar statistical features (Table 4.2). In the training dataset, there are 66,927 chat conversations with over 97,000 different users and only 142 users are sexual predators. After applying preprocessing strategies (see: Chapter 3.2), the percentage of positive samples (conversations involving predators) is reduced to 50% of the original size. The percentage of negative samples (conversation involving victims and regular users) is only reduced by fewer than 20%. There are 60% of the conversations fall into the length range of 0-20 in positive conversations. However, in negative conversations, about 80% of the conversations are shorter than 20 (Table 4.1).

The test dataset contains 155,128 chat conversations with over 218,000 different users and only 254 of them are sexual predators (Table 4.2). The test samples double the training

Table 4.1: Sentence length distribution of the PAN-2012 dataset.

Sentence length	Training dataset		Test dataset	
	Positive	Negative	Positive	Negative
0-20	1142	51362	2189	119545
21-40	182	6245	274	14711
41-60	129	2482	243	5971
61-80	113	1326	255	3163
81-100	110	853	217	2003
>100	291	2576	506	5817

samples in size. The distribution of the sentence length is similar to training dataset.

Table 4.2: Attributes of the PAN-2012 dataset

	Training		Test	
Type	Original	Filtered	Original	Filtered
Positive	2016	1088	3684	1880
Negative	64911	52854	151210	123229
Non-predators	97547	97291	218488	217997
Predators	142	138	254	215

Semantic Features and Behavioral Features

The organizers from PAN-2012 think that the percentage of suspicious conversations (predators involved) should be very low [3]. As people are not willing to share private conversations, they collected a huge number of conversations from Internet Relay Chat (IRC) channel [3]. One of the advantages of IRC is that the user can choose the topic.

4.3.2 IMDB Sentiment Reviews Dataset

Overview

IMDB Large Movie Review Dataset provides 50,000 binary labeled reviews extracted from IMDB for sentiment analysis task [15]. It is introduced to measure the performance of sentence vectors. In this dataset, highly polar movie reviews, with a rate score lower than

4 or higher than 7 on a scale of 10, are split evenly into 25,000 training samples and 25,000 test samples. The overall distribution of labels is balanced.

Statistical Features

The distribution of reviews' length is shown below (Table 4.3). The IMDB dataset is introduced to examine the performance of sentence vectors. Therefore, the length of reviews is analyzed and about 90% of the reviews are fewer than 60 words. More than 50% of the sentences are fewer than 20 words. The size of training datasets and test datasets is balanced.

Table 4.3: Sentence length distribution of IMDB dataset.

Sentence length	Training dataset		Test dataset	
	Positive	Negative	Positive	Negative
0-20	6807	6943	7051	6942
21-40	3832	3941	3775	3978
41-60	1133	1032	1043	1028
61-80	450	368	371	361
81-100	193	135	172	132
>100	85	81	88	59

Difference between PAN-2012 and IMDB

The distribution of length of conversations is different between PAN-2012 and IMDB datasets. The PAN-2012 dataset is composed with short conversations (Table 4.3). Most of IMDB reviews have a medium length (20-60 words). There are seldom cyber slangs in IMDB dataset. The conversations in PAN-2012 are dialogue-based and in IMDB dataset are based on reviews. On the other hand, review-based content sometimes includes both positive and negative sentiment information. It is easy to reach over-fit during the training procedure.

4.4 Experimental Setup

4.4.1 Language Model

The LSTM-RNN language model has four layers. According to the methodologies mentioned in Chapter 3, one embedding layer, two LSTM-RNN layers with 200 units and 50 (35) time steps as well as a Softmax layer are implemented on Tensorflow framework. There are two versions of language model as the average length of reviews (conversations) is different. The time step for the PAN-2012 is 35, and is 50 for IMDB sentiment dataset, which is longer than SCD's as the average number of sentences per input of IMDB dataset is larger (see Table 4.1 and Table 4.3).

4.4.2 Suspicious Conversation Detection

For the SCD task, LSTM-RNN language model is trained with the architecture shown in (Figure 3.8). The sentence vectors are the last hidden state of LSTM-RNN language model. Each conversation being represented by a group of sentence vectors is fed into a new LSTM-RNN binary classifier. The SCD classifier is implemented on Keras framework, it has a similar structure as LSTM-RNN language model, except that the classifier replaces Softmax layer with sigmoid layer (Figure 4.5).

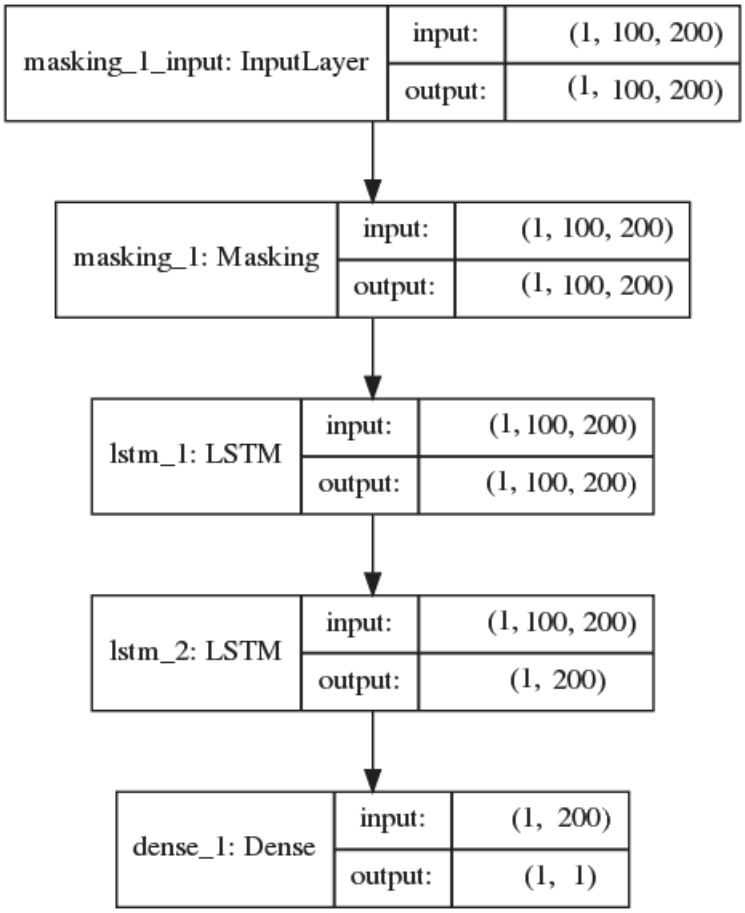

Figure 4.5: Structure and configurations of the SCD classifier.

4.4.3 Predators Identification

The predators identification classifier i.e., the participant classifier has four layers. The input layer with a maximum input length of 500 words is the first layer. The second layer is the embedding layer. The number of hidden units of this layer is 50. The third layer is the average pooling layer with 50 hidden units. The number of 50 is the default parameter of Fasttext [24]. The last layer is the Softmax output layer with three units. Those three units are the sentiment scores (Figure. 4.6).

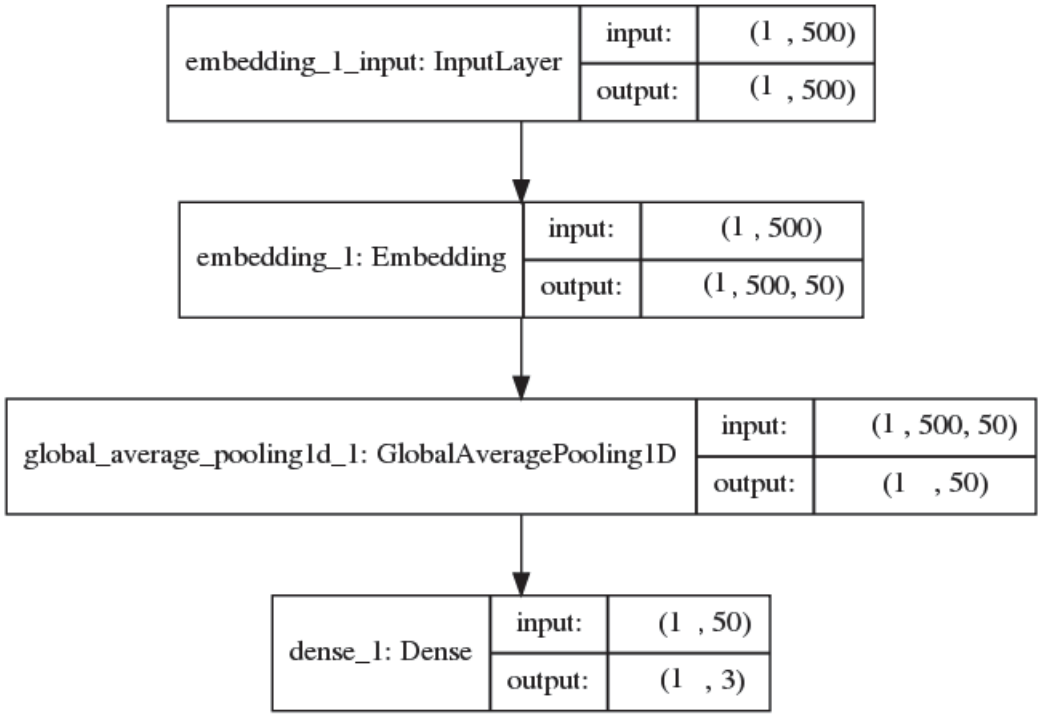

Figure 4.6: Structure and configurations of predator classifier.

4.4.4 IMDB Sentiment Task

The IMDB sentiment task is introduced for evaluating the classification performance of sentence vectors. This task shares the same structure as the SCD task. As most of the reviews fewer than 70 words (Table 4.3), the time step of LSTM-RNN is 70. The number of hidden unit is 200 (Figure. 4.7). The number of 200 is referred from [22]. In the next chapter, the results of experiment are analyzed.

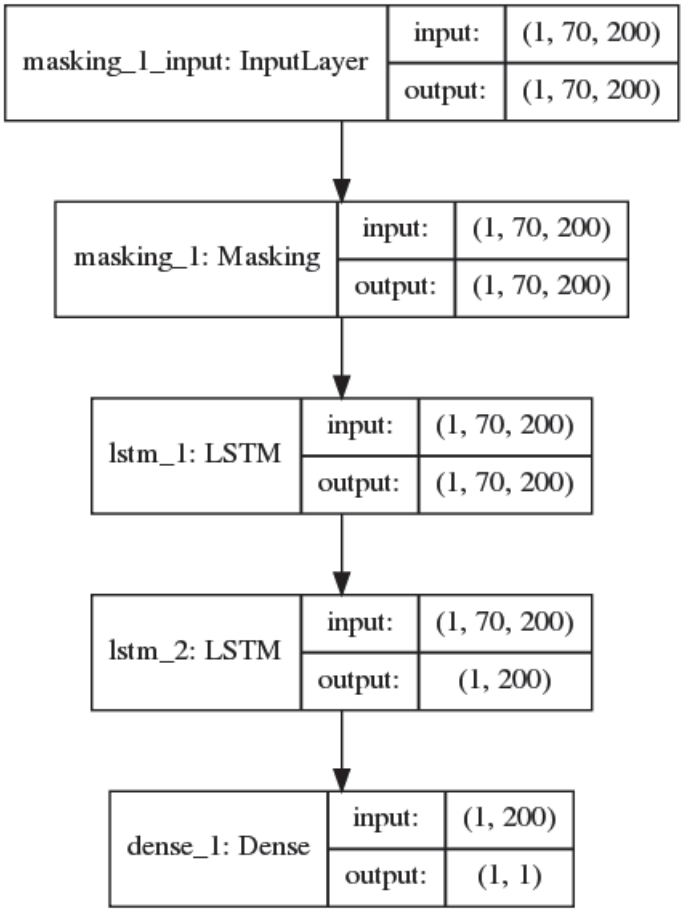

Figure 4.7: Structure and configurations of IMDB sentiment classifier.

Chapter 5

Results

5.1 Suspicious Conversation Detection

The result of suspicious conversation detection task is shown in Figure 5.1 and in Table 5.1. It is obvious that the best test result is obtained at epoch 5. The accuracy of sentence-vector model is 99.43%, exceeding the accuracy of 98.83% with the SVM obtained by [4] (Table 5.2). After 10 iterations, the performance of the suspicious conversation classifier becomes stable.

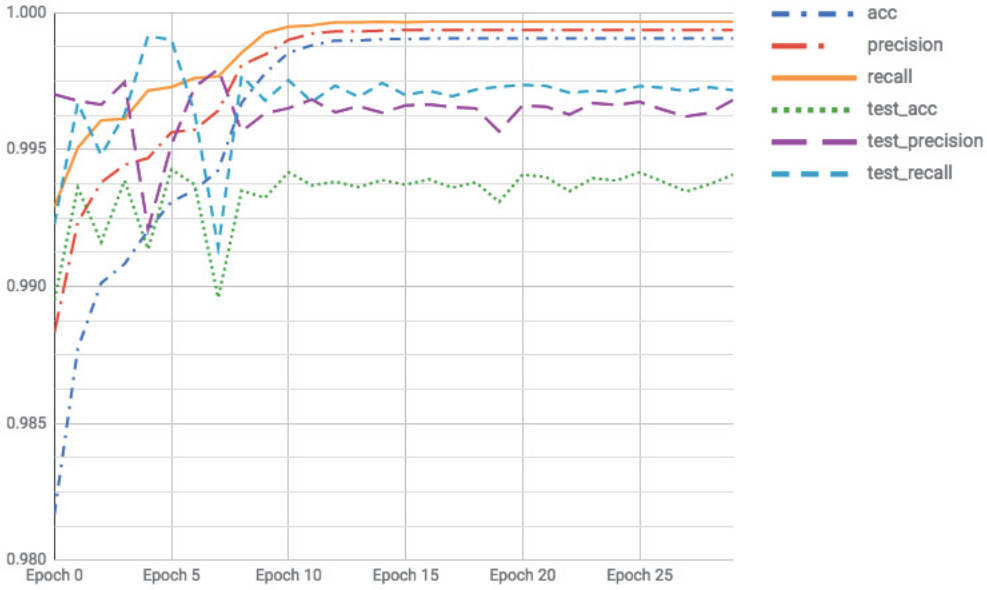

Figure 5.1: Performance of the SCD classifier.

Table 5.1: Best result at Epoch 5.

	Acc	Precision	Recall	F-1
Training	99.31%	99.56%	99.73%	99.65%
Test	**99.43%**	99.55%	99.87%	99.71%

Table 5.2: Performance of No.1 in PAN-2012 competition[4]

Algorithm	Weighting	Accuracy	F-1
SVM	binary	0.9848	0.9361
SVM	tf-idf	0.9883	0.9516
NN	binary	0.9874	0.9464
NN	tf-idf	0.9825	0.9254

5.2 Sexual Predator Identification

In the sexual predator identification task, the performance of Fasttext model is very stable (Figure 5.2). The result which exceed the best one (Table 5.5) from the official rank is shown in Table 5.3. The sentiment score is generated from Softmax layer of the Fasttext model. The scores of the same participants in different conversations are averaged. After

averaging, the classifier finds all of the predators in filtered test dataset (Table 5.4). The result (Table 5.6) shows that all predators have a very high sentiment score compared to non-predators. The complete score table is listed in the appendix.

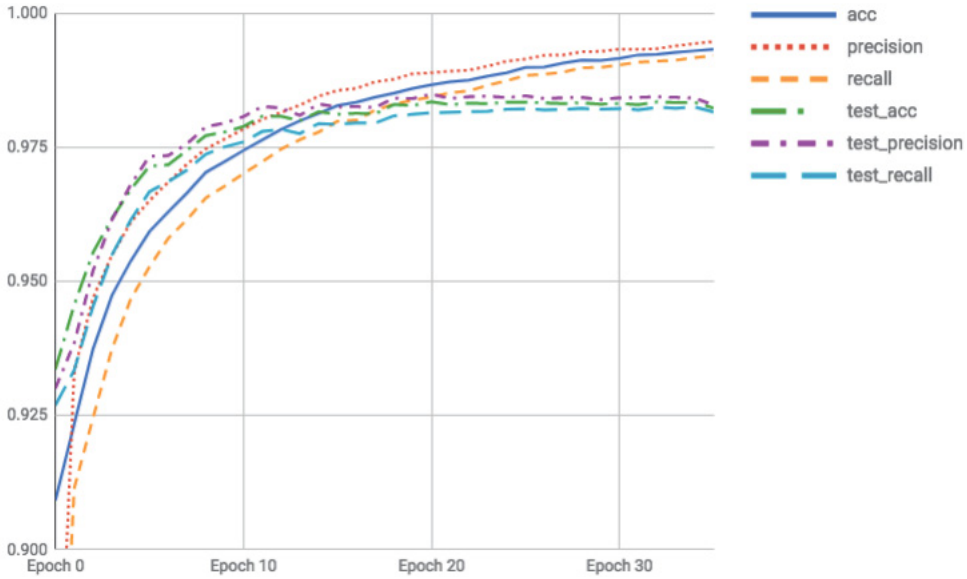

Figure 5.2: Performance of SPI classifier.

Table 5.3: Best result of SPI classifier.

	Accuracy	Precision	Recall
Training	99.00%	99.15%	98.85%
Test	98.35%	98.47%	98.22%

Table 5.4: Result after applying Sentiment Score.

Retrieved documents	Relevant documents	Accuracy	Precision	Recall	F-1	F-0.5
206	206	100.00%	100.00%	81.10%	0.8956	0.9555

5.3 IMDB Sentiment Reviews

The language model is built with the same method as SCD and the test perplexity is 126.903, which is hardly satisfactory. The perplexity is significantly higher than SCD's. The performance of IMDB sentiment classifier is shown in Figure 5.3. In the training chart,

Table 5.5: Official rank released by PAN lab [3]. The table reports the evaluation of all the runs submitted ordered by value of F score with _ = 0.5. Runs with ranking number are the ones used for official evaluation. RET. = Retrieved documents, REL. = Relevant document retrieved. P = Precision. R = Recall

Participant run	RETR.	REL.	P	R	F_=1	F_=0.5	Rank
villatorotello-run...[4]	204	200	0.9804	0.7874	0.8734	0.9346	1
snider12-run...[3]	186	183	0.9839	0.7205	0.8318	0.9168	2
parapar12...[8]	181	170	0.9392	0.6693	0.7816	0.8691	3
morris12...[5]	159	154	0.9686	0.6063	0.7458	0.8652	4
eriksson12...[9]	265	227	0.8566	0.8937	0.8748	0.8638	5

Table 5.6: Part of the predators and victims with sentiment score.

Index	Predators	Score	Victims	Score
1	004ed4354a09e2c33117335adb24e333	0.97	9eb10acea3e6eb0da7b37acef57a5097	0.03
2	00851429b21722a4d62f63a328c601ca	0.99	ef8fbb24e05c1d18efc7a75a812da6ed	0.02
3	00d36f64d208c95eeb70af477dfb368a	1.00	980ffbae20a666d965bb171413352750	0.01
4	00fe41de80eb7527c81f7915ab5a6479	0.67	001744005608bb20b997db6d8cabb3a9	0.27
5	013dab612d37dc4e2cce87da5239f537	0.92	b3d822f188649acd6401e8289193184a	0.02
6	0258ce41335ad5dca6c1a78cfeeaf0c3	0.93	2c7b43a489ac39d98fa8c0fac35fc506	0.07
7	0317e4305bb48c86727d9b72f720885e	0.76	061a7cb44143c259d3edfb892d9197cc	0.00
8	0cd9be63d9dbf98aaa03362487f4f2cb	0.92	b6fe182274453b707870b16e5d2ad562	0.04
9	0d3e4cee17e1ffaa7d33d252a4175ed9	0.98	0f49dfaaae5336ece90f22ae2c9f7585	0.10
10	0f34da674b672786397ec900138159df	0.98	961fc4821bc79eeb9991d658b181ae35	0.01
11	0fa23138f5b29012c1b55c5a54c072db	0.84	7d41c88321223a0598037d0bf8b229da	0.16
12	116396538c595a129c60228838d9fcbe	0.86	71ed74330fc613418796687c48f74ce9	0.06
13	11fa8ca63591175e5c17a8f6874b422d	0.80	980ffbae20a666d965bb171413352750	0.10
14	13396578cb61bd3ccd2b13c1650be421	0.93	75df7004b5b3fa600bc4482de4519bfd	0.08

it is evident that the model starts to overfit after 50 iterations. The training accuracy keeps increasing whilst the test accuracy and test precision show a downward trend. It means that the ability of capturing features among IMDB reviews is weaker than the ability to capture features in PAN2012.

Compared to the training result, the test result of the sentence-vector model on IMDB dataset indicates that the performance of this model is unstable. Although the sentence-vector model has an accuracy of 83.2% only (Table 5.7) which is lower than others (Table 5.8), the sentence-vector reduces the length of the inputs and accelerates the training and test speed.

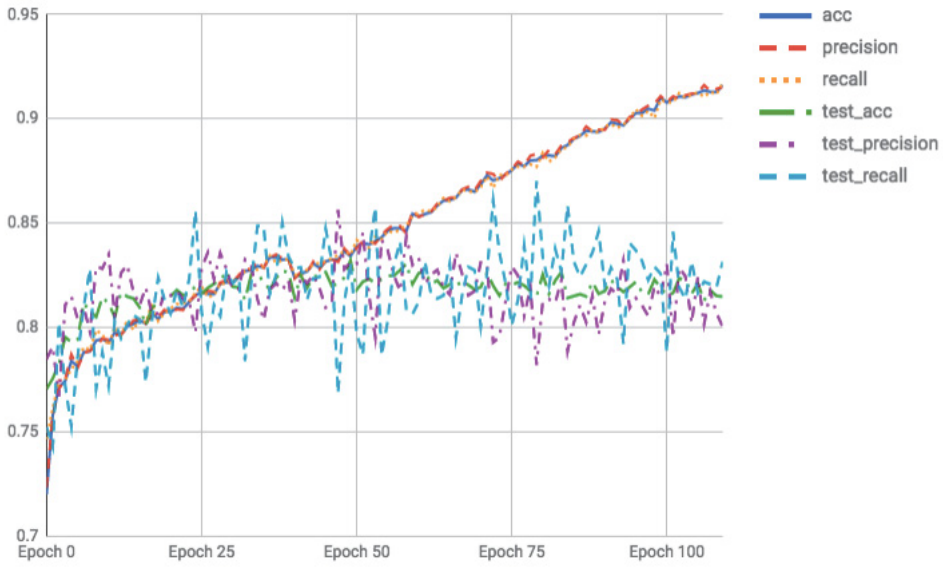

Figure 5.3: Performance of IMDB sentiment classifier.

Table 5.7: Training and test performance on IMDB dataset.

	Accuracy	Precision	Recall
Training	83.43%	83.46%	83.49%
Test	83.23%	83.13%	83.13%

Table 5.8: Comparison on the IMDB sentiment task.

Model	Training Accuracy	Test Accuracy
NBSVM-bi (Wang and Manning [23])		0.912
Paragraph Vector (Le and Mikolov [40])		0.927
Paragraph Vector (2-layer MLP)	0.971	0.945
Sentence Vector + LSTM-RNN	0.834	**0.832**

Chapter 6

Conclusions and Future Research

6.1 Conclusions

This thesis presents a novel method to identify sexual predators. The sentiment score from Softmax layer outputs is crucial in the final identification step. The approach of taking LSTM-RNN last hidden state as sentence vectors is highly efficient as long conversations are shortened by sentence vectors. The higher perplexity is not good enough to represent the sentences. That is the reason sentence-vectors-based classifier does not work well on IMDB dataset.

The sentence vector shows its potential of representing sentence via LSTM-RNN language model. However, during the research of sentence vectors, the perplexities of the language model obtained from PAN-2012 and IMDB sentiment reviews are quite different. Obviously, it is hard for datasets with rich-meaning texts to get lower perplexity. The reason is that the perplexity is correlated to the number of potential candidate words. A higher perplexity means poorer representation ability of the language model. The difference in perplexity may well explain the poor performance of the classifier on IMDB dataset.

LSTM-RNN can learn the dependencies among a group of words. As it is a significantly deeper neural network, the training speed may be decreased if the dataset is very large. Reducing the complexity of LSTM-RNN's architecture can accelerate the training procedure.

However, such reduction will limit the learning ability of the neural network. After applying sentence vectors, the length of each conversation or review is much shorter comparing to the use of word embedding. In this way, the classifier gains efficiency and accuracy without compromising the complexity.

Predators' sentiment scores are higher than the victims', therefore we can conclude that the hypothesis stated in Section 1.2 is supported by the sentiment score results summarized in Appendix A. Without averaging the score of participants, the result of predators identification is similar to Inches and Crestani [3]. Unlike Inches and Crestani [3] who filtered many conversations (only about 10% of samples kept for training), in our work, there are 80% of data for training. Therefore, although [3] and our work are equally accurate, the total number of suspicious conversations in our work is large. The average method excluded the influence of extreme samples.

6.2 Future Work

In the future, other language models will be explored to reduce the perplexity to see if the accuracy will be enhanced. In the meantime, attention mechanism [41] in neural network can also be introduced to detect keywords in predators' conversations. Potential research directions, which may bring further improvement, are listed as below:

(1) Character-level sentence vector. Recently, the character-level recurrent neural language model shows its impressive ability in sentiment analysis [42]. The character-level language model has inherent advantage in terms of speed, as the vocabulary is very small (usually less than 50 for English corpus). On the contrary, the vocabulary size of word-level language model is usually large than 100,000.

(2) As this research focuses on predators identification, our work does not apply more complicated models e.g., seq2seq model and bidirectional RNN etc. Such models will be useful if combined with sentence vectors.

(3) Another potential direction is the application of a tree-like structure to classification

work. More specifically, the basic idea is letting the neural network learn features in different time scales by using a composite structure (Figure 6.1). Human reading inspires this idea. Human always focuses on keywords and surrounding texts when skimming. The structure below is trying to make the neural network to learn from the surrounding texts by regulating time steps. From the current experiments, the accuracy on IMDB sentiment reviews is 89%. This method shows its potential.

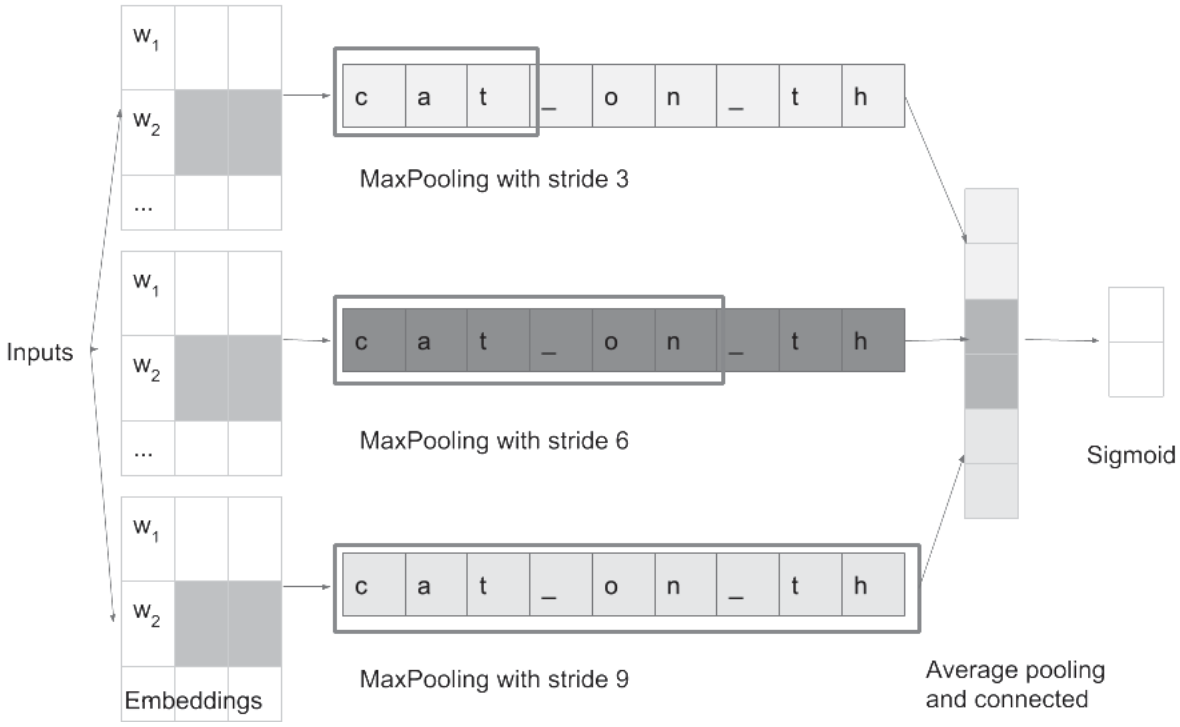

Figure 6.1: A tree-like structure for classification work.

References

[1] J. Wolak *et al.*, "Online predators: Myth versus reality," English, *New England Journal of Public Policy*, vol. 25, no. 1, p. 1, Oct. 2013. [Online]. Available: https://search.proquest.com/docview/1499593521.

[2] *Us attorney leaves lasting legacy in the prosecution of internet crimes against children*, English, May 2011. [Online]. Available: https://search.proquest.com/docview/868906453.

[3] G. Inches and F. Crestani, "Overview of the international sexual predator identification competition at pan-2012," in *CLEF (Online Working Notes/Labs/Workshop)*, vol. 30, 2012.

[4] E. Villatoro-Tello *et al.*, "A two-step approach for effective detection of misbehaving users in chats," in *CLEF (Online Working Notes/Labs/Workshop)*, vol. 12, 2012.

[5] C. Morris and G. Hirst, "Identifying sexual predators by svm classification with lexical and behavioral features," in *CLEF (Online Working Notes/Labs/Workshop)*, vol. 12, 2012, p. 29.

[6] C. Peersman *et al.*, "Conversation level constraints on pedophile detection in chat rooms," in *CLEF (Online Working Notes/Labs/Workshop)*, vol. 12, 2012.

[7] M. Popescu and C. Grozea, "Kernel methods and string kernels for authorship analysis," in *CLEF (Online Working Notes/Labs/Workshop)*, vol. 12, 2012.

[8] J. Parapar *et al.*, "A learning-based approach for the identification of sexual predators in chat logs," in *CLEF (Online Working Notes/Labs/Workshop)*, vol. 12, 2012.

[9] G. Eriksson and J. Karlgren, "Features for modelling characteristics of conversations," in *CLEF (Online Working Notes/Labs/Workshop)*, vol. 12, 2012.

[10] L. Gillam and A. Vartapetiance, "Quite simple approaches for authorship attribution, intrinsic plagiarism detection and sexual predator identification," *LNCS*, 2012.

[11] D. Vilario *et al.*, "Information retrieval and classification based approaches for the sexual predator identification,"

[12] J. M. G. Hidalgo and A. A. C. Daz, "Combining predation heuristics and chat-like features in sexual predator identification," in *CLEF (Online Working Notes/Labs/-Workshop)*, vol. 12, 2012.

[13] I.-S. Kang *et al.*, "Ir-based k-nearest neighbor approach for identifying abnormal chat users," in *CLEF (Online Working Notes/Labs/Workshop)*, vol. 12, 2012.

[14] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[15] A. L. Maas *et al.*, "Learning word vectors for sentiment analysis," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, Association for Computational Linguistics, 2011, pp. 142–150.

[16] Y. Bengio *et al.*, "A neural probabilistic language model," *Journal of machine learning research*, vol. 3, no. Feb, pp. 1137–1155, 2003.

[17] T. Mikolov *et al.*, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[18] T. Mikolov *et al.*, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[19] P.-S. Huang *et al.*, "Learning deep structured semantic models for web search using clickthrough data," in *Proceedings of the 22nd ACM international conference on Conference on information knowledge management*, ACM, 2013, pp. 2333–2338.

[20] Yelp, *Yelp dataset*. [Online]. Available: https://www.yelp.com/dataset/challenge.

[21] A. Severyn and A. Moschitti, "On the automatic learning of sentiment lexicons," in *HLT-NAACL*, 2015, pp. 1397–1402.

[22] J. Hong and M. Fang, "Sentiment analysis with deeply learned distributed representations of variable length texts," Tech. Rep., 2015. [Online]. Available: http://cs224d.stanford.edu/reports_2015.html.

[23] S. Wang and C. D. Manning, "Baselines and bigrams: Simple, good sentiment and topic classification," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, Association for Computational Linguistics, 2012, pp. 90–94.

[24] A. Joulin *et al.*, "Bag of tricks for efficient text classification," *arXiv preprint arXiv:1607.01759*, 2016.

[25] Y. LeCun *et al.*, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.

[26] T. Mikolov, "Statistical language models based on neural networks," *Presentation at Google, Mountain View, 2nd April*, 2012.

[27] S. Hochreiter, "Untersuchungen zu dynamischen neuronalen netzen," *Diploma, Technische Universitt Mnchen*, vol. 91, 1991.

[28] R. Pascanu *et al.*, "On the difficulty of training recurrent neural networks," in *International Conference on Machine Learning*, 2013, pp. 1310–1318.

[29] A. Karpathy, "The unreasonable effectiveness of recurrent neural networks," May 2015. [Online]. Available: http://karpathy.github.io/2015/05/21/rnn-effectiveness/.

[30] C. E. Shannon, "A mathematical theory of communication," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 5, no. 1, pp. 3–55, 2001.

[31] I. Sutskever *et al.*, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[32] D. E. Rumelhart *et al.*, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, p. 533, 1986.

[33] T. Luong *et al.*, "Neural machine translation (seq2seq) tutorial," 2017. [Online]. Available: https://github.com/tensorflow/nmt.

[34] M. Chen, "Efficient vector representation for documents through corruption," *arXiv preprint arXiv:1707.02377*, 2017.

[35] H. Zhao *et al.*, "Self-adaptive hierarchical sentence model," in *IJCAI*, 2015, pp. 4069–4076.

[36] T. Yu *et al.*, "Leveraging sparse and dense feature combinations for sentiment classification," *arXiv preprint arXiv:1708.03940*, 2017.

[37] M. Ebrahimi, "Automatic identification of online predators in chat logs by anomaly detection and deep learning," PhD thesis, Concordia University, 2016.

[38] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *The Journal of physiology*, vol. 160, no. 1, pp. 106–154, 1962.

[39] Y. LeCun *et al.*, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[40] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *International Conference on Machine Learning*, 2014, pp. 1188–1196.

[41] D. Bahdanau *et al.*, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[42] A. Radford *et al.*, "Learning to generate reviews and discovering sentiment," *arXiv preprint arXiv:1704.01444*, 2017.

Appendix A

The Sentiment Scores

Table A.1: List of sentiment score of predators and victims.

Index	Predators	Score	Victims	Score
1	004ed4354a09e2c33117335adb24e333	0.97	9eb10acea3e6eb0da7b37acef57a5097	0.03
2	00851429b21722a4d62f63a328c601ca	0.99	ef8fbb24e05c1d18efc7a75a812da6ed	0.02
3	00d36f64d208c95eeb70af477dfb368a	1.00	980ffbae20a666d965bb171413352750	0.01
4	00fe41de80eb7527c81f7915ab5a6479	0.67	001744005608bb20b997db6d8cabb3a9	0.27
5	013dab612d37dc4e2cce87da5239f537	0.92	b3d822f188649acd6401e8289193184a	0.02
6	0258ce41335ad5dca6c1a78cfeeaf0c3	0.93	2c7b43a489ac39d98fa8c0fac35fc506	0.07
7	0317e4305bb48c86727d9b72f720885e	0.76	061a7cb44143c259d3edfb892d9197cc	0.00
8	0cd9be63d9dbf98aaa03362487f4f2cb	0.92	b6fe182274453b707870b16e5d2ad562	0.04
9	0d3e4cee17e1ffaa7d33d252a4175ed9	0.98	0f49dfaaae5336ece90f22ae2c9f7585	0.10
10	0f34da674b672786397ec900138159df	0.98	961fc4821bc79eeb9991d658b181ae35	0.01
11	0fa23138f5b29012c1b55c5a54c072db	0.84	7d41c88321223a0598037d0bf8b229da	0.16
12	116396538c595a129c60228838d9fcbe	0.86	71ed74330fc613418796687c48f74ce9	0.06
13	11fa8ca63591175e5c17a8f6874b422d	0.80	980ffbae20a666d965bb171413352750	0.10
14	13396578cb61bd3ccd2b13c1650be421	0.93	75df7004b5b3fa600bc4482de4519bfd	0.08
15	135e36c23b4583cded18d002aa5ae99e	0.97	d4ea20cbe75fff6e1a84882d38deb72a	0.03
16	13f79bca695765c026ec0bac88057ed3	0.77	40625103c6a477d392455b7c72f3f582	0.05
17	154e01cc3d25b6e2f8c35a7570ff9eab	1.00	e5ae90703304332214a41c01b6932953	0.10
18	1643a1e356c2529d13182721b98cf7c3	0.99	1bd4a96355c82d6ae72e343525e0f532	0.05
19	1833c12eb28c2e7df1a70a562b577866	0.94	f1fbe0a5fe54d45d2bdb5da657c77ec7	0.01
20	1caa782b9543e4306251525f6eb627a3	1.00	815de8eb13c20cc0ff35384abac6cad2	0.00

21	1ce293a325f7dd80acf15df1ff91c9f3	0.98	57077572896c8c0bc695424a87b94ff6	0.00
22	1e0a3122f95e35a5c8f3b8d58b5bab6b	0.97	5c5b806fbd1826340209616ddb9ed767	0.02
23	1f1a75629a91fde59452519a827524e6	0.99	30dc9744917f95d1254fc34bb05b3210	0.00
24	1fca6116ec740de552045f66b6651c56	1.00	82a42c3ebfd83735f1aaf93cf8adf7e3	0.00
25	2057ae915271cbda54e23ab90ff2a901	0.98	a6d5de9b5e00b181fc3be41fcd94953b	0.50
26	214ec93027486ebd3f24fdec660d616e	0.88	ffeee55c53e9669f016238c0b61c0313	0.03
27	2192d364fbb5402f1d404db42edadbe4	1.00	961fc4821bc79eeb9991d658b181ae35	0.00
28	22f6e583a3bd9165ab2fbba58d0b8ee4	1.00	220840d2c4fda35d80b9e3855263d7b9	0.00
29	23b5a168f3f9a1ced4cb1d9d6d04b765	0.99	f5231a4bd4c29f4d613b450ba123162e	0.00
30	246ec88472d59ba9422025039825d62f	0.88	e372ad16b9fbfee4d9cdeff523cc06a9	0.07
31	2856ad87904c49564c17c86d3a58ff21	1.00	9498cfff62ca3992ca606d9f3744891a	0.14
32	29d5ef89c71b7632c0b1f995c21eaf0a	0.88	815de8eb13c20cc0ff35384abac6cad2	0.21
33	2a25997e28333954d0d873716768ec34	0.90	3ecc23e95e70d0e24a74f811c3ab08f2	0.04
34	2d1e0e4088a65d61728e8b979636efe7	0.91	459f7bbcf73a8dcdb5222825291cd04b	0.35
35	2e0d170f2addfb0048f9424a2daa5a73	1.00	f748dcbd3f115eca664eca55c32dbfc2	0.01
36	2eba3cbb71e6ea5af3ede4d7b898f99d	1.00	3e4a51f98397c7b41ea8eafa7d0f6a12	0.00
37	30b317a34a320b9ba5e23f5bca538b5f	1.00	980ffbae20a666d965bb171413352750	0.00
38	31e98ef0d10792e3e35fa231164e9199	1.00	4f1ae0e7c29bee0792fb1b5f2843984c	0.02
39	340e382d71025a15ec052d0ad9393ef6	1.00	94f8a0e034ca9bea4912808a22482d3a	0.00
40	3485817703e8f1cbd2efa9ea8ccf33fe	1.00	3e4a51f98397c7b41ea8eafa7d0f6a12	0.02
41	35a3f9f9bd68e3d37bef8bae91b9d956	0.93	b67171b8f016f8384f5fc86a899ae354	0.00
42	398cab8240d8a5a9f5f201115c0337c6	1.00	8f47cc183d1bade83ba1ef5debb83eaf	0.59
43	3b134d48a7081997ca2f5a1246756362	0.85	6be1cba45fefdf24824c32dafb09cdc1	0.01
44	3d6b07242981c46ce8e39a53e9a2aeb2	1.00	82a42c3ebfd83735f1aaf93cf8adf7e3	0.00
45	3e6bb10fbe0e8349709c65b56519c034	0.98	e03aa9707bd13f180c517ae1a47e9da2	0.09
46	3f549ccaf0d22e91d26c65f43bed2bf0	0.87	5c91acf8c2808994d9681a0ee5d28ea3	0.66
47	417ac946ab237e5d57c64a0ab6d8b34c	0.89	39cf15419c600a8ba4779ec994b98577	0.01
48	41e9fb56564ec12597cbb36786d47f6d	0.99	ac23b9b6adc8307577177476343d4f9b	0.04
49	426d0b70843d16c615f6e754c5b718d1	0.92	36b5f84e4baff27948d4a21f91b7226b	0.40
50	427af0d4da055d2aedbd0cae1dd9cdcd	0.89	82a42c3ebfd83735f1aaf93cf8adf7e3	0.00
51	43e7ccd4248879279780e6bfeb0b733c	1.00	457b65aa8f082907ee645289708d77f6	0.00
52	44db68c3dae6f5923c44af555c28c02e	0.98	bdbd79de6f1ec029467d0977a12a69a0	0.00
53	46772080e6727b055bbfe7b50211d4cc	1.00	7a3f4b2ed72d412c16a349e72ace8284	0.00
54	46a159ebabbde8fde4ff78a38e920508	1.00	7a23d93b4f1799cd39c11648b52f601a	0.04

55	471b5cd792ff3cc22bda06bd170aab57	0.92	406508697627cbadba540669704114ac	0.04
56	4944564d67ca6d1f00e479f444401732	0.96	e01a627489e8ec6f168ded8f20fe7bef	0.37
57	494c93b9b7eee9d6c576891426e509c5	1.00	961fc4821bc79eeb9991d658b181ae35	0.46
58	4a9332d7466b98d11c23e4447b26460a	0.95	b6fe182274453b707870b16e5d2ad562	0.06
59	4b4887e2dc505b40902905b122fde2cc	0.99	6a00303d6ed74aa525c3ff12914c5730	0.05
60	4cdf39d72c65b4655d6541769a777ad6	1.00	0f33af8bf285dd95c329bf725744420d	0.00
61	4f9a0a4c13aa247f7824f4b59a868b57	1.00	220840d2c4fda35d80b9e3855263d7b9	0.00
62	505d828e96b70fd4c5205428265d8492	0.93	b6fe182274453b707870b16e5d2ad562	0.00
63	519dae81bfe020e6be9d1ab535da59a6	1.00	ef2c464312e0ff5bcc8d709061b52952	0.09
64	54b595f1920b5b1988e907ea693303b4	0.80	a5cfd0f80588fca7f8633b6957876c7a	0.09
65	5778ac8dcffe9bbacf3c0415fdb362d6	0.98	565027ae1da1cfbcc15ac20ce04390a5	0.02
66	5882abb1a367ec8bf440c5c5a0f1b5a4	0.74	8dcf8b6239f3362f9718d0b9f4a1a107	0.20
67	5883af7738972312e1284e0c489ea183	0.89	82a42c3ebfd83735f1aaf93cf8adf7e3	0.02
68	58c43e0e5588dd90891495bad7827045	1.00	b027fba4e06922a744d8b08aea6fa5c5	0.12
69	5914020f36aff3c419faf247b6f258e0	0.93	1cd17fce23a3889173b1589e9a7b28cc	0.06
70	5929d79f59895b6093f3dd9663f62a9e	0.93	8ea24013de621efe3b3465d360c45c3d	0.19
71	5a304ea8f1a9aa03d68e15b1eabb9fad	1.00	d5053979f47f33305af03c4e723e77d3	0.03
72	5bb7e166f2c2896bb004f3480e02ae46	0.99	82a42c3ebfd83735f1aaf93cf8adf7e3	0.00
73	5cdb7dd92d140d9e5a08dfc19ae27d29	0.88	9df9d18534885d4387f2997aefa941cf	0.11
74	5ebba19a77c082af105971d6c125b662	0.87	961fc4821bc79eeb9991d658b181ae35	0.03
75	60550136cc7137c7bce945ae1fd82967	0.89	fb1d96b82911c7e1561bcd24eedd8bda	0.04
76	622e2124832a7a5acf9a2b2555c25784	0.95	5c5b806fbd1826340209616ddb9ed767	0.53
77	62760245391c6d56088d814bea04baad	0.92	871228b3ecce54d587815f064069fe94	0.03
78	62760245391c6d56088d814bea04baad	0.93	9d2f75377ac0ab991d40c91fd27e52fd	0.16
79	6345bce4ee032d568eca99747974bdf1	1.00	85fbc933668116f84a12127cd54dcad8	0.00
80	65ecc13d728602045d1d84b77e0474ff	0.94	6b042182b54c9c5f366b661b9fdca2bf	0.00
81	66b8348255da2fb9e27a4c6c6cb439ce	0.91	7f0bea2dff091e94609462ae3b3c851c	0.03
82	676fa29dfde8be4cf9b66fa6c84f9712	0.99	4ed1b85231ced8f57fc28d8062b9773f	0.00
83	68e88c75d75c09f0d45d6e9a75e9ddfb	0.81	f7ba507db5b5b1150eabf5707f0334dd	0.02
84	699b069fef306f158bfe52cf05113b36	0.94	b6fe182274453b707870b16e5d2ad562	0.05
85	6a276896d461828a3f702844bd82988f	0.95	e102b937f5fdf21d398d6ab59ca7cb14	0.23
86	6af7c6e2cc70fef59333790cd8e2b18f	1.00	c664b9f7df7ffc275d0b9aa716ac010a	0.68
87	6ef9f9649aa87afdb5cdbeecee5e2b52	1.00	5c91acf8c2808994d9681a0ee5d28ea3	0.00
88	72e2bd2a69d91902caa26e8702f01c5a	1.00	406508697627cbadba540669704114ac	0.01

89	732fc6ff0b18799fc5fca4dedacc53eb	0.92	3d3760cef49636d3746be30522b4897d	0.14
90	7365be0fc06896693916f0f82d5f7f71	0.67	7bf36bdea5e8c3320d31e58a07a516e0	0.00
91	741a79eea7fd536df71ec6722de14765	1.00	e56d1b90b826ed4457141819a26ad068	0.01
92	743d52489f870e18bbd8b5b7b9b0f765	0.83	8c4078d55ba07096949e82f0993a423b	0.01
93	77fc3343794c5216a1cf41b742019a01	0.78	6a37e1546ace47686185ac3fab4f99b4	0.00
94	7a98f41fc8de58902662208997f6bf0d	0.99	8c5daf40c7da181aaaae27f2f946fc32	0.01
95	7b0c80c33906e696cf4ed37bb244730f	1.00	931ec03eeee8c0899a03830bc14879fb	0.68
96	7bb9da10e9230e70080c12f53f294186	0.77	31e05d07bc6afef6ea044c9471b52437	0.20
97	8164381b4ae95713c7266cba00fec1df	0.95	b654cfec344468ab6f8116f55e8f0524	0.12
98	828181f3b2267abbb73fe963db405799	0.82	482bf4b451d320b2e16f61d535ae89c6	0.00
99	8337ec42f09d2dc8798fb9e0c49f4adb	0.98	adaf167b95b79198535d0902f7a31d55	0.00
100	84fb828731f4e234c54c82158127e73e	0.81	e03aa9707bd13f180c517ae1a47e9da2	0.06
101	8596c082ca02cf3d2e40682389f76a47	0.97	4423d67017f371bb5b7218053f06def1	0.01
102	863944481b27174408f90548a3ad21da	1.00	931ec03eeee8c0899a03830bc14879fb	0.00
103	86acb75ad942a8df784694ad33c83068	0.79	a62c0ff13833a0371552a56a071fbd59	0.10
104	86acb75ad942a8df784694ad33c83068	0.82	64add4e6ed448cd3f56be9adf09a464f	0.04
105	879421089f4a256e5677c3ae5ff1d007	0.74	5c5b806fbd1826340209616ddb9ed767	0.00
106	87eb0510ab6472e60cd0927b83efd695	0.95	e102b937f5fdf21d398d6ab59ca7cb14	0.01
107	8911f2a3c2dff01a285992ce7d8ffdb7	0.83	c025a40898e66796588a7ca54c5506e4	0.13
108	89393c5f8206aec3e70d0acba7ae0bff	0.96	cc63b734fe4cd8cc45e26383c06c3a45	0.11
109	89919b2006145973148fbeddb3b2c8b1	0.56	0a0b5d1b3dc71006100b74ab85f6496e	0.40
110	8bcb850e59d7845b03be61c9af071a7a	0.80	933e78953b20e38832e9cf552f232b14	0.03
111	8f01f3b4d2a831348fab53b14554f647	0.98	79a6de39d7b31b3c62ce1c21c8a10a02	0.32
112	8fa35fb5fdb736bea000e1723d64f888	0.85	961fc4821bc79eeb9991d658b181ae35	0.15
113	91d438bf311a18ab6d1c321f26a18915	0.92	e54693d48ef94e967f29890c1cfb072f	0.00
114	92694694425a2c90fb417eabc2e8c9f5	1.00	0da31bd56f003f3466804bdc021c1e7a	0.08
115	941dd80263686adb071df2172badd426	0.94	8956fe7bf020de2f0e9593443b3d9d1e	0.14
116	949d060631cd32c4b9029eaf95f3414d	0.79	4e1de5cfd5b7472a999fa538b53160c4	0.05
117	9577d9bd8723a3fa3574b7f6b4ce496f	0.86	09baa477b045fd5abca2867933200de3	0.01
118	9692f883e561c47204ccfd5baa90e74b	1.00	82a42c3ebfd83735f1aaf93cf8adf7e3	0.00
119	981413827a6f886823c1af945aefac80	0.95	ea00941154bf3843b5b146ebcd643c96	0.12
120	991de8460a4282b22bae77887a179a76	0.91	b5eae2ee96c59691b7268449e2459726	0.01
121	9b41c89e139a6247450957dce9d57a61	0.89	7e8fb00f4320922655ffd3a6d2fba1ef	0.00
122	9cb9467c4859b06f79a98d737ada9e3b	0.87	a47eace67ec36839ddb9cd868bdc3a33	0.11

123	9feb1fc1d20ef1c7b1db0eda0983be4f	0.92	5c91acf8c2808994d9681a0ee5d28ea3	0.04
124	a12332f18b35f3717dd7c9ac99b00fd6	0.99	aec3f14fa5d0f642d334640acf8798bc	0.00
125	a19fe0297028decce0232f30cd8cfdee	0.98	c35f0c98fd27e760f04225ca03cfb01d	0.00
126	a1a8e6a5db3374ac3254c9113bde167f	0.80	e03aa9707bd13f180c517ae1a47e9da2	0.04
127	a1e84919f4222f969d4778bf1e8cf8ec	0.99	51f03c9fa010e8a83c2eecd4985712ec	0.00
128	a23878d255460147a5430b03a9c83236	1.00	220840d2c4fda35d80b9e3855263d7b9	0.00
129	a26b8d01e634c3e60b76d786355c250f	0.76	e62deef1e514d6d9f31dfce6e6561a29	0.07
130	a3eb372dfbda110a479bbe38c2e1e13d	0.70	f5231a4bd4c29f4d613b450ba123162e	0.00
131	a5e8d5a6986418cae2faa2c72bb48d57	0.73	5c5b806fbd1826340209616ddb9ed767	0.10
132	a5ee20257eb8318ec9669b2eb5acb56d	0.82	5c91acf8c2808994d9681a0ee5d28ea3	0.37
133	a6fac78fa1190cda784c4c63a93b402b	0.99	48b9682a5513419503df7f6f089c2d5d	0.09
134	a76df474ab62275e0c9bb70a7f126874	1.00	a099e31473be457f656571aab7501232	0.00
135	a84c8ffe8d99ae00ff0801a2dfaae86d	0.78	ea83f356e7a3329c1f9982649127f07b	0.27
136	a8b182fd0d335a0966627c61cd0f25e5	0.95	0cd1a56c42f4b100d423249301ac8eab	0.10
137	a8e6e3985a82dfde8ee95b5f099ec606	0.76	e03aa9707bd13f180c517ae1a47e9da2	0.04
138	aa16e1dfa95805e1e5e477a9789d43b8	0.89	47eeb90302a26cb6a1a795d4ade33f13	0.00
139	ac07079f18fcab57692a57e092678052	0.93	a0d64886c5c1d4b86eed2f6e5f2da228	0.01
140	ad18bf6462a3d1b98678dd38b27d8cf3	0.94	78f700d3bbf54dd64d40f4ea49f66cb3	0.02
141	ad3403c013a364bbde185b702aa5735d	0.90	f52b29edf8e1a7d8e3a2b2e99dbd80e0	0.00
142	aea823db920e3cd17bb2bfb91968b9cc	0.99	4c2e4f195132a9888d10b1753d3d8e9a	0.02
143	aeff16c570c7ae7285ccc23dc3105f20	0.50	7185e3ea274acb228a6e5309c598570e	0.19
144	afa81bc41103f8a693eff3e530e5071d	0.94	6d144ccd29ca4950b3da60164f04a0ab	0.02
145	b08cc9daf730722a8ee4cd9a4a20c179	0.93	5c5b806fbd1826340209616ddb9ed767	0.11
146	b0c51b89e81a656ab852c5bb385a10d2	0.88	f0de9cd1b67f83664dbd1c63858a9660	0.20
147	b16b5830e07a56afbab8214056774670	0.71	18a355f80308b63aea6efff77cfee4df	0.00
148	b18ae7c450091f1f200e896d765cce6d	1.00	c936ffd9889f3eaca95042f95649ba46	0.00
149	b61c86937f29437dba66cae2be8a9734	1.00	001744005608bb20b997db6d8cabb3a9	0.00
150	b62e6a62244b39a8f3e53aa275b642ee	0.98	ea83f356e7a3329c1f9982649127f07b	0.11
151	b667823f42c50b5aee6ff2122f6c1d26	0.95	2981258a000393a121125dc3c17e120e	0.12
152	b6a782084087165405382131b5f28388	1.00	1b94226cc32bb3b6077da2625bc41dc5	0.00
153	b6dcae8d89b7d90d0d1bbc20d57e038b	1.00	0675c0d47157fedeee47d0ebe5cb1c5f	0.01
154	b6ef9895ab9c6b4784c3fa25bbcbed26	0.95	f748dcbd3f115eca664eca55c32dbfc2	0.02
155	b8931a8b614fb54f6051ffc75f39db29	0.93	21db5e5ead0c0943992f22b04f224b5b	0.05
156	b8e3a82f26d3320fa091e31879163cd3	1.00	7728db1e1ab65fe9a3b62919786da322	0.04

157	bdafa86f462af0c18683a184784b0ea7	1.00	711c49aa37a9e2d34a60751534afa955	0.00
158	be26a9249ee21581097e6f87388f108d	1.00	ce5e3455a0f07f9646d46fbffd1845c1	0.01
159	befd81c00fb58b4f4b06e1b70c44ed22	0.98	609fe28735f0f537adab04efe4d4bd74	0.00
160	c05dab2bc28c161ccef69000520dc050	0.93	5c5b806fbd1826340209616ddb9ed767	0.00
161	c3e302119676d8e7ddd7ba5791d2876a	0.91	8c4078d55ba07096949e82f0993a423b	0.00
162	c4155cd04b12e8bf58f693f572cfa5b3	0.96	b6fe182274453b707870b16e5d2ad562	0.02
163	c483a618171c8a6bd691ea7f238f8e02	1.00	a6d5de9b5e00b181fc3be41fcd94953b	0.00
164	c5502c7c9bb5e28508a3e19ec869f6d2	0.99	457b65aa8f082907ee645289708d77f6	0.04
165	c772a30d41ec8c754d9541fffcaf65b6	0.83	e4bad96310d75df2770ecdcd28cc7bb8	0.06
166	c92808e9cfb0834a62a670e613637377	0.89	b7ef10a3deaa560a5d5dcad89a941e41	0.18
167	c938fccbd1690526f6045b28820c0a48	0.85	6e8c876c80a2ce6412b4ea28715c7ca1	0.07
168	c9ead9fa1ac71e6001d63ac5a136f9bf	0.98	2f4986a1329d408089e739afad4d16d6	0.00
169	ca01f49c85d74e87fbaddbb693caad24	0.98	c10c2dfe2fba9d6800341fb16ac29990	0.36
170	cae47192005181a1385ede9804683459	0.88	1aa74fb207bdd9e22fe0c57f4093e34c	0.03
171	cba51b9a845eaaae97c19dfd9ab70bae	1.00	dff059af5730d3e67cde479267f38529	0.00
172	cbfd26b58220285268afb6e6196b3953	0.87	0a0b5d1b3dc71006100b74ab85f6496e	0.07
173	ce3e508482f7e3d5d9e53280b394658a	1.00	a54df4ddd4f148feb4869e852e766436	0.05
174	cfdda4bd1e7eecc09268b3fdecc66b48	0.88	5c5b806fbd1826340209616ddb9ed767	0.00
175	d22949ad1bb6742c510ec2542de36c33	0.95	11367ef1dcacb4a7f5a0c6804d0c1b6e	0.08
176	d2cd98d625d8f8d91f78497efd39a74f	0.75	cdc113531c0bb92df5ef1708dfe6ef6a	0.04
177	d4abf350fcdc1450060a68ec20a7f053	1.00	82a42c3ebfd83735f1aaf93cf8adf7e3	0.00
178	d7232b053d73b0b7e45605128210527b	0.83	711c49aa37a9e2d34a60751534afa955	0.01
179	d8ddf110fd8a92c11132d7501321c76f	0.91	8c4078d55ba07096949e82f0993a423b	0.01
180	d98d28e6fb1888861c7140af1f2b74fd	1.00	f4113d73c0b80c35c5e085e01f736ab4	0.00
181	d9a3b807fed99050e9dee44d00c60221	0.87	95e6690e70956f690f4dd7faa80d1054	0.05
182	dacf132a918dc8a6ad5206a92e262ea4	0.90	ebd4084217e44b8d49128722cdec749e	0.00
183	dbcd9c84d9ec6bde74aa95c2a432db64	0.96	f08145bf128af0fc074c4506fd42f998	0.01
184	dee39ba63f872d10458ea11b7939f606	0.98	a48e811a20a36e7db264d191eda5aebd	0.04
185	defdb34500e13fde60c3be70ade5bec8	1.00	3e4a51f98397c7b41ea8eafa7d0f6a12	0.00
186	df684f7b3182ba07cabc501e28ac716a	1.00	c0973cf4a28d7d51a039eb2ebd2b4212	0.00
187	dfdf23529f9d613f84c490b05520847f	1.00	f5231a4bd4c29f4d613b450ba123162e	0.00
188	e0231602b0e6a83856ffe3099eceff89	0.98	24a7e6091f82cd4745cd4adf52d087ab	0.00
189	e2c813f967fe4cb97e2f4c769d8cf2ab	0.88	f0de9cd1b67f83664dbd1c63858a9660	0.00
190	e2f00473c1d8bc8331b29ab36e99a215	1.00	a0d925c439259c97e99c5b8bac49c596	0.03

191	e3e5b973e71fd9597cfcd99e56e560c6	1.00	5c50dd18ff0b53e722a435ca72d298c0	0.00
192	e50b5df92f1b6d75079d353cbc06d40f	0.93	b6fe182274453b707870b16e5d2ad562	0.00
193	eb38e8279981c9f04dd6641cfdcf7200	0.98	78f700d3bbf54dd64d40f4ea49f66cb3	0.04
194	ebf53d8bdad8e57a3abcd044e76995d5	0.91	a5badaedc7c21d1d69882eef8ae210b1	0.73
195	eca016cc839f80f82594980b742bd66c	1.00	d6af724f3213a65f081d4ab1d0bd10b3	0.00
196	ecd4475669c6d77bff54f09ead8e6ea2	0.95	1b9f0700e07dc64cebfca3e9cd3d723d	0.02
197	ed246e489407df944749dc0870274679	0.90	a48e811a20a36e7db264d191eda5aebd	0.12
198	f3623baecefb4518f4a96244666575a0	1.00	e88c8c910cc83c938ffa697e084dfff7	0.10
199	f3ad33ee8b47c479ad986f8913d2958b	0.97	47eeb90302a26cb6a1a795d4ade33f13	0.00
200	f3d58ec3d6459576b1ff2d5a4b0db260	0.77	5878236d40a1f4093c07b4506b2d7a2c	0.27
201	f7808b0404f746ef62d36250b04c9fd5	0.91	4ed1b85231ced8f57fc28d8062b9773f	0.00
202	f8f4f2edad89eb78dd11b3a133b63a65	0.80	a94e3c79b963bd835001f1e89648046d	0.00
203	fadde1cb70225e72e78a5836425471f5	0.89	18a355f80308b63aea6efff77cfee4df	0.00
204	fbac8433e1312fcdd99538af65c48cb7	1.00	a3af6ad0cf42ffe9d072e52980e35cb9	0.37
205	fce23ce4bcc7bcdef65385dca0575523	0.80	b03efd14f0f503f604facbdb66aa8065	0.12
206	fdf55c9225072183b4110c81d233170a	1.00	4d9a74f4b4727a9a2c12b816f9ffb782	0.00
207	fe24ec053260d5cb09ebdfeb28bbbb41	0.68	f52b29edf8e1a7d8e3a2b2e99dbd80e0	0.03
208	fe784e376f0fec7691b114f16d7f953e	0.96	815de8eb13c20cc0ff35384abac6cad2	0.18
209	fe784e376f0fec7691b114f16d7f953e	1.00	7c510f7c0d23c2e0c68452d8a1c4f311	0.02

Appendix B

Training Logs

Table B.1: Training log of suspicious conversations classifier.

epoch	acc	loss	lr	precision	recall	test_acc	test_loss	test_precision	test_recall
0	0.981670834	0.064239611	0.001	0.988315016	0.992907813	0.989503245	0.030433353	0.997009846	0.992269125
1	0.987730679	0.038143008	0.001	0.992345005	0.995060731	0.993606914	0.021395787	0.99677627	0.996698721
2	0.990124692	0.03104671	0.001	0.993793737	0.996068641	0.991619876	0.024874741	0.996635898	0.994790506
3	0.990822945	0.029745989	0.001	0.994442731	0.996123892	0.993866094	0.020655314	0.997459403	0.996281233
4	0.992044889	0.026411013	0.001	0.994683504	0.99714812	0.991360697	0.030922482	0.992095681	0.99913911
5	0.993092272	0.023181856	0.001	0.995631009	0.997277967	0.994276461	0.020609269	0.99516226	0.999010525
6	0.993516211	0.021665116	0.001	0.995725627	0.997610324	0.993714905	0.02087828	0.997287518	0.996307689
7	0.994239401	0.01845503	0.001	0.99642092	0.997664039	0.989589639	0.030542243	0.997928315	0.991413492
8	0.996708228	0.011962818	0.0005	0.998082432	0.998534032	0.993498923	0.023729459	0.995662063	0.997703783
9	0.997780545	0.008511666	0.00025	0.9984655	0.99925293	0.993239745	0.026522336	0.996316282	0.996784433
10	0.998528674	0.006599102	0.000125	0.999002666	0.999484111	0.99416847	0.026090417	0.996507283	0.997530581
11	0.998802989	0.005792717	6.25E-05	0.999234353	0.999533405	0.993693307	0.028994139	0.996823465	0.996741571
12	0.998977551	0.005235823	3.13E-05	0.999310698	0.999637615	0.993822897	0.029615334	0.996360991	0.997331442
13	0.99897755	0.005138251	1.56E-05	0.999310436	0.999637615	0.993628514	0.032878343	0.996582338	0.996919118
14	0.999027427	0.005016995	7.81E-06	0.999336145	0.999663591	0.993887691	0.031505052	0.996345455	0.997427051
15	0.999027427	0.004927453	3.91E-06	0.999362122	0.999637615	0.993714906	0.03119694	0.996605243	0.996988255
16	0.999052364	0.004936947	1.95E-06	0.999362122	0.999663591	0.993909289	0.030598156	0.996642769	0.99714076
17	0.999052366	0.004914342	9.77E-07	0.999362122	0.999663591	0.993606913	0.030722792	0.996545053	0.996943376
18	0.999052364	0.004908533	4.88E-07	0.999362122	0.999663591	0.993801298	0.032853421	0.996499551	0.997187057
19	0.999052365	0.004904883	2.44E-07	0.999362122	0.999663591	0.993088557	0.035249219	0.995663015	0.997291846
20	0.999052367	0.004904687	1.22E-07	0.999362122	0.999663591	0.994082076	0.030539524	0.996608156	0.997359425
21	0.999052366	0.004903807	6.10E-08	0.999362122	0.999663591	0.993995684	0.032810482	0.996566739	0.997322019
22	0.999052366	0.00490319	3.05E-08	0.999362122	0.999663591	0.993477325	0.032823296	0.996273313	0.997074065
23	0.999052365	0.004902906	1.53E-08	0.999362122	0.999663591	0.993952487	0.030242408	0.9966901	0.997139919
24	0.999052365	0.004902726	7.63E-09	0.999362122	0.999663591	0.993866094	0.030759574	0.996627573	0.997114134
25	0.999052365	0.004902611	3.81E-09	0.999362122	0.999663591	0.994168471	0.027264981	0.996738415	0.997311929
26	0.999052365	0.004902543	1.91E-09	0.999362122	0.999663591	0.993822897	0.031970249	0.996450707	0.99725247
27	0.999052366	0.004902504	9.54E-10	0.999362122	0.999663591	0.993477324	0.032815973	0.9962053	0.997132686
28	0.999052365	0.004902493	4.77E-10	0.999362122	0.999663591	0.993736504	0.033195102	0.996339184	0.997272282
29	0.999052365	0.004902489	2.38E-10	0.999362122	0.999663591	0.994082075	0.030099588	0.996806153	0.997165125

Table B.2: Training log of predators classifier.

epoch	acc	loss	lr	precision	recall	test_acc	test_loss	test_precision	test_recall
0	0.909223119	0.377899998	0.001	0.851615178	0.841527491	0.933695652	0.230369718	0.930140586	0.926968827
1	0.923484706	0.256864503	0.001	0.933372217	0.911474948	0.945693191	0.188850313	0.938583823	0.933490566
2	0.937417902	0.212808147	0.001	0.946956867	0.924563708	0.955434783	0.155332893	0.952013699	0.945365053
3	0.947410396	0.178152724	0.001	0.954944507	0.937370989	0.961751436	0.130906967	0.96150403	0.954963084
4	0.953837493	0.152694581	0.001	0.96104908	0.946706699	0.967186218	0.11265928	0.968081149	0.961484824
5	0.959326328	0.133717008	0.001	0.965092368	0.952758491	0.971534044	0.101215345	0.973376843	0.966776046
6	0.963032464	0.119006287	0.001	0.968570961	0.958059673	0.971800656	0.089830546	0.973523598	0.968662838
7	0.966550948	0.107807761	0.001	0.971912931	0.961578157	0.974507793	0.082856116	0.975722544	0.970836751
8	0.970397823	0.098161267	0.001	0.974753584	0.965612685	0.977255947	0.07828269	0.978887548	0.973789992
9	0.972415087	0.090442621	0.001	0.976624532	0.967817602	0.977912223	0.073085091	0.97958026	0.97514356
10	0.974479264	0.083488603	0.001	0.978456879	0.970163258	0.978937654	0.07097886	0.980785231	0.976066448
11	0.976449615	0.077627018	0.001	0.980281324	0.972368174	0.980844955	0.069258448	0.982686042	0.978014766
12	0.978373053	0.072564329	0.001	0.981562638	0.97471383	0.980844955	0.064706641	0.982397821	0.978424938
13	0.980015012	0.067739887	0.001	0.98293804	0.976496528	0.980004102	0.063732363	0.981018726	0.977625103
14	0.981375493	0.063332939	0.001	0.984596556	0.977903922	0.981706317	0.062308216	0.983154119	0.979491386
15	0.982829799	0.059916664	0.001	0.985673036	0.979921186	0.981234619	0.061291662	0.982557726	0.979347826
16	0.983486583	0.056896824	0.001	0.986040975	0.980108838	0.981398687	0.060758896	0.982660599	0.979593929
17	0.984424845	0.053865166	0.001	0.987263228	0.981985363	0.981296144	0.060753264	0.982456554	0.979552912
18	0.985175455	0.051069943	0.001	0.987757189	0.983111278	0.983059885	0.060266124	0.984257272	0.980926989
19	0.986019891	0.048494663	0.001	0.988792203	0.983814975	0.982854799	0.058840778	0.984152442	0.981193601
20	0.986723588	0.046309425	0.001	0.988981268	0.984471758	0.983552092	0.059124999	0.984993332	0.981501231
21	0.987286545	0.044464207	0.001	0.989270415	0.985222368	0.983018868	0.058751444	0.984174146	0.981583265
22	0.987568024	0.042588671	0.001	0.989444346	0.985597673	0.983305989	0.058606756	0.984499735	0.981685808
23	0.988365547	0.040481841	0.001	0.99021473	0.986629762	0.983203445	0.059707441	0.984607857	0.981788351
24	0.988975418	0.038865029	0.001	0.99113907	0.987474198	0.983470057	0.058680014	0.984332888	0.982136998
25	0.989960593	0.037181915	0.001	0.991525372	0.988459373	0.983490566	0.058369681	0.984664444	0.982239541
26	0.990054419	0.035967267	0.001	0.992233508	0.988693939	0.983347006	0.05975584	0.98417162	0.982054963
27	0.990758116	0.034476006	0.001	0.992281833	0.989069244	0.983223954	0.060040964	0.984290813	0.982136998
28	0.991321073	0.033028521	0.001	0.992896143	0.98991368	0.983367514	0.060701083	0.984368762	0.982321575
29	0.99127416	0.031732554	0.001	0.992941845	0.989960593	0.983100902	0.063528617	0.983968073	0.982136998
30	0.991649465	0.030816541	0.001	0.993374657	0.990429724	0.983264971	0.062295221	0.984299056	0.982301066

Table B.3: Training log of IMDB sentiment classifier.

epoch	acc	loss	lr	precision	recall	test_acc	test_loss	test_precision	test_recall
0	0.720199742	0.554319194	1	0.723487037	0.746514713	0.770618557	0.484363287	0.785025643	0.752451773
1	0.758094394	0.501560903	1	0.758154659	0.761347594	0.775692655	0.470112161	0.790863142	0.742751729
2	0.771625322	0.482687753	1	0.770846212	0.774776588	0.782699742	0.462731403	0.767174754	0.802608323
3	0.774363724	0.476585297	1	0.775111635	0.775008423	0.795505799	0.448009522	0.810902421	0.772455808
4	0.784068943	0.465941323	1	0.787141145	0.780211442	0.793089562	0.447828981	0.814916409	0.752699988
5	0.781572165	0.463315687	1	0.780682821	0.783880905	0.79429768	0.433582809	0.804297815	0.785691115
6	0.788297358	0.454156299	1	0.78781652	0.790673893	0.811775129	0.421483067	0.80972457	0.810142262
7	0.788700064	0.451451732	1	0.789031725	0.789317037	0.807748067	0.42190779	0.798085815	0.828286006
8	0.793854704	0.445921032	1	0.791197077	0.798851802	0.805170747	0.426518604	0.829282294	0.770816839
9	0.794619845	0.442863409	1	0.794709771	0.796641807	0.814594072	0.413702573	0.828301239	0.788262621
10	0.792928479	0.441793164	1	0.794179399	0.791411845	0.811694588	0.41407838	0.834997777	0.772118981
11	0.797680412	0.435983752	1	0.796349513	0.800285756	0.805090206	0.416067025	0.805950741	0.803206224
12	0.796593106	0.435412534	1	0.796956258	0.796171364	0.816929768	0.412878025	0.826155449	0.79489288
13	0.80078125	0.431553688	1	0.799481416	0.803644734	0.814916237	0.411818707	0.830018909	0.801412624
14	0.801788015	0.428098472	1	0.803373678	0.798985543	0.814030284	0.412412138	0.817243295	0.806006433
15	0.803600193	0.426660524	1	0.803987503	0.804178489	0.809761598	0.413614292	0.81075151	0.80389542
16	0.801989369	0.42366988	1	0.801970937	0.80237345	0.801143686	0.416656671	0.819240161	0.773771181
17	0.807385631	0.421749152	1	0.805868455	0.811764462	0.809519974	0.408267202	0.806428052	0.803770884
18	0.804083441	0.422668898	1	0.805224468	0.803282213	0.812097294	0.40858852	0.806859019	0.823968791
19	0.807586985	0.418450686	1	0.808476965	0.806806373	0.814835696	0.406402552	0.816662567	0.813706539
20	0.807425902	0.416094244	1	0.807341013	0.809171448	0.814996778	0.41140276	0.815453595	0.806733669
21	0.809519974	0.414259617	1	0.808858235	0.811333388	0.818057345	0.406861082	0.817573748	0.818253271
22	0.809076997	0.41116472	1	0.808509804	0.811764461	0.815560567	0.401559035	0.813642278	0.809633636
23	0.812258376	0.406848933	1	0.812655182	0.812602095	0.816204897	0.401542163	0.812825146	0.823460528
24	0.815600838	0.403112617	1	0.815476904	0.816758379	0.819909794	0.404557824	0.797865031	0.856019361
25	0.816889497	0.402813628	1	0.815530463	0.819822034	0.816285438	0.404214822	0.828656178	0.807477029
26	0.816164626	0.401990277	1	0.817634044	0.814642586	0.819104381	0.396448941	0.835988523	0.790849984
27	0.816124356	0.3986251	1	0.817211416	0.815591969	0.821037371	0.394326631	0.823167967	0.810912382
28	0.821037371	0.393209117	1	0.821523386	0.821288282	0.822567655	0.396083433	0.835772312	0.805539541
29	0.821762242	0.39217359	1	0.822788649	0.820902912	0.826916881	0.386929106	0.828166675	0.826790541
30	0.821560889	0.390387485	1	0.821514083	0.822671415	0.81966817	0.400214738	0.81850365	0.821329419
31	0.82421875	0.384544946	1	0.825194699	0.822719117	0.819265464	0.39471024	0.818587247	0.824394585
32	0.827118235	0.384380962	1	0.828201841	0.827228714	0.813466495	0.402044874	0.832643717	0.784089086
33	0.827561211	0.38000476	1	0.827821469	0.828045708	0.82554768	0.384754767	0.82981618	0.821852299
34	0.829574742	0.378286402	1	0.83185168	0.827826118	0.825789304	0.391033701	0.813685215	0.848693806
35	0.828809601	0.378554381	1	0.828753873	0.829441793	0.821923325	0.392746068	0.803709619	0.848731569
36	0.833722616	0.374308294	1	0.833584191	0.834504367	0.818218428	0.393857764	0.818217267	0.816111055
37	0.833722616	0.370974625	1	0.835011297	0.831139791	0.823775773	0.392397027	0.821305885	0.825073653
38	0.832595039	0.369123658	1	0.833232468	0.832263344	0.827239046	0.384577066	0.817985581	0.850404495
39	0.832715851	0.367245577	1	0.832904602	0.832882833	0.826433634	0.38731899	0.822771174	0.834108082
40	0.823936856	0.387851858	1	0.823588019	0.825288364	0.81443299	0.404063394	0.803516732	0.828447485
41	0.825789304	0.385934494	1	0.82656316	0.824850835	0.824903351	0.393024375	0.824403819	0.828887151
42	0.826957152	0.381616041	1	0.82639419	0.827744531	0.819104381	0.400955993	0.82125938	0.81339988
43	0.830782861	0.37702618	1	0.83158684	0.830356376	0.824259021	0.386410897	0.82520278	0.823562385
44	0.828648518	0.376102575	1	0.827907438	0.830942287	0.825144974	0.385662076	0.81896842	0.824276487
45	0.831709085	0.372593237	1	0.832617963	0.831497548	0.826594716	0.389714604	0.809631715	0.847421558
46	0.832635309	0.370028781	1	0.832593038	0.833267614	0.819909794	0.398621961	0.811593465	0.831805212
47	0.832433956	0.371365488	1	0.83158006	0.834672771	0.821359536	0.398024148	0.856349829	0.768533418
48	0.836662371	0.363943138	1	0.837905614	0.836643354	0.823936856	0.392425399	0.82802002	0.811913133
49	0.834326675	0.364438997	1	0.834609651	0.83493405	0.832313144	0.38199033	0.831316723	0.831346739
50	0.838877255	0.361109883	1	0.838207853	0.841921547	0.818218428	0.389967505	0.830625232	0.797281886