Detecting sentences that may be harmful to children with special needs

Merav Allouch
Computer Science Department
Ariel University
Ariel, Israel
merav@g.jct.ac.il

Amos Azaria
Computer Science Department
Ariel University
Ariel, Israel
amos.azaria@ariel.ac.il

Rina Azoulay
Dept. of Computer Science
Jerusalem College of Technology
Jerusalem, Israel
rrinaa@gmail.com

***Abstract*—Children and adults with special needs may find it difficult to recognize danger and threats as well as socially complex situations. Thus, they are under the risk of being victims of exploitation, violence and attacks. In addition, they may find themselves unintendedly insulting their friends, relatives or caregivers. In this paper, we propose an autonomous agent to assist the special needs person (child or adult) in the goal of recognizing risky or insulting situations. The autonomous agent will detect these situations and will signal them to the user (by text, speech, or other signaling forms). We composed a dataset containing 13,490 sentences, categorized into one of four classes: a 'normal' sentence, an insulting sentence, a negative sentence about a different person, or a risky sentence that may indicate a dangerous situation for the special needs person, which requires immediate intervention. We used several machine learning methods, and we found that the most accurate methods were the random forest method with 100 estimators, a voting method using several classifiers, and a convolutional neural network (CNN) with embedding. All of these mechanisms reached an accuracy close to 70% in classifying the sentences in the test set. Finally, using an ensemble method comprising a panel of the 5 best CNN based methods, improves the accuracy of the results and the F1-score. Our results demonstrate the feasibility of building an assisting agent that will accompany the special needs children and adults, and assist them in their daily social interactions.**

***Index Terms*—Human-agent interaction, Special needs children, Hate speech detection, Text classification**

I. Introduction

Children and adults with special needs may need help to understand their environment. They can be threatened by persons, even in their home or school area. For example, such children may agree to follow strangers and thus are at high risk of being harmed by them.

In addition, these children may create destructive relationships, and they can suffer from various kinds of abuse and bullying. In particular, they can be harmed by people with malicious intentions and they might even not notice it. In addition, they may also talk in a way that may harm people around them, or may be used against them, where they are laughed at or exploited.

The overall goal of our work is to develop an autonomous agent to assist children with special needs in their communication with other people. In order to help these children, the agent must be aware of the child's interactions, translate the audio contents into text, recognize the text classification, and detect if a special situation occurs (i.e. a risky situation, or a situation involving insulting context). Given the recognized situation, the agent should be able to give the child relevant feedback, according to the special situations that occurred, or even to warn his/her parents or caretakers if it recognizes a risky situation.

Thus, the aim of our study is to design an autonomous agent that will be able to detect insulting or risky sentences, in order to be able to provide relevant feedback when such insulting sentence or sentences are detected. In order to reach this goal, We composed a dataset of approximately 13,490 sentences, falling into the following four categories: 'normal' sentences; insulting sentences[1]; negative sentences about a third person; or risky sentences that may indicate a dangerous situation for the special needs person that requires immediate intervention.

Next, we proceeded by searching for machine learning methods that can be useful for the sentence categorization task. In order to do so, we chose several machine learning methods

We show that given a large enough sentences dataset, with classified sentences, we are able to predict the classification of new unseen sentences, with a mean of 70% accuracy per sentence, when using the random forest or the CNN based method. In addition, we developed a panel based on 5 CNN based systems, chosen from 10 CNN based systems. Each CNN system was built by training with 90% of the training set, and validated by the rest 10% of the training set (the validation set). Then, the best 5 CNN systems were chosen, and for each sentence in the test set, a voting criterion was used for classification. Using this voting panel, the average accuracy over the test set increased to 72.2% (std=0.009, for 50 trials) and the F1 score to 0.714 (std=0.009), higher than all other individual methods. Our results demonstrate the feasibility of building an assisting agent that will accompany the special needs children and adults and assist them in their daily social interactions

Our paper is organized as follows. In Section II we provide a brief overview of relevant studies. In Section V we describe

[1]There is also a fifth category of sentences where their meaning depends on the context of the situation in which they were stated; however this category is not included in the current study because such interpretation requires additional information about the situation, rather than the text itself

the Machine Learning methods used in our study, and in Section IV we describe the algorithms used to prepare the dataset before starting the learning methods. In Section VI we describe our main results, and finally, in Section VII we provide conclusions and directions for future work.

II. Related Work

Children and adults with special needs may encounter difficulties in their communication with other people, and thus, they may be vulnerable to risks, danger and misunderstandings. We intend to develop an assistant artificial agent, and in this study we consider the first step in its development: providing the agent with the ability to understand the current social situation, given the last sentence(s) that was/were said by the child/adult or to him. In order to implement the idea of an autonomous assistant agent, we need to solve the relevant algorithmic challenges. Firstly, the autonomous assistant agent should be able to recognize problematic situations that the child encounters. Thus, we proceed in this overview with some related work on text classification methods, and we concentrate on work on emotion recognition and bullying recognition.

Sentiment analysis is often used in order to determine the sentiments and emotions of the writer or a speaker of text or speech [12]. Zhang et al. [15] provide a review on current algorithms in sentiment analysis using deep learning. Libeskind et al. [14] detect abusive Hebrew texts in comments on Facebook, using highly sparse n-gram representation of letters. Since comments in social media are usually short, they suggest four dimension reduction methods that classify similar words into groups, and they show that the character n-gram representations outperform all the other representations.

Socher et al. [23] first proposed a semi-supervised Recursive Autoencoders Network (RAE) for sentence level sentiment classification, which obtains a reduced dimensional vector representation for a sentence. Later on, sentences of varying lengths and induces a feature graph over the sentence that is capable of explicitly capturing short and long-range relations. The aim of a sentence model is to analyze and represent the semantic content of a sentence for the purposes of classification or generation. The sentence modelling problem is at the core of many tasks involving a degree of natural language comprehension, e.g. sentiment analysis, paraphrase detection, entailment recognition, summarization, discourse analysis, machine translation, grounded language learning, etc.

Dos Santos and Gatti [8] proposed a Character to Sentence CNN (CharSCNN) model. CharSCNN uses two convolutional layers to extract relevant features from words and sentences of any size to perform sentiment analysis of short texts. Guggilla et al. [4] presented an LSTM- and CNN-based deep neural network model, which utilizes word2 vec and linguistic embeddings for claim classification (classifying sentences as factual or emotional). Huang et al. [19] proposed to encode the syntactic knowledge (e.g., part-of-speech tags)in a tree structured LSTM to enhance phrase and sentence representation. Akhtar et al. [16] employed several multi-layer perceptron based ensemble models for fine-gained sentiment classification of financial microblogs and news. Qian et al. [22] presented a linguistically regularized LSTM for the task. The proposed model incorporates linguistic resources such as sentiment lexicon, negation words and intensity words into the LSTM in order to more accurately capture the sentiment effect in sentences.

Nobata et al. [20] used a Vowpal Wabbits regression model and NLP features to detect hate speech on online user comments from two domains which outperforms a state-of the-art deep learning approach. Their features are divided into four classes: N-grams, linguistic, syntactic and distributional semantics.

Dadvar et al. [9] describe a method used to detect bullying users on YouTube. They used a multi-criteria evaluation system to obtain a better understanding of YouTube users' behaviour and their characteristics through expert knowledge. Based on experts' knowledge, the system assigns a score to the users, which represents their level of "bulliness" based on the history of their activities.

As the number of categories increases, the accuracy level that can be reached decreases, because it is more difficult to find the right category. This is even more relevant in situations where the category numbers are not scaled (as in SST1), but each number has a different meaning, similar to our work.

Chkroun and Azaria [5], [6] have developed Safebot, a chatbot system that converses with humans. This system allows humans to teach it how to reply to new statements (this is similar to [1], [7]). Safebot uses human feedback to identify offensive behavior. When Safebot is told that it said something offensive, it apologizes and adds the offensive sentence to its database. It then avoids using such sentences again. There has also been work on deceptive speech detection [2], [11].

III. Dataset Details

In the previous section we describe related work concerning sentiment classification. Most of the datasets used in previous related studies are based on comments about movies or services (e.g., movie review) or on forums or twitter posts. However, the text said by people or children may be different than such reviews or posts, because talking at home, in class or near friends, etc., can be different from the terms used in written text of comments or forums. This is especially true when considering children's conversations. Consequently, insulting context as well as language indicating threats may be different.

Given this difference, available on-line dataset resources of essays, comments and recommendations are not entirely appropriate for training an agent to enable it to determine types of spoken conversation.

Thus, we composed a new dataset to suit our needs. The sources of the sentences in our study are as follows:

1) An initial seed of 100 unintentional insulting sentences was obtained by performing interviews with parents of children with ASD, performed by the autism center, as described by [17].

TABLE I
DISTRIBUTION OF SENTENCE TYPES

Sentences Type	Count	Frequency	Average len	Vocab.
Normal sentences	2910	21.6%	6.5 (3.02)	2,611
Context-dependent[2]	2269	16.8%	6.95 (2.77)	1,957
Insulting third person	2644	19.6%	7.21 (3.33)	2,746
Insulting sentences	3511	25.9%	6.92 (3.12)	2,855
Indicating risk	2173	16.1%	7.78 (3.23)	1,467

2) Another group of sentences was provided by workers of MTurk [3], in response to our surveys, which are described in Appendix VII. The MTurk workers, who were located in USA, were asked to provide sentences for each of the following categories: insulting sentences, sentences which are context dependent, repetitive or strange sentences (which were associated to the non-insulting sentences), and sentences indicating risk. The survey explanation contained some examples of sentences, most taken from the initial seed described above. The payment per assignment was \$1, and we collected 83 assignments. In order to increase the number of sentences that indicated risk, we performed an additional survery, in which the MTurk workers were asked to provide 10 sentences indicating risk (only). The payment for each assignment was \$0.1, and we collected 51 such assignments. In total, 2170 sentences were gathered in this manner.
3) Some of the sentences were collected offline by students who were asked to provide relevant sentences.
4) Additional sentences were gathered from expert talks about safety, and in particular, safety of children with special needs.
5) An additional source was text from on-line groups and forums, concentrating on groups of children with special needs.
6) Other sentences were taken from news articles and from responses to news articles, where we collected sentences that can be said by children, or to a child.

Our dataset contains context relevant to children, and it is categorized into five categories: 'normal' sentences; insulting sentences, negative sentences about a third person; or sentences indicating risk that may indicate a dangerous situation for the special needs person, requiring immediate intervention and context-dependent sentences that are sentences with meanings that depend on the context of the situation in which they were told. Nonetheless, in the current study we did not consider the context of dependent sentences, since deciding about them requires additional information about the situation, rather than the text of the sentence itself.

The distribution of the sentence types and the details of the sentence lengths in the database (DB) are described in Table I. As can be seen, the average sentence length is very similar in the different sentence types. Following the above details, we continue with a description of the preprocessing algorithms used for preparing the dataset before starting the type categorization process.

IV. DATASET PRE-PROCESSING

In order to prepare our dataset for the different methods, we ran a preprocessing algorithm. Note that the categorization algorithms we used belonged to two groups: (a) classical machine learning algorithms implemented by the Scikit-learn Python library; (b) the Embedding-CNN method, as implemented by Keras, using the TensorFlow backend. For each of the groups, a different preprocessing algorithm was used: The reason for the need of different processes is as follows: The preprocessing algorithm for the classical Scikit-learn -based methods runs some generalizations on the sentences (from both the training set and the test set) words, and then a bag-of-words is created for each sentence, transformed to TD-IDF, and sent to the machine learning algorithm. In contrast, the algorithm for the Embedding-CNN method starts also with some, more simple, preprocessing of the sentences; then it uses Word2Vec based on Google news vector [10] for the embedding process, and the result of the Embedding process is sent to the CNN.

We should emphasize that the training set built in the preporcessing algorithm of the embedding+CNN method includes, in addition to 90% of the original sentences, also additional sentences that were used for the training set. Part of the sentences, which have a clear meaning (appear 10 times or more in the database, with a frequency of 75% or more for appearing in one of the types), were added to the training set as phrases, if they were not substrings of the sentences of the test set. In addition, phrases taken from the Movie Review (MR) database that were not substrings of the test set were also added to the training set. MR phrases that had neutral or positive marks were categorized as *normal* sentences, and MR phrases with a negative meaning were categorized as *third person insulting sentences*. Enlarging the training set was essential for the CNN method, since the number of weights needed to train is huge, thus a much larger set for training these weights is necessary.

We proceed by describing the details of the Machine Learning methods used in this study.

V. METHODS DESCRIPTION

The first method we used is the Extra-Tree method (which stands for extremely randomized trees) that was proposed in [21], with the main objective of further randomizing tree building in the context of numerical input features, where the choice of the optimal cut-point is responsible for a large proportion of the variance of the induced tree.

The most successful method was the Random Forests. Random forests are bagged decision tree models. Each decision tree in the forest considers a random subset of features when forming questions and only has access to a random set of the training data points. This increases diversity in the forest leading to more robust overall predictions and the name random forest. In our study, the Random forest was based on 100 estimators, and as described below in Section VI, it reached the best results.

The KNN algorithm is a robust and versatile classifier that is often used as a benchmark for more complex classifiers such as Artificial Neural Networks (ANN) and Support Vector Machines (SVM). Despite its simplicity, KNN can perform better than more powerful classifiers and is used in a variety of applications such as economic forecasting, data compression and genetics.

A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane, which categorizes new examples. In two dimensional space this hyperplane is a line dividing a plane into two parts where each class lies on a different side of the hyperplane.

Ridge Classifier works similarly to LogisticRegression with l2 penalty, but it uses the Ridge regression model for multi-class classification in the following way to create a classifier: 1.Use label binarizer to create multi-output regression, one for each class (One-Vs-Rest modelling) and train the Ridge regression model. 2.Get prediction from each class' Ridge regression model (a real number for each class) and then use argmax to predict the class.

The Naive Bayesian classifier is based on Bayes theorem with the conditionally independent assumptions between features. A Naive Bayesian model is easy to build, with no complicated iterative parameter estimation which makes it particularly useful for very large datasets. Despite its simplicity, the Naive Bayesian classifier often does surprisingly well and is widely used because it often outperforms more sophisticated classification methods.

MultiLayer classifier implements a multi-layer perceptron (MLP) algorithm (a neural network). MLP is a supervised learning algorithm that learns a function by training on a dataset. Given a set of and a target, it learns a non-linear function approximation for either classification or regression. It is different from logistic regression, because there can be one or more non-linear layers, called hidden layers, between the input and the output layers.

MLP trains on two arrays: array X of size (n-samples, n-features), which holds the training samples represented as floating point feature vectors; and array y of size (n-samples,), which holds the target values (class labels) for the training samples. We used a network with three hidden layers, each containing 100 sigmoid nodes.

The Voting classifier trains all the above methods, and then for each sentence of the test set, performs a voting protocol over the above methods and chooses the category suggested by the majority. The methods used in the Voting classifier are: Random forest, Extra trees, K nearest neighbors, SVM, Ridge Classifier, Bayesian inference method, and MLP.

Next, we describe the template of the Convolutional Neural Network (CNN) used for our text classification task. CNN is a class of deep, feed-forward artificial neural networks (where there is no cycle connections between the nodes) and use a variation of multilayer perceptrons designed to utilize minimal preprocessing. These are inspired by animal visual cortex. In CNN the result of each convolution will dismiss when a special pattern is detected. By changing the size of the kernels and concatenating their outputs, allows the detection of patterns of variant sizes (2, 3, or 5 adjacent words). Patterns could be expressions (word ngrams) like I hate, very good and therefore CNNs can identify them in the sentence notetheless to their position. The structure of the CNN used is taken from [18], where a CNN template for classification is suggested, and their template reached the best result for our database. In this model, the first convolution layer used had a filter length of 5 and an used ReLU as its activation function. The next part is a maxPooling layer, followed by a dropout of 0.2. Next, two additional convolutional and maxPooling layers, followed by a simple layer with 128 neurons and a ReLU activation function, and finally, a softmax layer with one output for each category.

Given the above machine learning methods, we proceed with the presentation of our results.

VI. Results

First, we describe our results from testing classical machine learning methods, imported from the Scikit-learn library, on the sentence databases, using our preporcessing Algorithm. We ran 50 experiments, where in each of them, the sentence dataset was randomly split into a training set and a test set. Table VI presents our results. As depicted in the table,

method	**average and (std) accuracy**	**average and (std) F1 score**
Random Forests	0.710 (0.016)	0.708 (0.016)
Extra Trees	0.661 (0.016)	0.661 (0.017)
KNeighbors	0.549 (0.018)	0.547 (0.018)
SVM	0.680 (0.017)	0.674 (0.017)
Ridge Classifier	0.684 (0.017)	0.682 (0.017)
bayes	0.627 (0.018)	0.625 (0.018)
MultiLayer	0.638 (0.017)	0.638 (0.017)
Voting	0.712 (0.016)	0.711 (0.016)

the random forest method achieved the best results, with the ability to correctly predict the type (normal sentence, insulting sentence, third person insulting sentence, or sentence indicating risk) with 71% accuracy and 0.708 F1 score. Other successful methods, with very close performance, are the SVM (with 68% accuracy and 0.67 F1 score) and Ridge Classifier (with 68% accuracy and 0.68 F1 score). The Voting classifier reached solutions very close but slightly higher than that of the Random Forest method (average accuracy of 71.2% and average F1-score 0.711).

We proceed by describing the results of the Embedding-CNN method, described in Sections IV and V. The CNN was trained for 10 epochs, using Adam optimizer [13], with a batch size of 128. The average accuracy level on the test set was 69.6% (std 0.008) and the F1 score was 0.681 (std 0.008). Note that, as described in Section IV, the embedding-CNN method was trained on 90% of our conversation database, and in addition, phrases from movie reviews that were also used in the training set. With this combined training set, the accuracy of the CNN was higher than most of the machine learning methods, Nonetheless, a random forest method, with 100 estimators, and the Voting classifier, reached slightly higher results, while it required a smaller training set and shorter training time w.r.t. the Embedding-CNN method.

Finally, we checked whether a set of neural networks can achieve better results than a single network. Thus, we created a random generated panel of 10 CNN based classifiers. The structure of each classifier was as follows: after the embedding process, a 1-D convolutional level was used, with 128 filters and a softplus activation function. Then, a max pooling process was done, followed by a dropout of 10%. Then, another 1-D convolutional layer was used with 32 filters and a linear activation function followed by max pooling. Thereafter, a third 1-D convolutional layer was used, with 128 filters and hyperbolic tangent activation method + max pooling. Then, a flatten layer (size 128) with sigmoid activation function was used, and its outputs were sent to a softmax layer. The batch size was set to 64, and we ran 10 epoches.

Each classifier was trained on 90% of the training set, and we chose the best five classifiers, based on their accuracy on the validation set (the rest 10% of the training set). Then, we determined the type of each sentence by a vote between the five best classifiers, which we called the panel. This voting panel increased the accuracy and the F1 score of the classifications. In particular, after 50 runs, the average accuracy rose to 72.2% (std 0.009) with an F1 score of 0.714 (std 0.009), resulting in higher accuracy and F1 scores than that reached by each of the experts individually.

VII. Conclusions

In this study, we considered the problem of categorizing sentences said by or to children, in order to determine insulting sentences or sentences indicating risk told to or by the child.

We developed a dataset comprising sentences said daily, where were classified into 4 different categories: 'normal' sentences; insulting sentences; negative sentences about a third person; or sentences indicating risk. We tested the performance of several different machine learning techniques on the task of the dataset categorization learning, and we found that the best predictors among the machine learning methods were the Random forest method, with an accuracy of 71% and 0.708 F1 score , and the Embedded+CNN method, with a 69.6% accuracy and 0.681 F1-score.

Finally, we created a random generated panel of 10 CNN based systems, each trained with 90% of the training set, and we chose the 5 best systems as experts, checked according to the validation set (10% out of the training set). We found that the accuracy of the voting panel and the F1-score were 72.2% and 71.4, respectively, which is better than the other methods, and better than each individual CNN system.

Our results can be useful in developing an automated agent that will be attentive to the special child's social interactions, will detect insulting sentences that are said unintentionally or offensive sentences said to the child, and it will be able to suggest appropriate responses.

References

[1] Amos Azaria, Jayant Krishnamurthy, and Tom M Mitchell. Instructable intelligent personal agent. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[2] Amos Azaria, Ariella Richardson, and Sarit Kraus. An agent for deception detection in discussion based environments. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, pages 218–227. ACM, 2015.

[3] Michael Buhrmester, Tracy Kwang, and Samuel D Gosling. Amazon's mechanical turk: A new source of inexpensive, yet high-quality, data? *Perspectives on psychological science*, 6(1):3–5, 2011.

[4] I.Gurevych C. Guggilla, T. Miller. Cnn-and lstm-based claim classification in online user comments. In *In Proceedings of the International Conference on Computational Linguistics*, 2016.

[5] Merav Chkroun and Amos Azaria. "did i say something wrong?": Towards a safe collaborative chatbot. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[6] Merav Chkroun and Amos Azaria. Safebot: A safe collaborative chatbot. In *Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[7] Merav Chkroun and Amos Azaria. Lia: A virtual assistant that can be taught new commands by speech. *International Journal of Human–Computer Interaction*, pages 1–12, 2019.

[8] M. Gatti C.N.dos Santos. Deep convolutional neural networks for sentiment analysis for short texts. In *In Proceedings of the International Conference on Computational Linguistics*, 2014.

[9] Maral Dadvar, Rudolf Berend Trieschnigg, and Franciska MG de Jong. Expert knowledge for automatic detection of bullies in social networks. In *25th Benelux Conference on Artificial Intelligence, BNAIC 2013*, pages 57–64. TU Delft, 2013.

[10] Google. Word2vec googlenews binary database.

[11] Martin Graciarena, Elizabeth Shriberg, Andreas Stolcke, Frank Enos, Julia Hirschberg, and Sachin Kajarekar. Combining prosodic lexical and cepstral systems for deceptive speech detection. In *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, volume 1, pages I–I. IEEE, 2006.

[12] V. Ravi K. Ravi. A survey on opinion mining and sentiment analysis: Tasks, approaches and applications. *Knowledge-Based Systems*, 89:1446, 2015.

[13] D.P. Kingma and J. Ba. Adam: A method for stochastic optimization.

[14] C. Liebeskind and S. Liebeskind. Identifying abusive comments in hebrew facebook. In *ICSEE*, 2018.

[15] B. Liu L.Zhang, S. Wang. Deep learning for sentiment analysis:a survey. In *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discover*, 2018.

[16] D. Ghosal A.Ekbal M. Akhtar, A. Kumar and P. Bhattacharyya. A multilayer perceptron based ensemble technique for fine-grained financial sentiment analysis. In *In Proceedings of the Conference on Empirical Methods on Natural Language Processing*, 2017.

[17] R. Azoulay E. Ben-Izchak M. Zwilling D. A. Zachor M. Allouch, A. Azaria. Automatic detection of insulting sentences in conversation. In *IEEE International Conference on the Science of Electrical Engineering*, 2018.

[18] A. Maheshwari. Report on text classification using cnn, rnn & han, 2019.

[19] X. Zhu M.Huang, Q. Qian. Encoding syntactic knowledge in neural networks for sentiment classification. *ACM Transactions on Information Systems*, 35, 2005.

[20] Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. Abusive language detection in online user content. In *Proceedings of the 25th international conference on world wide web*, pages 145–153. International World Wide Web Conferences Steering Committee, 2016.

[21] D. Ernst P. Geurts and L. Wehenkel. Extremely randomized trees,. *Machine Learning*, 63:342, 2006.

[22] J. Lei Q. Qian, M. Huang and X. Zhu. Linguistically regularized lstm for sentiment classification. In *In Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2017.

[23] E.H.Huang A.Y.Ng and C.D.Manning R.Socher, J.Pennington. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *In Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2011.