The Random Forest Classifier is implemented in random_forest/

* generate_meta_data.py generates the metadata

* classify_emails.py takes the generated metadata, categorises it into training/testing/validation and implements the random forest classifier

The K-Means Clustering Algorithm is implemented in k-means_tfidf/

* top_terms.py scores the importance of words in emails using tfidf

* plot_top_terms.py uses k-means to plot the top terms

The Naive Bayes Classifier is implemented in bayes/

* generate_meta_data.py generates the metadata

* calculate_statistics.py calculates the statistics behind the metadata features

* classify.py uses the calculated statistics to determine the probability of an email belonging to either human or nonhuman by comparing its metadata attributes to the overall statistics using a Gaussian Probability Density Function

The Parallel Convolutional Neural Network is implemented in parallel_cnn/

* prepare_data.py finds the most frequently used words and assigns them a score, this score will then be used to vectorise the emails so they can be fed into the neural network

* classification_network.py implements the neural network and performs a few more modifications on the data

Introduction

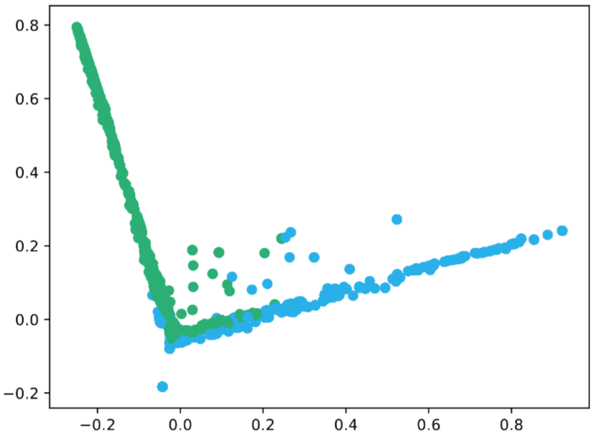
Many email categorisation applications focus on filtering spam & phishing emails from legitimate emails. Any further email organisation & categorisation is left to the user. This project aimed to build a neural network capable of categorising non-spam human emails – correspondences between people – from non-spam nonhuman emails – receipts, bills, reminders, registration confirmations, and so on. To tackle this, a labelled dataset of 818 human emails and 935 nonhuman emails were collected from my student email address and sorted into training, validation, and testing datasets in an 80 to 10 to 10 ratios, respectively.

Methods & Results

Because this is an application I want implemented on email servers, the privacy of the user's emails was a concern. As such, my first attempt at categorising the emails was done through the metadata of the emails. I collected the number of lines, word count, character count, number of letters, number of digits, and number of special characters of each email. I chose these metadata elements because they are easy for the email service provider to extract and send to third-party applications without putting the user's privacy at risk.

The metadata of the emails in the training dataset was used to train a random forest classifier. If the metadata between human and nonhuman emails were starkly different, the random forest classifier should be able to find a set of rules that would allow each email to be categorised as such. Unfortunately, whether I trained the random forest classifier with 10 or 10,000 decision trees (and many in-between), I was not able to hit a 20% accuracy in sorting the emails. So, one or more of the following happened: I did not have enough emails for the random forest classifier to be able to find distinguishing features in the provided metadata, there simply aren't distinguishing features between human and nonhuman emails in the chosen metadata, and/or I chose the incorrect metadata features to examine.

But, this failure lead to the next question: if there aren't distinguishing metadata features between human and nonhuman emails, are there distinguishing features in the contents of the emails? To examine this, I used a k-means clustering algorithm (with k = 2) to cluster the words found in an email based on how important they are to their respective category. To find the importance of the word to its respective category, the term frequency-inverse document frequency (td-idf) – a numerical statistic that reflects the importance of a word in a group of documents – of all the words in an email category was calculated. These scores were then fed through the k-means clustering algorithm to produce the graph seen below.



The green points represent the human scores while the blue points represent the nonhuman scores. This graph shows that there is a difference in the contents of both the human and nonhuman emails. However, what was most interesting was that their differences seemed to converge in the lower-left corner, indicating that some human and nonhuman emails may be indistinguishable from each other based on content alone. With a small dataset of just over 1,700 emails, it could explain why the random forest classifier had trouble distinguishing between the emails if a good portion of the human- and nonhuman-labelled emails seemed too similar.

That said, the random forest classifier looked at metadata while the k-means algorithm looked at the content frequency of the emails, so one does not necessarily relate to the other. To check how similar (or dissimilar) human and nonhuman emails were (and to hopefully find a way to classify them), the previously collected metadata was fed through a Naïve Bayes classifier that calculated the probability of a given email's metadata's attribute's value to belong to either the human class or nonhuman class using a Gaussian Probability Density Function. The overall classification accuracy was 57.2%, which was not bad, but it wasn't great either. Interestingly, human emails were correctly classified about 80% of the time while nonhuman emails were correctly classified only about 30% of the time. So, there either was something about the metadata that didn't allow for distinguishing nonhuman emails from human emails or not enough emails were collected for a true difference to reveal itself.

At this point, I decided to put the metadata aside and focus on classifying the emails based on the frequency of the words that appear in them. A glance at the top terms of each category shows the top ten terms for humans include seven words and three special characters. In contrast, the top ten terms for nonhumans only include two words, one of which is “https”, and the remaining eight are special characters.

To categorise emails based on word frequency, all the words in each email were replaced with a score representing their frequencies. A word limit was set on the emails. Any email over the word limit was cut at the word limit and any email under the word limit was padded to reach the word limit. This converted the emails into a vectorised form that can be read by a neural network. These vectorised email forms were then batched and fed through a parallel convolutional neural network (P-CNN).

A parallel convolutional network (as opposed to a sequential one) was chosen so that the relations between words in phrases of different sizes can be computed. Each layer of the P-CNN is given a kernel size. This kernel size represents the number of words a phrase would have, which the convolutional layer would explore phrases of that length. For example, a layer with kernel size of two would extract the features of every two words while a kernel size of five would extract the features of every five words. The outputs of these convolutional layers were then put through a ReLU nonlinearity followed by max pooling before being concatenated and fed through a dropout layer, connected fully in a linear layer, and put through a sigmoidal function to generate a value between 0 and 1, each extreme representing either nonhuman or human.

The network was trained and tested with varying numbers of convolutional layers, kernel sizes, and number of kernels with the SGD and Adam optimisers with different learning rates, momentums (when applicable), weight decays (when applicable), and decaying learning rates. However, no matter the parameters or optimiser, a similar pattern kept showing itself: the training accuracy and loss would hover around a number and stay there with barely any learning (a few times the network seemed to learn in the first few epochs but then it would either unlearn or spastically hover around a specific accuracy or loss, as if in confusion). At best, the network had a 54% accuracy. At worst, it had a 46% accuracy. It was not able to learn the differences between human and nonhuman emails based on word frequency and their relations.

Discussion

The Naïve Bayes classifier used earlier had the best accuracy of 57.2% when compared to the other two tried categorisation methods: a random forest classifier (> 20% accuracy) and a parallel convolutional neural network (54% accuracy at best; 46% accuracy at worst).

Future considerations in human versus nonhuman email categorisation can explore combining the probabilities from the Naïve Bayes classifier with the remaining metadata in the random forest classifier to hopefully have a better classification accuracy. Retrieving more metadata information (such as the number of uppercase/lowercase words, average length of sentences, etc.) would also aid the random forest classifier to better classify the emails.

Another possibility can explore the use of an enconder (the first half of an autoencoder) to generate an encoded 'signature' for human and nonhuman emails, respectively. This may lead to interesting (and hopefully successful!) results in classifying human and nonhuman emails.

Finally, combing a long short-term memory (LSTM) or gated recurrent unit (GRU) in a recurrent neural network with a convolutional neural network may allow for better classification of emails.

Unfortunately, due to time and deadline restrictions, the previously mentioned considerations were not explored. That said, whatever method is ultimately chosen, being able to attach the network to an email server to continuously learn would be the most ideal scenario. A significant reason for the poor performance of the networks appeared to be due to the small dataset of emails. Having access to more emails for both humans and nonhumans would be immensely helpful in better training any of the previous neural networks.

To add to all that, increasing the categories (for example, adding sub-categories for the human and nonhuman emails) would be expected to help in classifying the different emails as it gives more specificity to the different kinds of emails. The graph produced through the k-means clustering algorithm showed that the human and nonhuman emails exist on a spectrum – it would be interesting to explore what sort of emails are balancing on the boundary of human and nonhuman and what sort of emails are absolutely human or absolutely nonhuman.