Auto spam removal in instant chats

Eli Eliyahu, Elyasaf Schwer
Advisor: Dr. Lee-Ad Gottlieb

October 2018

Abstract

Our project regards instant messaging chats groups and provides a solution to the spam problem that is so prevalent in those groups. The goal is to provide a comprehensive and simple solution for the spam problem. We provide a Telegram bot that can be added to any Telegram supergroup to remove spam there using known classification algorithms. In addition, there is a web interface to manage various settings for each group. We show that there are good results on five different datasets which prove that our solution is viable. We utilize the various attributes of instant chats such as the time of the message and the limited number of senders in each group to maximize the successful prediction rate of spam messages.

1 Introduction

Today the usage of online chatting groups is very widespread and common.

People use those groups for a wide range of topics. E.g. commute sharing, studying groups, kindergarten groups and many more.

From our own experience, we noticed that there are a lot of spam messages in those groups which diverts the participants from the real topic of the group. In addition, the spam messages making it hard for the users to follow and participate in the discussion of the real topic and make them miss important messages which reduce the efficiency of the group. The act of maintaining the messages and preventing spam messages is a tiresome and time-consuming process for the admin. This problem grows as more participants are in the group.

Hence, considering the current advancement in artificial intelligence we decided to automate this process using deep learning techniques which will automatically remove unrelated messages.

In detail, we will have two classes: spam messages and good messages. To label each message to the matching group we will use Logistic Regression, Naive Bayes, Support vector machine and Neural Network models.

Furthermore, we will take to our advantage the characteristics of instant chats groups. Namely, we will observe the results by incorporating features such as sender id, sending time and more which will provide the system more certainty whether the message is spam or not.

2 Related works

- Spam detection in Twitter:
  The suspicious behaviors of spam accounts on Twitter is studied. The goal is to apply machine learning methods to automatically distinguish spam accounts from normal ones. Results show that the Bayesian classifier has the best overall performance in term of F-measure. The trained classifier is also applied to the entire data set. The result shows that the spam detection system can achieve 89% precision.

  http://ieeexplore.ieee.org/abstract/document/5741690/
  http://hughchristensen.co.uk/papers/socialNetworking/SECRYPT2010$_{W}ang.pdf$

- Better Naive Bayes classification for high-precision spam detection:
  In this paper, they proposed several improvements to the NB classifier that make it well suited to applications requiring high precision, such as spam filtering. Although the proposed techniques have been shown to be quite effective for NB, they are also applicable to other learners. Further investigation of their utility will be the subject of future work.

  http://onlinelibrary.wiley.com/doi/10.1002/spe.925/full
  https://pdfs.semanticscholar.org/8b18/16e7cdb305f40127ed672c860dddd282b32d.pdf

3 Failed approaches

We have made the following three attempts. They didn't give us bad results, but also have not improved what we have got without them:

- Word2vec.
- Adding sender's unique name as a word to the message before converting the message to a vector.
- Bi-grams.

4 Detailed description of our work

Our project consists of two parts:

- The research part where we test different machine learning models with different methods to store the messages.
- The application which consists of the web interface where the admin of a group can manage various settings of each of his groups and the Telegram bot.

4.1 Analyzing spam prediction using different machine learning techniques

To develop and research the best automatic spam prediction mechanism we used the research part in our project.

Here we examined four known machine learning classifiers: Support vector machine, Neural Network (hidden layers: 3500, 3500, 1500, 1000, 500), Logistic Regression and Naive Bayes. To get the classifiers to work the best possible with the least bugs we used a Python machine learning library named scikit-learn which implements the above-mentioned classifiers.

As the data sets for our researches we used instant group chats histories of Whatsapp groups that we participate in and manage. Thus we know the topic good enough to classify the messages correctly. To export the data we used the application ”Backup Text for Whatsapp” because the Whatsapp built-in chat export doesn't export all the history. Our datasets are:

- Ride sharing group history between Ariel to Ramat Gan.
  (11800 Good messages, 858 Spam messages)
- Ride sharing group history between Ariel to Karnei Shomron.
  (14622 Good messages, 803 Spam messages)
- The two above-mentioned ride sharing groups combined.
  (26422 Good messages, 1661 Spam messages)
- Computer Science group history of our class of 2015.
  (6545 Good messages, 4460 Spam messages)
- Ariel University synagogue group history.
  (6673 Good messages, 3077 Spam messages)

We saved each data set described above in four different methods:

- Each message consists only from the text the user sent.
- Each message consists from the text the user sent and the hour the message was sent.

- Each message consists from the text the user sent, the hour the message was sent and the unique name of the sender.
- Each message consists from the text the user sent and the unique name of the sender.

To train the models and to allow them to predict a message we converted each message to a vector using the Bag of Words method. We treated each word, each punctuation mark and each line break as a unique cell in the vector, i.e. a ”word”. To convert the messages that are saved with the hour they were sent we added an extra cell to the final vector that contained the value of the hour (1-24) and when the message included a sender it was appended to the message in plain text before the conversion to the vector.

Moreover, as detailed in the next section, we implemented other advanced algorithms such as Bi-grams and Word2Vec.

As described in the Evaluation and results section our researches yielded that the combination with the best results is the Neural Network classifier with each message saved as it's text and the hour it was sent.

4.2 The application

4.2.1 The web interface

To provide the admins a simple and intuitive way to manage their groups we developed the web interface that is integrated with the Telegram bot. Our website is based on CherryPy framework. We rely on CherryPy because it is now more than ten years old and it is has proven to be fast and reliable. It is being used in production by many sites including demanding ones.

The interface allows the admin to change the following settings:

- Activate or deactivate the identification and removal of spam messages in a group.
- Activate or deactivate the automatic label of messages as not spam in a group. (Read more at the bot part).
- View and edit the labeled messages so far for a group.

Once the user gets his password from a private chat with the bot he can log in to the web interface using his Telegram username as his web interface username and the password generated for him by the bot and choose the group to manage.

To secure the web interface we have done several things:

- The password that is generated is a combination of English letters and numbers and it's length is 8 characters.
- Once the bot generates the new password it sends it on a secure channel via telegram private chat.

- The password isn't stored in the database, instead, we store the output of Argon2 algorithm on the password. When the user authenticates his password again passes through the Argon2 algorithm then this output is compared with the one that is stored in the database. We use Argon2 because it's a key derivation function that was selected as the winner of the Password Hashing Competition in July 2015. Argon2 maximizes resistance to GPU cracking attacks making it harder for hackers to run a brute force attack.

- After the user authenticates successfully we store his session only for 60 minutes.

- We use a secure connection to the web interface thanks to the built-in support in CherryPy of SSL.

4.2.2 The Telegram bot

Once the bot is added to a Telegram supergroup with admin privileges it's ready to work. Our communication with the bot and the groups that it participates in is through the Telegram Bot API.

By default, the bot only watches admin's label messages in a group, namely the labels the admin gives for a message (by replying to a message with 0 for not spam and 1 for spam and for the removal of the spam message) and then removes the label message the admin sent.

As our tests suggest a group must have more than 1000 spam messages and more than 1000 good messages to achieve a good prediction by the bot. To make the process of achieving such a number of labels the admin can use the web interface to enable an option to make the bot label automatically all messages as good messages and from time to time when the admin sees a spam message he can label it.

There is also available a private chat with the bot where the admin can get help about the usage of our system and he can require the bot to generate a new password to authenticate to the web interface. More about that in the web interface section.

Based on our tests the bot uses the Neural Network model to identify spam messages.

Because of the long training time of the model we defined a ”down time”. It's a fixed time that is the bot will iterate all the groups he is in and retrain the Neural Network models using the messages labels that were added that day. By retraining at a down time we ensure less load on the server and that there is a low probability for new messages to be added and waiting 24 hours until they will be included in the retraining.

5 Evaluation and results

Group 1 - Free ride sharing - Ariel Ramat Gan

MSG + TIME + SENDER

	RECALL	PRECISION	ACCURACY	F_MEASURE
MLP	0.966	0.966	0.95	0.966
GNB	0.953	0.953	0.953	0.953
LR	0.98	0.942	0.94	0.960
SVM	0.966	0.947	0.935	0.957

MSG + TIME

	RECALL	PRECISION	ACCURACY	F_MEASURE
MLP	0.96	0.96	0.94	0.96
GNB	0.94	0.959	0.925	0.949
LR	0.966	0.947	0.935	0.957
SVM	0.953	0.947	0.925	0.950

MSG + SENDER

	RECALL	PRECISION	ACCURACY	F_MEASURE
MLP	0.98	0.942	0.94	0.96
GNB	0.806	0.945	0.82	0.87
LR	0.953	0.922	0.905	0.937
SVM	0.966	0.947	0.935	0.957

MSG

	RECALL	PRECISION	ACCURACY	F_MEASURE
MLP	0.953	0.959	0.935	0.956
GNB	0.933	0.958	0.92	0.945
LR	0.966	0.947	0.935	0.957
SVM	0.953	0.947	0.925	0.950

Group 2 - Free ride sharing - Ariel Karnei Shomron

MSG + TIME + SENDER

	RECALL	PRECISION	ACCURACY	F_MEASURE
MLP	0.975	0.952	0.937	0.964
GNB	0.975	0.936	0.922	0.955
LR	0.996	0.929	0.931	0.961
SVM	0.966	0.926	0.928	0.960

MSG + TIME

	RECALL	PRECISION	ACCURACY	F_MEASURE
MLP	0.986	0.962	0.955	0.974
GNB	0.996	0.935	0.937	0.964
LR	0.996	0.929	0.931	0.961
SVM	0.996	0.926	0.928	0.960

MSG + SENDER

	RECALL	PRECISION	ACCURACY	F_MEASURE
MLP	0.975	0.952	0.937	0.964
GNB	0.972	0.942	0.925	0.957
LR	0.996	0.932	0.934	0.963
SVM	0.993	0.925	0.928	0.958

MSG

	RECALL	PRECISION	ACCURACY	F_MEASURE
MLP	0.996	0.929	0.931	0.961
GNB	0.986	0.962	0.955	0.974
LR	0.996	0.929	0.931	0.961
SVM	0.996	0.923	0.925	0.958

Group 3 - The two above-mentioned ride sharing groups combined

MSG + TIME + SENDER

	RECALL	PRECISION	ACCURACY	F_MEASURE
MLP	0.990	0.943	0.943	0.966
GNB	0.958	0.956	0.930	0.957
LR	0.990	0.943	0.943	0.966
SVM	0.988	0.941	0.940	0.964

MSG + TIME

	RECALL	PRECISION	ACCURACY	F_MEASURE
MLP	0.972	0.961	0.945	0.967
GNB	0.984	0.945	0.940	0.964
LR	0.988	0.941	0.940	0.964
SVM	0.986	0.947	0.943	0.966

MSG + SENDER

	RECALL	PRECISION	ACCURACY	F_MEASURE
MLP	0.984	0.96	0.953	0.971
GNB	0.958	0.956	0.930	0.957
LR	0.990	0.941	0.941	0.965
SVM	0.981	0.945	0.938	0.963

MSG

	RECALL	PRECISION	ACCURACY	F_MEASURE
MLP	0.972	0.961	0.945	0.967
GNB	0.984	0.937	0.932	0.96
LR	0.988	0.941	0.940	0.964
SVM	0.986	0.945	0.941	0.965

Group 4 - Computer Science group history of our class of 2015

MSG + TIME + SENDER

	RECALL	PRECISION	ACCURACY	F_MEASURE
MLP	0.846	0.947	0.837	0.894
GNB	0.4	0.909	0.481	0.555
LR	0.836	0.940	0.824	0.885
SVM	0.84	0.936	0.824	0.885

MSG + TIME

	RECALL	PRECISION	ACCURACY	F_MEASURE
MLP	0.91	0.91	0.854	0.91
GNB	0.403	0.909	0.483	0.558
LR	0.813	0.945	0.810	0.874
SVM	0.883	0.910	0.835	0.896

MSG + SENDER

	RECALL	PRECISION	ACCURACY	F_MEASURE
MLP	0.88	0.910	0.832	0.894
GNB	0.4	0.909	0.481	0.555
LR	0.846	0.947	0.837	0.894
SVM	0.853	0.937	0.835	0.893

MSG

	RECALL	PRECISION	ACCURACY	F_MEASURE
MLP	0.913	0.901	0.848	0.907
GNB	0.403	0.909	0.483	0.558
LR	0.86	0.918	0.824	0.888
SVM	0.89	0.902	0.832	0.895

Group 5 - Ariel University synagogue

MSG + TIME + SENDER

	RECALL	PRECISION	ACCURACY	F_MEASURE
MLP	0.922	0.918	0.875	0.920
GNB	0.913	0.830	0.785	0.870
LR	0.927	0.914	0.875	0.920
SVM	0.840	0.934	0.828	0.885

MSG + TIME

	RECALL	PRECISION	ACCURACY	F_MEASURE
MLP	0.913	0.939	0.885	0.926
GNB	0.913	0.834	0.789	0.872
LR	0.886	0.924	0.853	0.904
SVM	0.913	0.934	0.882	0.924

MSG + SENDER

	RECALL	PRECISION	ACCURACY	F_MEASURE
MLP	0.95	0.92	0.896	0.935
GNB	0.913	0.830	0.758	0.870
LR	0.945	0.916	0.889	0.930
SVM	0.931	0.923	0.885	0.927

MSG

	RECALL	PRECISION	ACCURACY	F_MEASURE
MLP	0.918	0.935	0.885	0.926
GNB	0.913	0.834	0.789	0.872
LR	0.918	0.930	0.882	0.924
SVM	0.868	0.927	0.842	0.896

Cross-validation

Results on 10 random distributions for train and test (F-measure) : 0.909, 0.895, 0.920, 0.913, 0.868, 0.880, 0.889, 0.879, 0.899, 0.917, 0.935.

Word2Vec Result (MSG only)

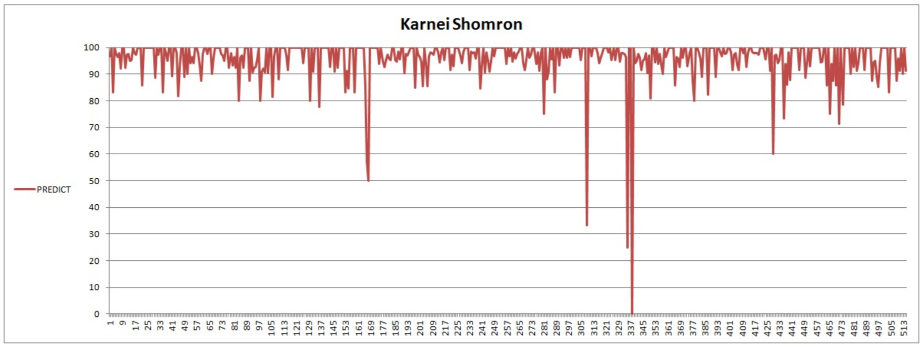
	RECALL	**PRECISION**	**ACCURACY**	**F_MEASURE**
GROUP 1				
MLP	0.949	0.993	0.955	0.971
GNB	0.973	0.973	0.96	0.973
GROUP 2				
MLP	0.819	1	0.835	0.9
GNB	0.82	0.946	0.805	0.879
GROUP 1 + 2				
MLP	0.67	0.99	0.78	0.818
GNB	0.588	0.96	0.645	0.73
GROUP 4				
MLP	0.8	0.75	0.657	0.77
GNB	0.78	0.977	0.771	0.87
GROUP 5				
MLP	0.849	0.806	0.727	0.827
GNB	0.845	0.493	0.51	0.62

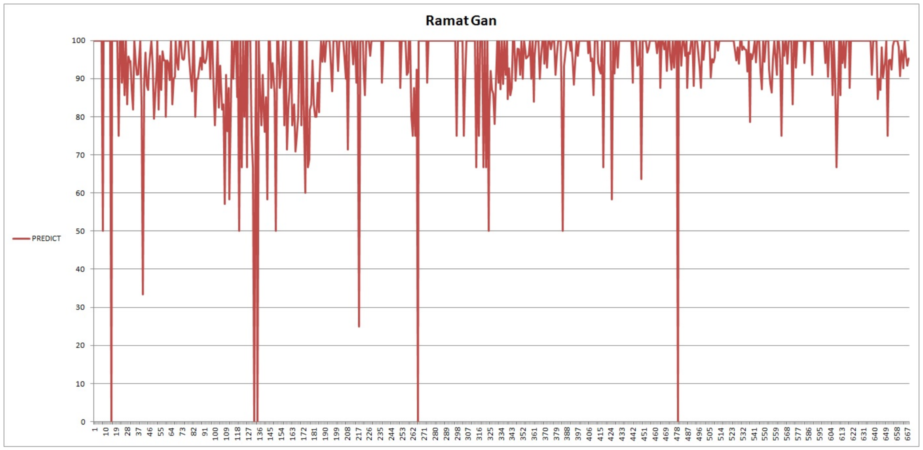
Bi-grams Result (MSG only)

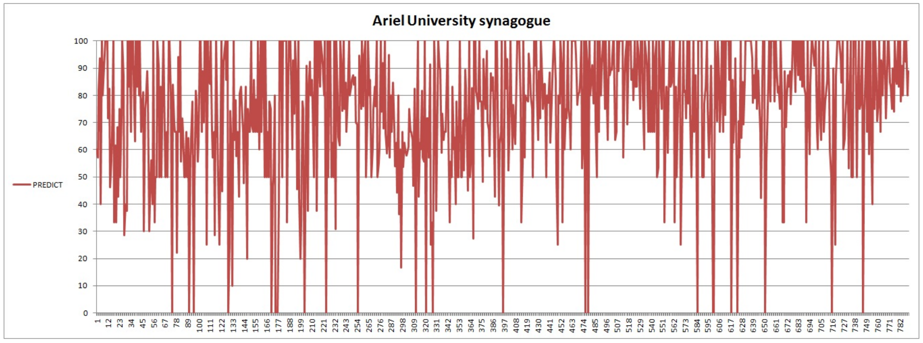
	RECALL	**PRECISION**	**ACCURACY**	**F_MEASURE**
GROUP 1				
MLP	0.966	0.917	0.91	0.941
GNB	0.873	0.916	0.845	0.894
LR	1.0	0.852	0.87	0.92
GROUP 2				
MLP	0.966	0.917	0.919	0.955
GNB	0.854	0.984	0.862	0.914
LR	1.0	0.900	0.904	0.947
GROUP 4				
MLP	0.959	0.854	0.839	0.903
GNB	0.940	0.844	0.817	0.890
LR	0.886	0.874	0.810	0.880
GROUP 5				
MLP	0.926	0.842	0.800	0.882
GNB	0.53	0.924	0.583	0.673
LR	0.596	0.913	0.627	0.721

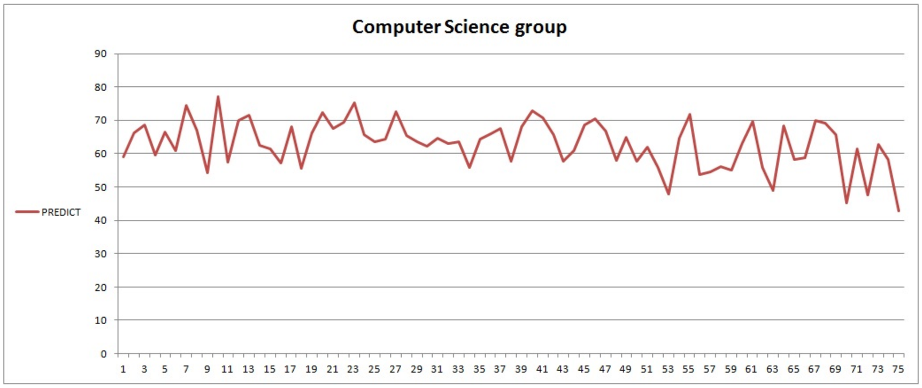
5.1 How many days of manual classification is needed in order to filter spam automatically with a high success rate?

In fact, after one day there may be good results. But as suggested in the following graphs, at days where the spam pertains a new subject the spam prediction of that day is relatively very low. The following graphs show the percentage of the predictions success rate each day given all the days before it as the dataset.



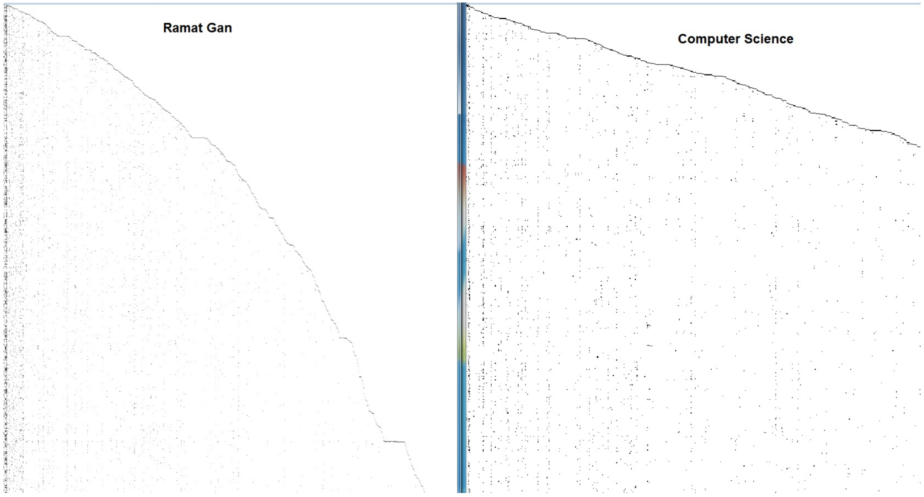






As seen in the graphs, the groups of free rides sharing have the greatest prediction success rate and in groups with broader subjects there is no stability of the successful predictions rate.

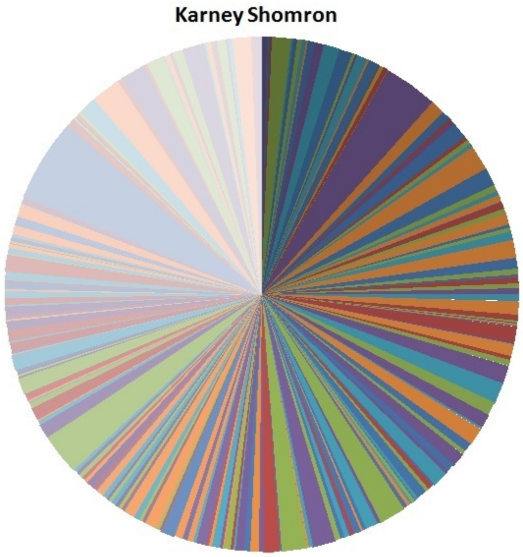
In addition, this can be seen by converting a vector matrix into a binary image. (Line vectors) -

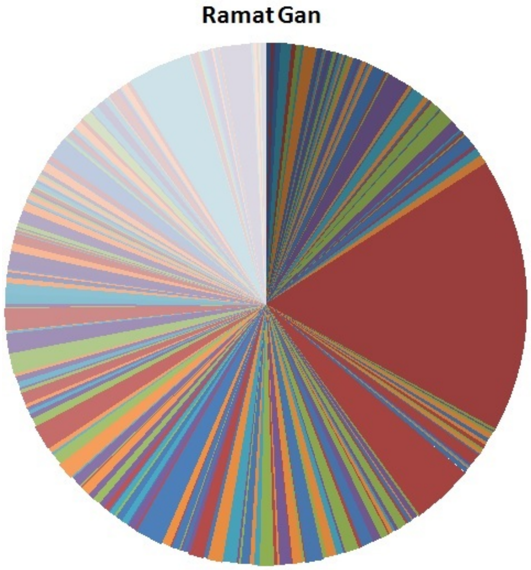


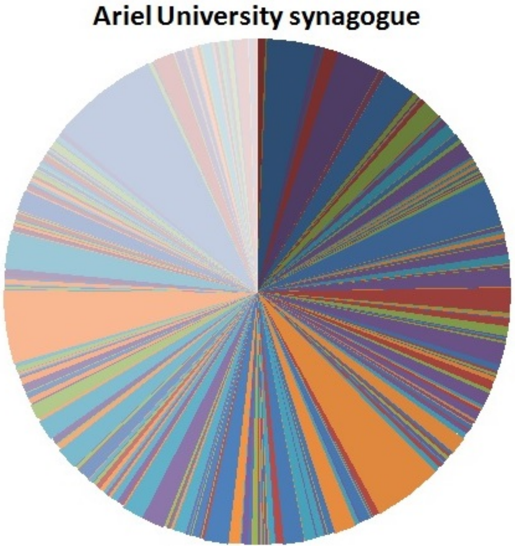
It can be seen that at the group of ride sharing between Ramat Gan and Ariel the initial subject remained as it was at the beginning, so there is a thick vertical line to the left, whereas in computer science the subjects change and there are more black points along all the width.

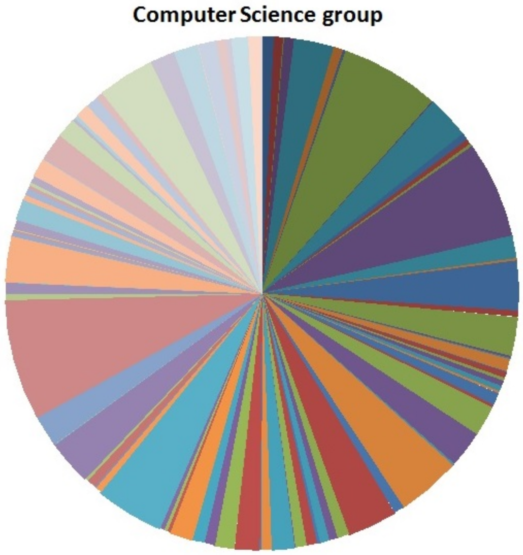
5.2 Utilizing the limited number of group participants

We've tested the option to add the sender's name as a Feature to the learning models. As can be seen in the results, it helped only in the Ramat Gan group. The following is a division of spam messages by group users. It turns out that spam is usually widespread among all users.









6 Conclusions

After we tested the four methods to store a message with the different machine learning algorithms we achieved a spam detection solution with a high success rate which proves that it can be of value in our application. Along with the web interface, we provide a comprehensive and simple solution for anyone to enjoy spam free chats that after sometime require no maintenance from the admin, thus saving him hours of manual spam removal.

7 Future work

- Migrate the train and prediction to Tensforflow code to make the spam identification and training faster by running the code on the GPU.
- Test Adaboost algorithm on our datasets.
- Convert the messages to vector using another method.
- Add an option to track users by their spam message amount and ban them from the group if they spam too much.