“FRIENDLY CHAT”- A chat application with multi-headed classification models for identifying abusive levels and their comparative study

Parkavi A¹, Sowmya B J² and Pushpalatha M N³
¹,²Department of Computer Science and Engineering, M S Ramaiah Institute of Technology, Bengaluru -560054
³Department of Information Science and Engineering, M S Ramaiah Institute of Technology, Bengaluru -560054
E-mail : ¹parkavi.a@msrit.edu, ²sowmyabj@msrit.edu, ³pushpalathamn@msrit.edu

Abstract: An application that tracks the use of offensive and vulgar language on chats on social media networks in order to prevent users from becoming disrespectful may play a role in changing how users interact. "Friendly chat" is a play on a community messaging program, which allows users to address a range of topics that are totally unregulated. Once things start heating up a user may become abusive towards other users, the application can disperse the situation quickly by implementing protective measures to stop the abusive user in his tracks. Abuse is however not the only type of "toxic" material to which a user is exposed when communicating with other users. Hatred of identification, intimidation, and other forms of obscenity may also become a part of the discussion. To identify a post as toxic, we need to refer to a reference, using the "toxic comment" dataset consisting of a large number of Wikipedia comments which have been rated by human critics who classify the comment to fall under one or more of the above-mentioned categories. Integrate this model in the real time data that we receive from “Friendly Chat” to classify a user into one or more of these categories. Use techniques of classification and machine learning such as Naïve Bayes, LSTM and Binary relevance and chain classifiers models. Our application’s capable of detecting abusive users.

Keywords: LSTM-Long short term memory, toxicity, obscene, Naive Bayes

1. INTRODUCTION

The world in which we live today is a world of social media that is about communicating with friends and colleagues around the world. The use of social media has made it possible for people of all ages to communicate and voice their opinions on the same site. Social media includes all manner of interactive media, such as online chat rooms, discussion boards and bulletins, and online discussion groups where people can discuss with their peers freely and openly. Furthermore, it can be difficult to discuss the issues that you care about. The constant threat of bullying along with online harassment means people will stop voicing themselves and will not be able to seek views that contradict popular belief. Current sites are struggling to suppress this kind of harassment online, contributing to their online community’s rapid decline as users are unable to express themselves freely without fear of being harassed or experiencing identity hatred for their opinions.

An application to monitor the use of abusive and vulgar language on chats on social media networks
to prevent users from being obscene can play a part in modifying the way users interact with each other. "Friendly chat" is a tool on a community messaging platform, which allows users to address a range of topics that are totally unregulated. Once things start heating up a user may become abusive towards other users, the application can disperse the situation quickly by implementing protective measures to stop the abusive user in his tracks. Abuse is however not the only type of "toxic" material to which a user is exposed when communicating with other users. Hatred of nationality, intimidation, and other forms of obscenity may also become a part of the discussion.

To classify a message as toxic we need a baseline to refer to, we use the "toxic comment" dataset which consists of a large number of Wikipedia comments which have been rated by human rater's who classify the comment to fall under one or more of the above-mentioned categories. Categories and incorporate this model in the real-time data that we receive from "Friendly Chat" to classify a user into one or more of these categories. Using classification and machine learning techniques such as simple Bayesian classifiers our application will not only be able to detect when a user is abusive but also be able to learn when a user is spewing.

The problem faced in social media application today is the lack of conversational filters. Social media platforms such as Instagram and Facebook do a lackluster job when it comes to identifying incidents of online abuse. These companies only focus on censoring or filtering the content being displayed on the website such as images but fail when it comes to restricting their audiences from participating in online abuse. The problem is even more starkly visible in the messenger applications provided on these platforms. These applications allow users to freely interact and talk, thus some more vulnerable users are susceptible to online abuse. Furthermore, applications such as Facebook allow users to search for and find other users to message. This makes it all the more dangerous to have an open online chat where other users can easily be threatened or blackmailed.

We are trying to address these issues by developing an App and to achieve the different objectives as follows

- To develop a full-fledged chatting application with a professional quality front end
- Develop a bug free backend using firebase to provide the user with a hassle free and the best experience possible while using the application.
- To understand the use of machine learning in conversational AI.
- To Develop multi-headed machine learning models that are capable of classifying messages sent by users into varying degrees of toxicity
- Explore all additional methods of classification using machine learning to obtain the best suited method that provides the most accurate results.
- To implement an administrator interface that is easy to use and gather data from as well as interact with users.
- To run comparative studies on the various developed models based on various factors and obtain visual representation of comparison results.
- To develop a kink free application that users will enjoy.

We use several tools and techniques to build an App and solve these issues, they are as follows

- Professional cloud based chatting application with real time updating capabilities
- Chat application will allow users to interact using text messages or images
- Naive bayes and LSTM classification model to classify varying degrees of abusiveness
- Binary relevance and chain classifiers classification model to classify varying degrees of abusiveness
- Administrator front end to classify messages sent by the users
- Comparative study of the 3 machine learning models implemented for classifying abusiveness
The motivation and the scope of the application being currently developed is solely for the purpose of conversation analysis in a chat. Since there already exists various chatting applications on social media platforms such as Facebook and Google the machine learning models developed can be used on ordinary chat data as well for private company analysis. This will allow companies to have a better idea about their user base and what kind of language is prevalent on their websites forum. The application being developed during the course of this project however is restricted to user analysis on a local machine and partially automated notifications being sent to the users regarding their behavior. The application will have a direct impact on the language the users use in everyday conversation making for a friendlier and accepting online chat environment. As another demonstration of the practical applications of the models we have also expanded the test cases to include detection of bad language in children's novels as well.

2. LITERATURE SURVEY

The papers referred in relevance to this App development revealed various techniques that have been used for similar classification and analysis problems. Furthermore, the papers also reveal various aspects of a variety of online interactions and different kinds of techniques that can be applied on group chats to perform topic and trend analysis as well. Various machine learning techniques and natural language processing analysis related paper work and books have been referred which will help us formulate our problem statement and mode work. The following papers in the table are been referred and conclusions.

| Paper Title                                      | Reference                                                                 | Conclusions                                                                 |
|--------------------------------------------------|---------------------------------------------------------------------------|-----------------------------------------------------------------------------|
| Humans and Bots in Internet Chat [1]             | Steven Gianvecchio, MengjunXie, Member, IEEE, Zhenyu Wu, and Haining Wang, Senior Member, IEEE, “Humans and Bots in Internet Chat: Measurement, Analysis, and Automated Classification” presented in IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 19, NO. 5, OCTOBER 2011. | classifying chatbots into 6 types                                             |
| Automatic Classification of Player Complaints in Social Games [2] | KorayBalcı and Albert Ali Salah, “Automatic Classification of Player Complaints in Social Games” presented in IEEE TRANSACTIONS ON COMPUTATIONAL INTELLIGENCE AND AI IN GAMES, VOL. 9, NO. 1, MARCH 2017. | classified layer complaints on an online card game. Searched through the complaints to distinguish actual and false complaints lodged by the players |
| Affect and Social Processes in Online Communication [3] | Marcin Skowron, Mathias Theunis, Stefan Rank, and ArvidKappas, “Affect and Social Processes in Online Communication—Experiments with an Affective Dialog System” presented in EEE TRANSACTIONS ON AFFECTIVE COMPUTING, VOL. 4, NO. 3, JULY-SEPTEMBER 2013 | provided an insight into the behaviour of humans in online interactions using a simulated environment. It proposed the use of an affective dialog system that was used as a tool to study various human emotional responses in an online communication setting. |
| Machines in the conversation [4]:                | Hyungjong Noh, Seonghan Ryu, Donghyeon Lee, Kyusong Lee, Cheongjae Lee, and Gary Geunbae Lee, “An Example-Based Approach to Ranking Multiple Dialog States for Flexible Dialog Management” presented in IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING, VOL. 6, NO. 8, DECEMBER 2012. | was able to detect trending topics during the JAM and create separate links to separate JAM pages which had content relevant to that particular topic Theme and trend analysis |
| An Example-Based Approach to Ranking Multiple Dialog States | W. S. Spangler, J. T. Kreulen and J. F. Newswanger, “Machines in the conversation: Detecting themes and trends in informal | presented a new hybrid dialog management that integrated a statistical ranking algorithm |
| Paper Title | Reference | Conclusions |
|-------------|-----------|-------------|
| for Flexible Dialog Management [5] | communication streams”, IBM SYSTEMS JOURNAL, VOL 45, NO 4, 2006. | into an example-based dialog management approach for chatting applications |
| Interaction Analysis of the ALICE Chatterbot: A Two-Study Investigation of Dialog and Domain Questioning [6] | Robert P. Schumaker, Associate Member, IEEE, and Hsinchun Chen, Fellow, IEEE, “Interaction Analysis of the ALICE Chatterbot: A Two-Study Investigation of Dialog and Domain Questioning”, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART A: SYSTEMS AND HUMANS, VOL. 40, NO. 1, JANUARY 2010. | How chatterbots are trained into giving not just binary, or conventional response. The paper discussed understanding their response to various interrogative styles. |
| Text-driven conversational avatar interface for instant messaging on mobile devices” [7] | Mario Rincón-Nigro and Zhigang Deng, “Technical Correspondence: A Text-Driven Conversational Avatar Interface for Instant Messaging on Mobile Devices”, IEEE TRANSACTIONS ON HUMAN-MACHINE SYSTEMS, VOL. 43, NO. 3, MAY 2013. | the purpose of a conversational avatar as an alternative for the betterment of the user experience on IM platform on mobile based applications. : 3D avatars |
| Syneske$$\text{t}$: An Open Source Library for Sentence-Based Emotion Recognition[8] | Asier Marzo, Oscar Ardaiz, María Teresa Sanz de Acevedo, and María Luisa Sanz de Acevedo, “Personalizing Sample Databases With Facebook Information to Increase Intrinsic Motivation”, presented in IEEE TRANSACTIONS ON EDUCATION, VOL. 60, NO. 1, FEBRUARY 2017 | incorporate emotion recognition in textual data into six emotional types defined by Ekman and makes use of the standard Ekman emotion classification methodology. |
| Influences on Query Reformulation in Collaborative Web Search [9] | Zhen Yue, Shuguang Han, Dacing He, and Jiepu Jiang, University of Pittsburgh, “Influences on Query Reformulation in Collaborative Web Search”, published by the IEEE Computer Society 0018-9162/14/$31.00 © 2014 IEEE. | utilisation and importance of chat data Analysis of chat data for query reformulation. |
| Recommending Nearby Strangers Instantly Based on Similar Check-In Behaviors[10] | Xiuquan Qiao, Wei Yu, Jinsong Zhang, Wei Tan, Senior Member, IEEE, Jianchong Su, Wangli Xu, and Junliang Chen, “Recommending Nearby Strangers Instantly Based on Similar Check-In Behaviors”, IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING, VOL. 12, NO. 3, JULY 2015 | Considering not just conventional feature or parameters for analysis. How to solve any sparseness disadvantage using, a Kernel Density Estimation (KDE) model |
| Detecting Predatory Behaviour in Game Chats:[11] | Yun-Gyung Cheong, Alaina K. Jensen, Elín Rut Guðnadóttir, Byung-Chull Bae, and Julian Togelius, “Detecting Predatory Behavior in Game Chats” IEEE TRANSACTIONS ON COMPUTATIONAL INTELLIGENCE AND AI IN GAMES, VOL. 7, NO. 3, SEPTEMBER 2015. | Bag of words representation Sexual assault on gaming platform NLP Classification |
| Detecting Misbehaviour in Online Video Chat Services:[12] | Qin Lv , University of Colorado Boulder, “Detecting Misbehavior in Online Video Chat Services”, presented in IEEE INTERNET COMPUTING MAY/JUNE 2013 | detecting misbehaviour in online video chat services by determining not only how much skin is exposed, but also whether a user shows his or her face, they can potentially detect misbehaving users with high accuracy. |
| ConChat: [13] | Anand Ranganathan, Roy H.Campbell, Arathi Ravi, and Anupama Mahajan University of Illinois at Urbana-Champaign. “ConChat: A Context-Aware Chat Program”, 1536-1268/02/$17.00 © 2002 IEEE JULY–SEPTEMBER 2002 PERVERSIVE-computing. | Using contextual cues, users can infer during a conversation what the other person is doing and what is happening in his or her immediate surroundings. |

| A Chat with | Peter I. Corke, “A Chat with Ruzena Bajcsy”, | In this interview Mrs Bajcsy |
Ruzena Bajcsy [14] presented in IEEE Robotics & Automation Magazine SEPTEMBER 2010. has explained about Cybernetics and Artificial Intelligence.

Personalizing Sample Databases With Facebook Information to Increase Intrinsic Motivation,[15] Asier Marzo, Oscar Ardaiz, María Teresa Sanz de Acedo, and María Luisa Sanz de Acedo, “Personalizing Sample Databases With Facebook Information to Increase Intrinsic Motivation” presented in IEEE TRANSACTIONS ON EDUCATION, VOL. 60, NO. 1, FEBRUARY 2017. Students using a database of self-social media information had a larger increase in intrinsic motivation than students using data from an unknown person or from a business.

Baselines and Bigrams: Simple, Good Sentiment and Topic Classification[16] Sida Wang and Christopher D. Manning Department of Computer Science Stanford University Stanford, CA 94305 {sidaw,manning}@stanford.edu Integration of Naïve Bayes model along with linear regression to classify the comments Model 1 is implemented using this paper’s functionality

3. IMPLEMENTATION AND RESULTS

Figure 1 shows the system architecture consisting of different layers such as Application layer, Logical layer and External Layer. A Chat Application at the application layer, Data Analysis and Integration in the logical layer and different set users at the External layer. User interactions are recorded and entered systematically on the database in the form of a JSON tree. The entries in the real time database have already been discussed. Once a sufficient amount of data has been collected the administrator will run the module to pull data on to the local system and obtain the JSON data in a CSV format. The administrator will then run the classification model with the highest accuracy to which will be pre-determined to classify the messages on the real time database. Once the classification is completed the administrator runs the final module to update the real time database with the results of the classification. Additionally, the administrator can send custom notifications to users using firebase console. Privilege’s to block a user’s account is also given to the administrator.
who can disable user accounts depending on the user’s behaviour and interaction.

3.1. Implementation of Machine Learning models
This section will cover the algorithms used in the machine learning models and how the models are implemented.

3.2. Naive Bayes model
The model utilizes a bag of words approach that will tokenize each training document into a set of unigrams and bigrams. This is followed by a sparse matrix construction. The sparse matrix will be constructed using the tf-idf vectorizer function which is included in the sklearn library in python. The function will remove the commonly occurring words that occur too often in the training set as well as the words that occur very rarely. For the remaining words the tf-idf value is stored in the sparse matrix. This matrix forms a uniform input to the classification models. The naïve Bayes model utilizes naïve Bayes along with linear regression to classify individual data points into one or more of the six categories as mentioned before. The input taken by this classification model is the sparse matrix obtained directly from the bag of words approach. The model will perform classification using naïve Bayes along with linear regression to perform a classification for every document in the test set. This classification is done one by one for each and every category.

3.3. Naive Bayes Model Algorithm
- Tokenize the documents present in the training set by removing punctuations and white spaces
- Using tf-idf vectorize function create a the sparse matrix for the test and training set
- Create a predictions matrix (m x n) where m is the number of test documents and n is the number of labels (6). Initialize the matrix elements to 0
- Loop through the labels and obtain a linear regression model along with the inverse log factor value for each label in the training set
- Predict the probability of the document falling into a category using predict_proba function on the test sparse matrix with every element multiplied by rPredictions obtained column by column in the predictions matrix

3.4. LSTM model
Here what we have tried to bring in is an improved version of the bidirectional LSTM model which uses neural networks provided by Keras module of Python. The model uses an already existing Glove word vector file that is returned from an unsupervised learning algorithm that obtains vector representations of words and is trained upon the global word-word co-existence statistics, from which we obtain word->vector matrix which is used as our base embedding file. This LSTM model has two full connected layers, i.e. wraps the hidden layer used by the LSTM model with a Bidirectional layer. The two layers will have its model fit the input as such from the training set and the reverse of the input sequence respectively. It also should be noted that the model is run for two epochs, and the dropout values have been adjusted accordingly in order to avoid an under or over-fitting scenario. This model like naïve Bayes support vector machine model helps classify individual data points into one or more of the six categories.

3.5. LSTM Algorithm
- Initialize the path name and include all the csv files required such as
- EMBEDDING_FILE -> the pre-existing GloVe model results,
- TRAIN_DATA_FILE -> the CSV file,
- TEST_DATA_FILE -> the test CSV file.
- Set some basic configuration parameters:
  o embed_size -> how big is each word vector
  o max_features -> how many unique words to use/ number of rows in embedding
vector
- maxlen -> max number of words in a comment to use
- Read our data and replace missing values in the training set with a default parameter.
- Modify each and every comment into a list of word indexes of equal length with necessary truncation or padding, since Keras processing requires us to do so
- Read the glove word vectors from the EMBEDDING_FILE into a dictionary from word-vector.
- Create the embedding matrix, with random initialization for words that aren't in GloVe, using the same mean and stdev of embeddings the GloVe already has.
- Create the simple bidirectional lstm model using keras command in 2 epochs
  - add some dropout to the LSTM since even 2 epochs are enough to overfit.
  - Add the activation function for the dropout to work on: relu, sigmoid
- Fit the model
- Get predictions for the test set and prepare a submission CSV

3.6. Binary Relevance and Chain Classifiers model
Binary relevance solves problems where each example is represented by a single instance while being simultaneously connected to multiple class labels Binary significance is perhaps the most insightful way to learn from multilabel instances. This functions by breaking down the learning function of multi-label into a number of independent binary learning tasks (one per label).

In view of its potential weakness in ignoring correlations between labels, here we have used Classifier chains. In this method, the first classifier is trained on any input X.

The subsequent classifiers will then be trained on the input X and the predictions of all previous classifiers in the chain. This approach attempts to draw the signals among the preceding target variables from the correlation. Binary Relevance and Chain Classifiers Algorithm
- Find correlation between features and the targets
- Clean the comments in the dataset by removing punctuations and whitespaces.
- Define X from entire train and test data for use in tokenization by Vectorizer.
- Import and instantiate TfidfVectorizer.
- Learn the vocabulary in the training data, then use it to create a document-term matrix.
- Examine the document-term matrix created from X_train.
- Transform the test data using the earlier fitted vocabulary, into a document-term matrix
- Predict the probability of the document falling into a category using the binary relevance and chain classifiers model
- Predictions obtained column by column in the predictions matrix

4. RESULTS AND INFERENCES
The scores represent the ROC-AUC scores depicting the training accuracy of the individual model’s category by category. The ROC-AUC score refers to the Area under the Receiver Operating Characteristic Curve. AUC is an abbreviation for undercurve area. It is used in classification research to decide which of the models the classes are best predicted by. An example of its application are ROC curves. Here, the true positive rates are plotted against false positive rates The ROC curve is an essential tool for assessing the diagnostic tests. In a ROC curve the true positive rate (Sensitivity) is plotted for different cut-off points of a parameter within function of the false positive rate (100-Specificity). ROC, visualization receiver operating characteristics (ROC) graphs are a useful technique for organizing and visualizing the output of the classifiers and are shown in table 1.
Table 1. ROC-AUC scores naïve Bayes model

| Model Name      | ROC-AUC Scores (Category Wise) |
|-----------------|--------------------------------|
|                 | Toxic | Severe Toxic | Obscene | Insult | Threat | Identity hate |
| Naïve Bayes     | 0.999465 | 0.988550 | 0.988550 | 0.985306 | 0.960053 | 0.96392 |

The accuracy score for LSTM model is provided by the in-built Python function as well when the model is being fit. Even the loss that occurs while the model is being fit on each of the epoch is produced. And the results are shown in table 2:

Table 2. Accuracy scores for LSTM

| Model Name | Epoch | Accuracy Scores (Epochwise) |
|------------|-------|-----------------------------|
|            |       | Loss | Accuracy | Val_loss | Val_acc |
| LSTM       | 1     | 0.0603 | 0.9791 | 0.0489 | 0.982 |
|           | 2     | 0.0446 | 0.9830 | 0.0480 | 0.982 |

The accuracy score for Binary relevance with Chain classifier model is provided by the in-built Python function as well when the model is being fit and the results are shown in table 3:

Table 3. Accuracy scores for Binary relevance with chain classifier model

| Model Name            | Accuracy Scores (Category Wise) |
|-----------------------|--------------------------------|
|                       | Toxic | Severe Toxic | Obscene | Insult | Threat | Identity hate |
| Binary relevance and  | 0.9676 | 0.9931 | 0.98322 | 0.981920 | 0.9981 | 0.9955 |
| chain classifiers     |       |            |         |        |       |             |

From the below graphs, it is clearly understood that the LSTM model is able to classify well and differentiate between the different levels of toxicity. The former model is only able to detect toxicity but is unable to read further into it and understand if the comment made is obscene, a threat, or identity hate.

Figure 2. Naïve Bayes Classification Result

Figure 2 describes the result of Naïve Bayes Classification, where it classifies the data into different toxic, obscene, threat and insult. Likewise different algorithms are applied to check the abusive levels in the friendly chat. Figure 3 represents the results of LSTM model and figure 4 shows the result of Binary relevance with chain classifier algorithm.
Figure 3. LSTM Classification Result

Figure 4. Binary relevance with chain classifier Classification Result

Figure 5. Identity-hate

Figure 6. Toxicity

Figure 7. Severe toxicity

Figure 8. Obscene
Figure 5 shows the comment vs category. It detects the category as Identity-hate and figure 7 shows the comment vs category and detects the category as toxicity and severe toxicity and it is as shown in figure 6. Figure 8 shows the category as obscene and figure 9 shows threat vs comment. From this app we are inferring that most of the chat contains the Identity-hate, obscene and toxicity.

5. CONCLUSION

This study was intended to create an application that would perform a toxicity classification. It also brings in a Multi-label classification model to understand and differentiate the levels of toxicity. Three machine learning models that were commonly used for this multilevel classification were tested: the naïve Bayes, LSTM and Binary relevance and chain classifiers model. Based on this study, we were able to propose a better model for the application. There are a number of possible areas for future scope. The real life deployment of this would result in a much more accurate result. With the large amount of data received, the LSTM model would be retrained and result in bringing out more accurate and reliable result. Abusiveness in multiple languages. This will drastically expand the scope of the application. The models can also be used for detection of abusiveness in children's books and any other written medium. Using the same concept we can further expand the scope of the application to not just cover vulgar text but also to the detection of vulgar images being sent online as well. Censorship is another domain which can be reached out into by detecting use of vulgar language in novel and by detecting abusive language in movie subtitles and censoring them out to have a more pleasant viewing experience.

REFERENCES

[1]. Steven Gianvecchio, MengJunXie, Member, IEEE, Zhenyu Wu, and Haining Wang, Senior Member, IEEE, “Humans and Bots in Internet Chat: Measurement, Analysis, and Automated Classification” presented in IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 19, NO. 5, OCTOBER 2011.

[2]. KorayBalcı and Albert Ali Salah, “Automatic Classification of Player Complaints in Social Games” presented in IEEE TRANSACTIONS ON COMPUTATIONAL INTELLIGENCE AND AI IN GAMES, VOL. 9, NO. 1, MARCH 2017.

[3]. Marcin Skowron, Mathias Theunis, Stefan Rank, and ArvidKappas, “Affect and Social Processes in Online Communication—Experiments with an Affective Dialog System” presented in EEE TRANSACTIONS ON AFFECTIVE COMPUTING, VOL. 4, NO. 3, JULY-SEPTEMBER 2013

[4]. Hyungjong Noh, Seonghan Ryu, Donghyeon Lee, Kyusong Lee, Cheongjae Lee, and Gary
Geunbae Lee, “An Example-Based Approach to Ranking Multiple Dialog States for Flexible Dialog Management” presented in IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING, VOL. 6, NO. 8, DECEMBER 2012.

[5]. W. S. Spangler, J. T. Kreulen and J. F. Newswanger, “Machines in the conversation: Detecting themes and trends in informal communication streams”, IBM SYSTEMS JOURNAL, VOL. 45, NO 4, 2006.

[6]. Robert P. Schumaker, Associate Member, IEEE, and Hsinchun Chen, Fellow, IEEE, “Interaction Analysis of the ALICE Chatterbot: A Two-Study Investigation of Dialog and Domain Questioning”, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART A: SYSTEMS AND HUMANS, VOL. 40, No. 1, JANUARY 2010.

[7]. Mario Rincón-Nigro and Zhigang Deng, “Technical Correspondence: A Text-Driven Conversational Avatar Interface for Instant Messaging on Mobile Devices”, IEEE TRANSACTIONS ON HUMAN-MACHINE SYSTEMS, VOL. 43, No. 3, MAY 2013.

[8]. Asier Marzo, Oscar Ardaiz, María Teresa Sanz de Acedo, and María Luisa Sanz de Acedo, “Personalizing Sample Databases With Facebook Information to Increase Intrinsic Motivation”, presented in IEEE TRANSACTIONS ON EDUCATION, VOL. 60, No. 1, FEBRUARY 2017.

[9]. Zhen Yue, Shuguang Han, Daqing He, and Jiepu Jiang, University of Pittsburgh, “Influences on Query Reformulation in Collaborative Web Search”, published by the IEEE Computer Society 0018-9162/14/$31.00 © 2014 IEEE.

[10]. Xiuquan Qiao, Wei Yu, Jinsong Zhang, Wei Tan, Senior Member, IEEE, Jianchong Su, Wangli Xu, and Junliang Chen, “Recommending Nearby Strangers Instantly Based on Similar Check-In Behaviors”, IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING, VOL. 12, NO. 3, JULY 2015.

[11]. Yun-Gyung Cheong, Alaina K. Jensen, Elín Rut Guðnadóttir, Byung-Chull Bae, and Julian Togelius, “Detecting Predatory Behavior in Game Chats” IEEE TRANSACTIONS ON COMPUTATIONAL INTELLIGENCE AND AI IN GAMES, VOL. 7, NO. 3, SEPTEMBER 2015.

[12]. Qin Lv, University of Colorado Boulder, “Detecting Misbehavior in Online Video Chat Services”, presented in IEEE INTERNET COMPUTING MAY/JUNE 2013.

[13]. Anand Ranganathan, Roy H. Campbell, Arathi Ravi, and Anupama Mahajan University of Illinois at Urbana-Champaign, “ConChat: A Context-Aware Chat Program”, 1536-1268/02/$17.00 © 2002 IEEE JULY–SEPTEMBER 2002 PERVERSIVE-computing.

[14]. Peter I. Corke, “A Chat with Ruzena Bajcsy”, presented in IEEE Robotics & Automation Magazine IEEE Robotics & Automation Magazine SEPTEMBER 2010.

[15]. Asier Marzo, Oscar Ardaiz, María Teresa Sanz de Acedo, and María Luisa Sanz de Acedo, “Personalizing Sample Databases With Facebook Information to Increase Intrinsic Motivation” presented in IEEE TRANSACTIONS ON EDUCATION, VOL. 60, No. 1, FEBRUARY 2017.

[16]. Sida Wang and Christopher D. Manning, “Baselines and Bigrams: Simple, Good Sentiment and Topic Classification”, Department of Computer Science Stanford University Stanford, Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, pages 90–94, Jeju, Republic of Korea, 8-14 July 2012.