USC - ISI* – Analyzing Users within Organizations

Mohit Surana, Mudit Bhargava, Rohini Kapoor, and Saranya Krishnakumar

University of Massachusetts Amherst

696DS Final Report

1 Abstract

Social engineering ("phishing") attacks are a major threat to the security of governments, organizations, and individuals. Particularly dangerous are attacks launched from within an organization by compromised user email accounts. In this work we explore unsupervised learning techniques to build representations of the typical behavior of users in an organization which could form the basis for detecting abnormal behavior from a compromised email account. We come up with different generative and discriminative neural network architectures and evaluate them using metrics like Hierarchy Prediction, Hits@K, etc. We observe that while some models are able to learn about network and hierarchy, other models tend to learn more about users and their behavior

2 Introduction

One of the most basic ways to understand a user's behavior within an organization is by using his/her designations for e.g (CEO, VP, Manager etc). The problem with this approach is that designations and their responsibilities are specific to the organization and industrial sector. For instance a Vice-President in one small company may not have the same roles and responsibilities as a Vice President in large multinational company. Another approach is to use Social Network Analysis based methods[1][2] for understanding user behavior. Although these methods have been successfully used in the past for anomalous links and hierarchy detection, we believe, that internal compromised email accounts is an intrinsically harder task and cannot ignore the textual content of email exchanges that happen within an organization.

Email contents are widely used by spam filters. Although spam filters classify emails as spam/not-spam, they have not proven to work well for detection of compromised accounts. This happens because spam filters utilize generic features for spam emails (e.g words/phrases often used in spam like lottery, double your, earn etc), but attacks from compromised user accounts are often hand-engineered, intelligent and targeted towards a specific recipient. Hence its necessary to learn from email contents in the context of a specific sender/receiver.

This has been studied in the past using probabilistic graphical models[3]. Although these models learn with minimal data requirements, inference and learning in such models is generally hard. With the availability of large quantities of unstructured text and recent research and success of neural network models and word embeddings[4] [5] on NLP related tasks, we attempt to solve the above problem using neural networks.

In all our models, users and emails are represented as fixed size embeddings, where email embeddings are extracted from pre-trained word vectors; and user embeddings are randomly initialized and learned via back-propagation while training. For the baseline, we build a multi-layer neural network that takes an input a

*This was an industry collaboration project where we were mentored by Marjorie Freedman and Ryan Gabbard at the University of Southern California - Information Sciences Institute.
We were also advised by John Lalor at the University of Massachusetts Amherst

sender and receiver embedding and tries to predict an email embedding as close as possible to the actual email embedding. A major drawback of the above approach is that email embeddings are calculated by averaging word embeddings of all the words present in the emails and predictions are only made based on sender and receiver representations. The model also assumes that embeddings for different emails between a sender and receiver are similar. To address the above problem, inspired by paragraph vectors[5], we came up with another model where the input to model is a sender embedding, receiver embedding, $word_{j-1}$ embedding and $word_{j+1}$ embedding and the model tries to predict $word_j$ embedding. Here the model uses the sender, receiver and current context in the email to predict the word embedding

Although such a generative approach has the potential to model communication style between a pair of users, these models are harder to learn. To address the above issue, we also experiment with a discriminative approach, where the model takes an input a sender embedding, receiver embedding and email embedding and tries to predict if the mail was sent between sender and receiver or not. More details and updates on models are presented in the subsequent sections.

3 Dataset

We explored the above problem on the Enron email dataset. The Enron dataset, collected as a part of the Enron scandal in 2001, is the largest real world email dataset which contains roughly half a million email exchanges between Enron employees. The dataset in its raw form is very noisy and contains special characters, duplicates, multiple email-ids etc. Over the years data has been trimmed down due to privacy issues and a lot of effort has been made to clean and organize the dataset since its release[6] [7]. To shift our focus away from the task of cleaning and onto the actual problem, we decided to use a pre-cleaned version of the dataset initially prepared by Shetty et al[6] which was later ported to a newer SQL version, and further refined by Hendrik [1]. The final dataset being used contains 250,000 emails from 150 core Enron members. A sample email is shown in Figure 1.

```
Message-ID: <30624101.1075862601659.JavaMail.evans@thyme>
Date: Mon, 26 Nov 2001 11:20:02 -0800 (PST)
From: chuck.kaniuka@ipgdirect.com
To: don.baughman@enron.com
Subject: Re:  TradersNews Energy-11/26
Mime-Version: 1.0
Content-Type: text/plain; charset=us-ascii
Content-Transfer-Encoding: 7bit
X-From: "Chuck Kaniuka" <chuck.kaniuka@ipgdirect.com>@ENRON
X-To: Baughman Jr., Don </O=ENRON/OU=NA/CN=RECIPIENTS/CN=DBAUGHM>
X-cc:
X-bcc:
X-Folder: \DBAUGHM (Non-Privileged)\Baughman Jr., Don\Inbox
X-Origin: Baughman-D
X-FileName: DBAUGHM (Non-Privileged).pst

Don:

I hope that all is well on your end (Enron/Dynegy merger?) & that your
holiday was great.  I'll be interested to hear about your hunting
fortunes!

As you suggested in our last conversation a few weeks ago, I'm
following up
with you shortly after Thanksgiving.  I just left you a Vmail message
to
continue the process in finalizing our win-win "business
arrangement/partnership" regarding the creation and trading of a TNE-
Cinergy
index on EOL. I look forward to hearing from you in the near future.
Thanks.

Chuck Kaniuka
Director-Sales
215-504-4288, ext. 217
```

Figure 1: A sample email from Enron dataset

Although the database took care of a lot of issues like email de-duplication, identity disambiguation we found that there were still many emails that contained threads and forward messages. These were cleaned

[1] http://www.ahschulz.de/enron-email-data/

up by filtering on keywords like 'Original Message', 'Forwarded by' etc. Apart from this, on using Stanford's pre-trained GloVe embeddings, we noticed that a lot of words were getting dropped (out of vocabulary for GloVe), leading to poor performance. This was happening because the email data contained words unusually segmented using arbitrary punctuation marks. To deal with this, we designed our custom cleaning logic but the problem persisted. Finally, we shifted to using the CoreNLP [8] parser's tokenizer which was able to convert sentences into a form more understandable by the word embedding model.

4 Related Work

In the past, many Social Network Analysis (SNA) and NLP based methods have been used on Enron dataset to study the email exchanges to infer elements such as user behavior and hierarchical structure of the organization. Chapanond et al[1] did spectral analysis of Enron dataset to learn its structural information, by constructing adjacency matrix from communication graphs. They used SVD on this matrix to conclude that there are 2 principle dimensions. By projecting the matrix into a 2 dimenthey formed clusters to form hierarchy.

Agarwal et al[2] used SNA to detect Enron's Organizational Hierarchy. Unlike the NLP based approach they were not relying on availability of email exchange between two employees for predicting dominance relation. By ranking degree centrality of every node in the network, they could predict dominance relation between pairs of employees with high accuracy. Their approach was restricted to pairs of employees related hierarchically in the gold standard and did not consider arbitrary pairs.

Although SNA based approaches try to discover different groups of nodes present within the social network, network properties alone are not adequate. For example, a tightly knit group of users who exchange emails frequently will appear to satisfy same role, but one of them might be a manager. Hence using the email content exchanged or topics distributions within email can help better predict the relation among employees.

This was pointed out by McCallum et al[3] in their paper on Author Representation model. The paper demonstrated that SNA based approaches mainly help to categorize the various nodes in network, but are not enough to model user behavior. They propose a probabilistic graphical model based approach that can group users based on their connections as well as the content of emails written by them. The Author Recipient Model, is an extension of LDA that learns topics distributions based on direction sensitive messages sent between entities, by conditioning the multinomial distribution over topics distinctly on both the author and the recipient of a message. This model also takes into account the social structure in which messages are exchanged. By using these person specific topic distributions, the model measures similarity between, and clusters people.

Inspired by their work and motivated by the success of neural networks in learning deeper non linear relationships, we propose novel neural network architectures that utilize email content as well as network structure to learn rich user representations in vector space. These can then be used in various tasks like hierarchy prediction and anomalous email detection.

5 Models & Approach

Below is the detailed description of our proposed models. As described before, in all the models users and emails are represented as fixed size embeddings, where email embeddings are extracted from pre-trained word vectors; and user embeddings are randomly initialized and learned via back-propagation while training

5.1 The Sender-Receiver(SR) based model

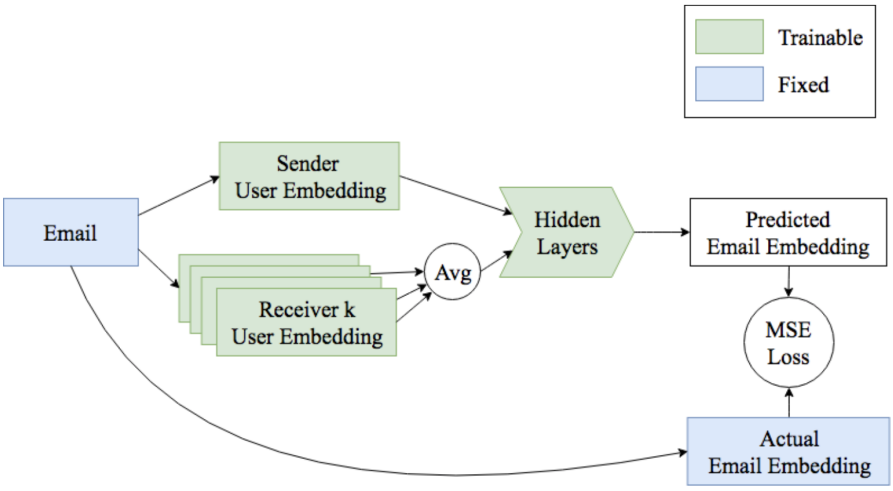

Figure 2: Sender-Receiver(SR) based model architecture

Inspired by the Author-Recipient Topic model[3], the baseline model attempts to learn sender and receiver embeddings that can predict email embedding close to actual email embeddings between them. The basic model architecture is shown in Figure 2. More details about the model are given below:

- For each email in the training data sender embedding and receiver embeddings (one email can have multiple receivers) are extracted. One sender-receiver embedding is prepared by concatenating the sender embedding with an average of the receiver embeddings.
- The concatenated sender-receiver embedding is passed through a multi-layer Neural Network which predicts the email embedding.
- The predicted email embedding is compared against the actual email embedding and loss is calculated using MSE loss function
- The actual email embedding is obtained by taking an average of word embeddings for each word in the email. Word embeddings are obtained by using a (pre-trained [9]/custom) word2vec model.
- The loss calculated on each training example is back propagated through the network all the way till sender and receiver embeddings. The embeddings and network weights are updated based on the calculated gradients

5.2 The Paragraph-Vector(PV) based model

For our next model, we use the word level learning of the sender and the receiver. This model is inspired by paragraph vector proposed by Le and Mikolov[5]. The basic architecture is shown in Figure 3.

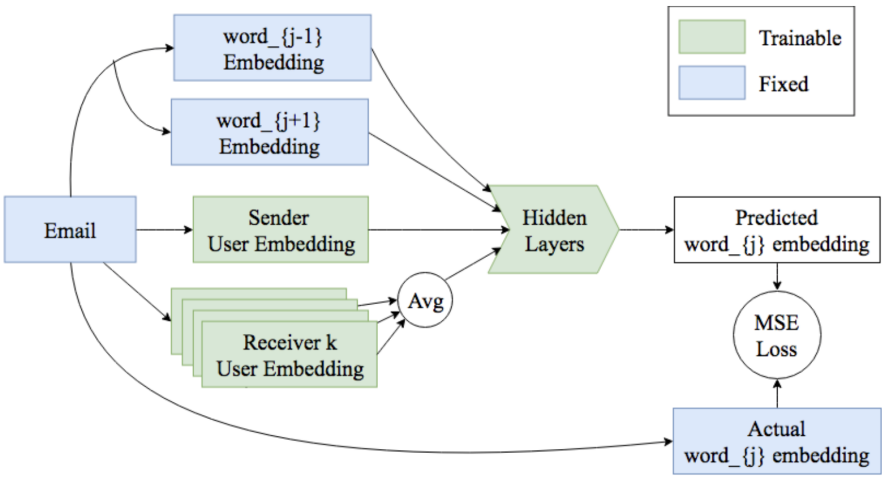

Figure 3: The Paragraph-Vector(PV) based model architecture

- Like the baseline model, for each email in the training data sender embedding and average receiver embeddings (one email can have multiple receivers) are extracted.
- For each email, for each word, we predict the word embedding, using the embeddings for the previous and the next word, along with the sender and receiver representations.
- For each word embedding prediction, we calculate the MSE loss, and use it to train the sender and receiver embeddings. The previous and next word embeddings are fixed during training.
- The training of the model is vectorized by creating a matrix of all concatenated valid combinations of previous and next word embeddings. Sender and receiver embeddings are concatenated with every row in this matrix via broadcasting. The matrix formed as a result is passed through the model and losses are computed. Broadcasting ensures that back-propagation updates the sender and receiver embeddings appropriately.

5.3 The Discriminative Model

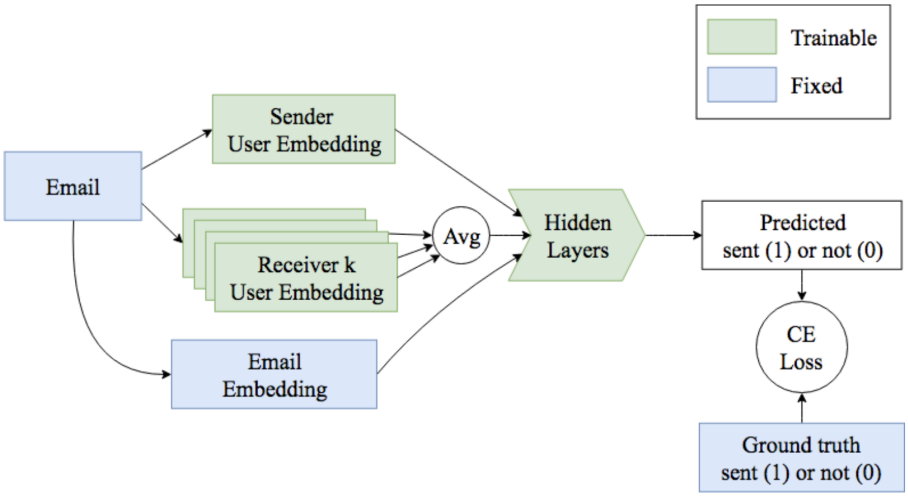

Figure 4: Discriminative model architecture

One other approach that we tried is to train a discriminative model. In this approach, for each sender receiver pair, we train the embeddings to be similar to the mail exchanged between them, and in parallel, also train them to be dissimilar to a mail not exchanged between them. The basic architecture is show in Figure 4.

The pseudo code is given below:

```
for every mail m in the dataset do
  for Every receiver r of m do
    s - sender of mail
    train the model with (s,r,m) with true label as 1
    randomly sample another message m' from the dataset, such that r was not receiver of m'
    train the model with (s,r,m') with true label as 0
  end for
end for
```

Algorithm 1: Discriminative Model

- For each email, we train the sender, receiver embeddings and that mail with the label 1.
- For now, for every positive example, we train just one negative example of mail for the sender receiver pair with label 0
- We trained a multi-layer Neural Network with this data, and used cross entropy loss.

5.4 Custom word vectors

An important aspect of all our models is the projection of ground truth emails words into embeddings of fixed size. We explored pre-trained models from Stanford GloVe[9] as well as trained our own custom word2vec model on the Enron dataset with the intuition that custom word2vec model may be able to learn more things specific to Enron vocabulary. Experimental results show a slight increase in performance of models on using custom trained word2vec models.

6 Evaluation and Results:

For model evaluation and results, we restrict ourselves to mails and interactions within the core 150 Enron members as we expect their representations to be much stronger and accurate.

For qualitative evaluation of trained models, we used t-SNE and basic clustering methods like Nearest Neighbors on the learned user embeddings. For quantitative evaluation on anomalous email detection, we consider metrics like Role Prediction, Hierarchy Prediction, Hits@K, and Mean Average Precision. We also experiment with a new evaluation metric where we model the losses observed for each user as a normal distribution and define thresholds for valid emails. The metrics along with their results are discussed in the section below

6.1 Qualitative Evaluation

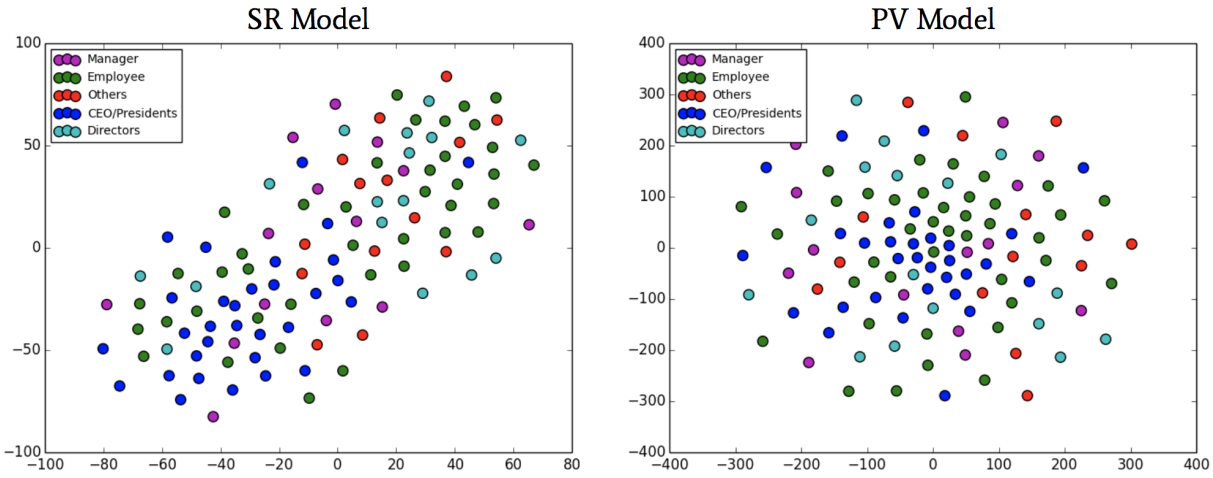

Figure 5: t-SNE plots for SR Model (on left) and PV Model (on right) color coded by user designation

We initially used qualitative evaluation metrics like 2D t-SNE and nearest neighbors on learned embeddings for evaluation of models. The intuition behind these metrics was that if the models learned well, they would be able to form logical clusters. It was soon found that qualitative analysis for this task was much harder than expected. With our limited domain knowledge (about Enron) it was difficult to understand what the embeddings were being grouped on. E.g the t-SNE plots color coded by designations (shown in figure 5) does not seem to show any logical clusters. Similarly, it was hard to understand why 2 people were being put close together by nearest neighbors without enough knowledge about them. Due to these difficulties we moved to other quantitative measures for performance evaluation of our models

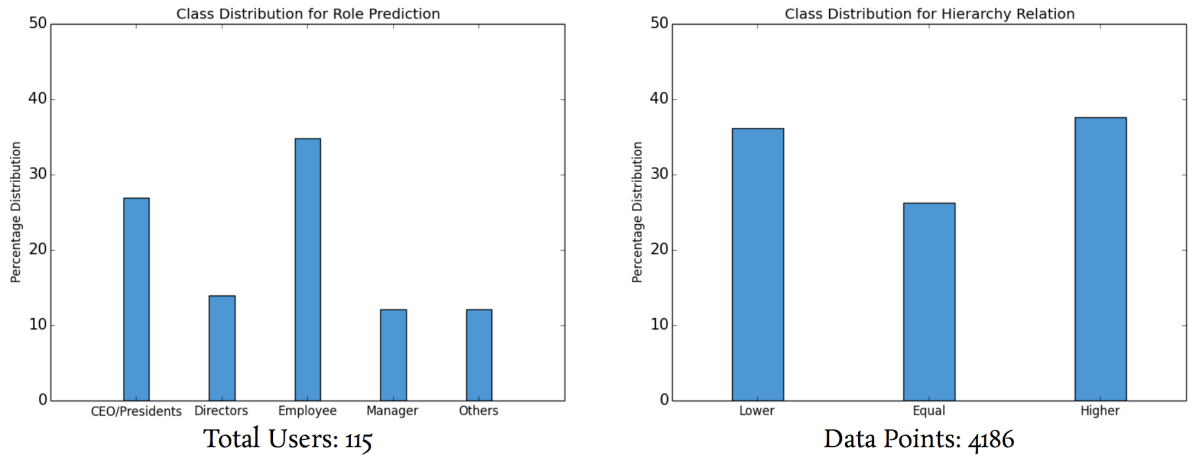

Figure 6: Class distribution for Role Prediction (left plot) and Hierarchy Prediction (right plot)

6.2 Role Prediction

To understand how good the user representations were, we trained a SVC classifier with an RBF kernel on learned user embeddings to predict designations (CEO/Directors/Employees/Managers/Others) provided to us in the dataset. The class distribution was unbalanced and is shown in figure 6, left plot. Since, there were only 115 users whose designations were known, we used a leave-5-out cross validation approach. The results obtained (shown in figure 7, left plot) were just slightly better than random baseline (35.6% based on most probable label). The SR model obtained the highest accuracy of 48.69%. Class-wise F-score analysis (figure 7, right plot) of the best performing model shows that model isn't learning much and is predicting everything as two most frequent roles i.e Employees & CEO/Presidents. We believe that the learning of vector representations in isolation from the extrinsic task does not help it generalize to the task of role prediction. Additionally, the fine grained prediction of roles is inherently a difficult task and it is even harder due to presence of very little data (115 designations).

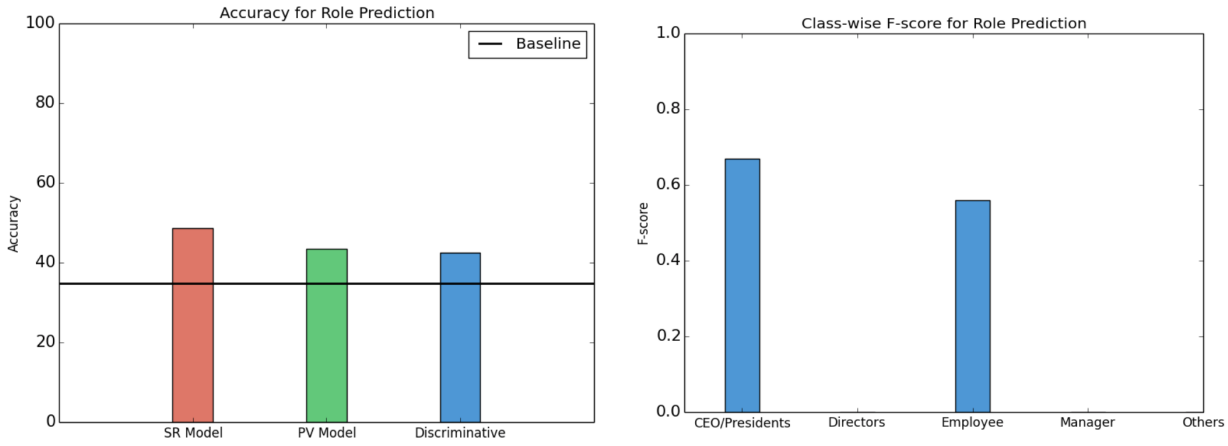

Figure 7: Results for role prediction: The plot on the left shows the accuracy obtained by the models. The black line denotes the baseline (based on most probable model). The plot on the right shows the F1 score per class for the best performing model (SR Model)

6.3 Hierarchy Prediction

As discussed in the section above, the task of role prediction may be dissimilar and too granular for what the models are trying to learn. Hence, we experiment with a higher level task of Hierarchy Prediction. Like in role prediction, learned embeddings are inputs to the task. Given a pair of learned embeddings (say A, B), we train a classifier to predict +1 (if designation(A) > designation(B)), -1(if designation(A) < designation(B)) and 0 (if designation(A) = designation(B)). All the learned embeddings are split into train and test using a 75/25 split and the results are reported on unseen test data. Figure 6, (right plot) shows an almost equal distribution of data across the classes. Figure 8, (left plot) shows the accuracy obtained by the models on Hierarchy Prediction task. The class-wise F1 scores of highest performing model (in figure 8, right plot) show that the model is able to do well on classes +1(higher) and -1(lower), but has a hard time with class 0(same). One possible reason is that designations where there seems no certain hierarchy (e.g traders v/s employees were encoded as 0)

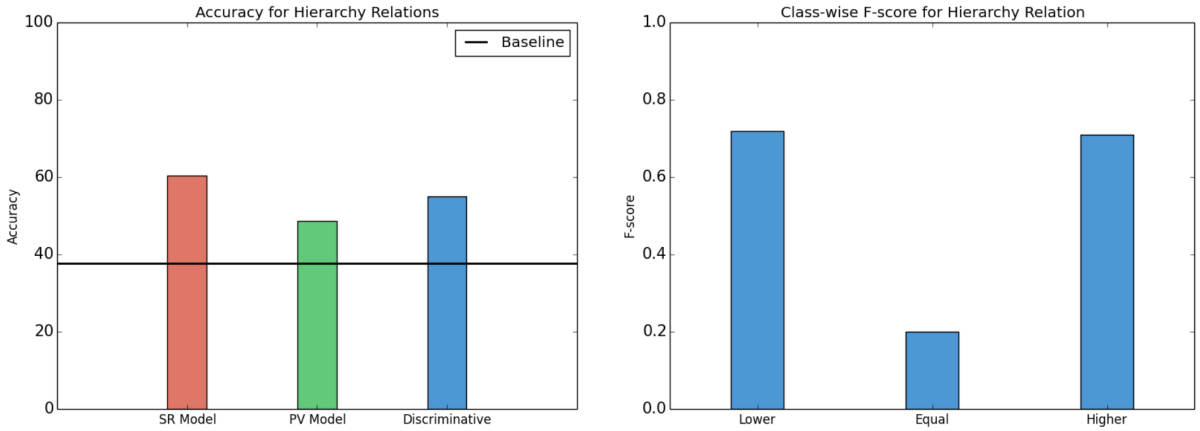

Figure 8: Results for Hierarchy Prediction: The plot on the left shows the accuracy obtained by the models. The black line denotes the baseline (based on most probable model). The plot on the right shows the F1 score per class for the best performing model (SR Model)

6.4 Anomalous Email Detection

6.4.1 Hits@K & AP@K

We first split the emails into training (70%), test(20%) and validation(10%) sets based on the time stamp of the mails received. This is similar to how we expect our model to be used in a real world scenario, where inference about a new mail is done by looking at behaviour exhibited in past mails.

After training the models, we generate negative examples to test how well our model identifies anomalous emails. This is done by replacing the original email contents by another email not sent between the same sender receiver pair. Both the valid emails and the negative examples are then passed through the SR and PV models to obtain the MSE loss and discriminative model to obtain the probabilities.

Using the (average) error from the SR and PV models, and the prediction probabilities from the discriminative model, we rank the real and the fake emails to see how well we are able to discriminate between them and formalize the results through metrics like Hits at K and Mean Average Precision across sender-receiver pairs.

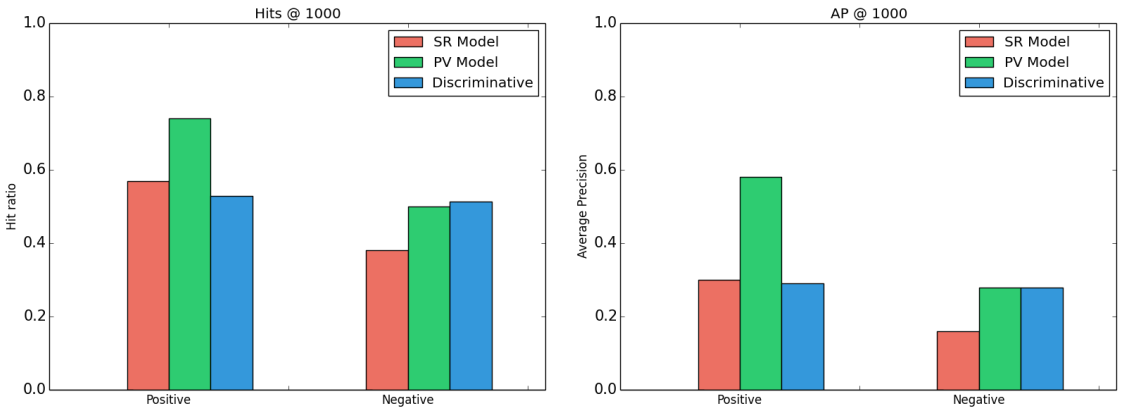

Figure 9: Results of evaluation using the top and bottom 1000 emails when ranked by the MSE loss or probability. We expect the top 1000 to be valid emails and bottom 1000 to be invalid emails

We see that the PV model outperforms the other models on the valid emails from our dataset. Whereas the performance on the negatively sampled emails is not as good as that on the valid mails. The discriminative model detects invalid emails better than the other models and we attribute this to the fact that this model has already been trained with the objective of predicting negative emails.

6.4.2 Issues with the negative emails sampled

The negative sampling method employed involves using valid emails, with the sender and receivers permuted. This is not a good method to create legitimate negative examples and because it is often the case that generic emails are picked up which may not be far from a legitimate looking mail.

Another end of the spectrum is to use some kind of document similarity heuristics to pick similar or very different examples but these are also not good indicators of the kinds of emails that we are trying to flag as anomalous. This weakens our evaluation metrics for our models as well.

To decouple the negative sampling from the evaluation process, we decided to move ahead with a few other approaches to detect anomalous emails.

6.4.3 Ranking all senders for each mail

Given a mail, we replace the actual sender of the mail with all the possible senders from our email corpus. After passing these mails through our models, we get a list of errors that correspond to how likely it is that the given sender had actually sent the mail. We then see how may times we can accurately return the original sender among a list of top K senders.

Model / K	5	10	20	30
SR Model	11%	18%	29%	39%
PV Model	21%	29%	41%	50%
Discriminative Model	8%	15%	24%	31%

Of the three models, the PV model performs the best, but the results are not satisfactory. We see that even in the top 30 (of 150) users, we are not able to accurately predict the original sender. This happens because the errors across different users are not normalized. The weights learnt by the model cause some users to have small errors for all their mails whereas other users end up having large errors for even valid mails.

6.4.4 Modeling loss as Normal Distribution

To tackle the problem of normalization of errors across users, we studied the losses found in the mails sent by users as shown in figures 10 and 11.

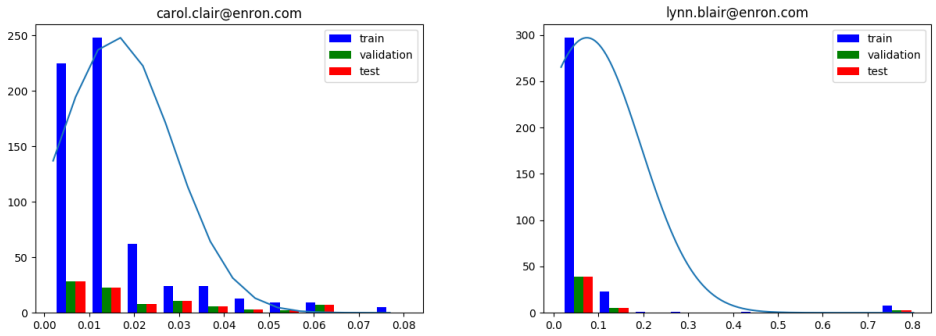

Figure 10: Errors for Sender-Receiver(SR) based model

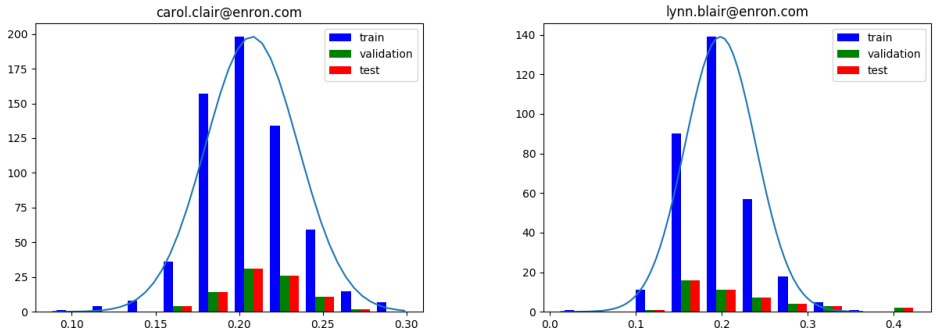

Figure 11: Errors for Paragraph-Vector(PV) based model

The losses resemble a normal distribution with a person characterized by the mean loss in their training mails and diversity in conversations is captured by the standard deviation in their errors. It is worth noting that this generalizes to their mails from the validation and test set as well.

We then need to fix a threshold on the standard deviation beyond which which we flag emails to be invalid. To do this, we plotted the percentage of training and validation emails that correctly get classified as we vary the threshold and this is shown in figure 12.

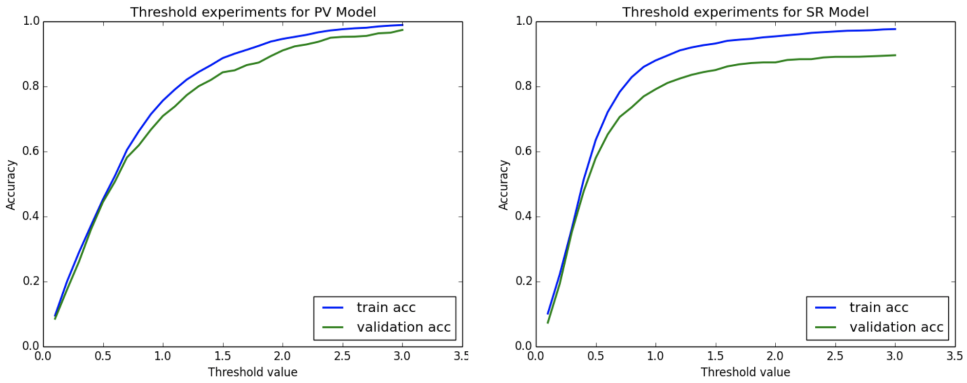

Figure 12: Threshold (co-efficient of standard deviation) and the corresponding % of mails predicted as valid

We target a validation accuracy of 85% and based on our plots, fix the thresholds for the PV Model as 1.5 and 1.7 for the SR Model. These thresholds achieve test accuracies of 85% each and we thus conclude that the threshold generalizes to the test data as well.

The next step is to see how well this generalizes to anomalous mails. We picked two random employees and studied their typical mail conversations manually. Based on these mails, we hand-engineered four mails to showcase how our models perform in the task of anomalous email detection as shown in Figure 13:

- Valid email from within the dataset - should be flagged as valid
- Organization based broadcast email - should be flagged as valid
- Generic spam email - should be flagged as invalid
- Personalized organization targeted illegitimate email - should be flagged as invalid

Original Email

To: mary@enron.com

enclosed is the counter party form that we have come up with for multi-product guarantys . depending upon the products that need to be covered ,each guaranty will need to be modified to make sure that it describes the proper entities and products . i will send along the enron form of this shortly but please pass this along to the appropriate people in the credit group . jeff , stacy and harry , are you going to change the physical product form of guaranty to look more like this ?power has changed their form and it may help for us to all be consistent especially when we are asked to amend an existing guaranty to include more products . if you do decide to amend your form ,please send the new form to credit . thanks for everyone s input .

Carol

Spam Email

Subject: geta degree from home, Mas#ters, Bachelors or PHD

Call 413 276 4583 toinquire about our degree programs.

Whether you areseeking a Bachelors, Masters, Ph.D. or MBA

We can provide youwith the fully verifiable credentials to get your career BACK ONTRACK!

No testing orcoursework required Call: 413 343 5653

we are sorry if youdid not want to receive this mail.

To be removed fromour list please call 413 345 6787

Targeted Email

To :bill.fowler@enron.com |tanya@enron.com

enclosed is a list of documents that are for review. i will be out of town for the next couple of days. Please go through them and send me the review on my personal id - carol.cdc@gmail.com. feel free to share any thoughts or concerns with me.

Carol Clair

Broadcast Email

To :linda.ward@enron.com |cynthia.rivers@enron.com|amy.mulligan@enron.com| tracy.minter@enron.com| christine.mcevoy@enron.com| albert.hernandez@enron.com| john.buchanan@enron.com| terry.kowalke@enron.com

For those of you that confirmed for the Sunday welcome event, we will be sending out more details in the future.

Figure 13: Snippets from original, targeted, spam and broadcast emails

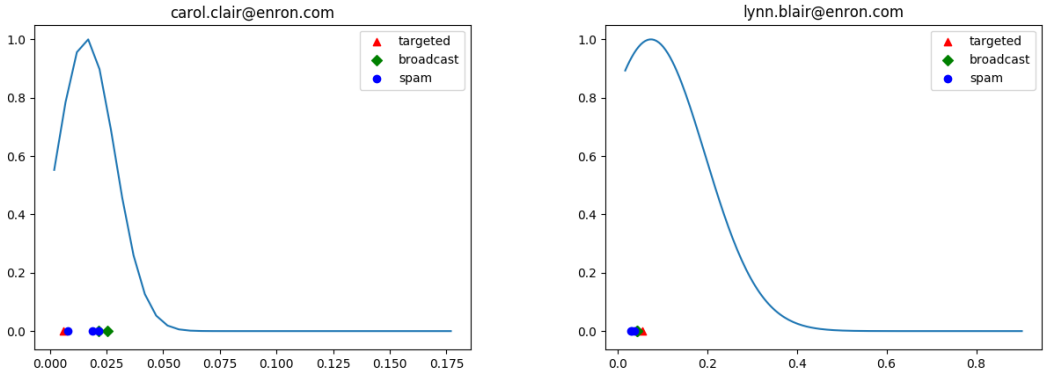

Figure 14: Errors for anomalous mails for Sender-Receiver(SR) based model

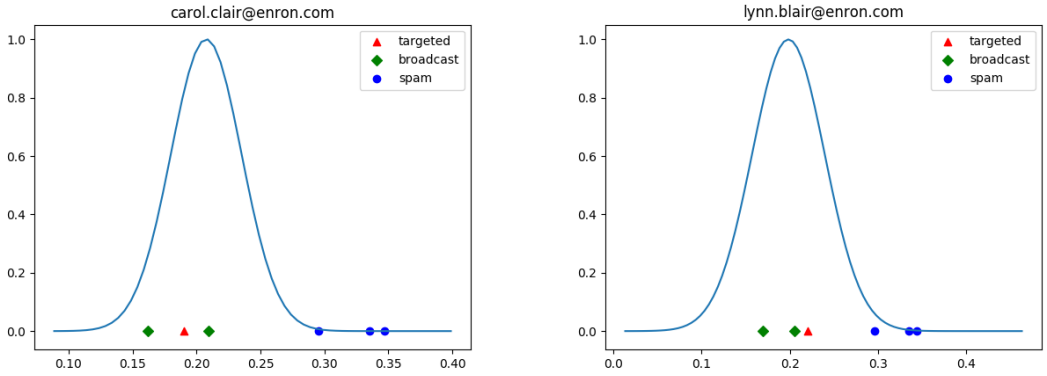

Figure 15: Errors for anomalous mails for Paragraph-Vector(PV) based model

We notice the following results:

- Organization based broadcast email - should be valid and shows very small deviation for both models
- Generic spam email - should be invalid and but is only detected as anomalous by the PV model and not by the SR model
- Personalized organization targeted illegitimate email - should be flagged as invalid but it is difficult for the classifiers to predict this

Although this metric showed promise and did deliver in the context of valid and spam email detection, it was unable to accurately disambiguate the targeted malicious emails.

7 Conclusion

In the paper, we explored different generative and discriminative Neural Network architectures and conducted numerous experiments on the Enron dataset. Each model its own strengths and weaknesses with no single model performing the best on all the tasks.

The SR and Discriminative models perform well on the hierarchy and role prediction tasks due to their structural similarity with SNA based methods. The PV model did not do well on these tasks because the learned user embeddings probably capture the idiosyncrasies exhibited in the form of user specific writing styles which may not be indicative of the kind of position a person holds. The PV model, however, does performs better on the task of anomalous mail detection. Instead of looking at an average of all the words in an email, which collapses onto a single point as emails get longer, it iterates over the mail, word-by-word and can characterize the words typically used in email exchanges between specific pairs of users. It was also observed that the models perform slightly better when custom trained word2vec is used instead of pre-trained Stanford glove

Due to the difficulties in sampling negative emails for evaluation metrics, we propose a new metric where we model training loss for each user as a normal distribution and define user specific thresholds (based on standard deviation) within which all valid emails should lie. This is able to correctly classify valid and spam emails but is unable to identify a targeted malicious attack.

8 Future Work

In the future, we would like to take this work forward by looking at different stages of the pipeline from a new perspective.

The Enron dataset contained a lot of messy data which was often a hurdle to efficient learning of our models. Despite multiple rounds of cleaning, we still see scope for further refinement cleaning e.g (html text, special characters). Handling these will ensure that the emails are correctly parsed and strong email word embeddings are obtained.

Another major limiting factor was the unavailability of anomalous emails. If such a dataset is available, we believe the models and the evaluation metrics (like Hits@k and Map) will give a better picture. On the same lines we would want to work on evaluation metrics that are independent of anomalous emails, for e.g our approach on modeling errors as user specific normal distribution

Also, our models currently do not address the cold start problem i.e users with low number of training mails do not have strong embeddings associated with them. One way to make the model more robust to sparsity of emails is to learn generic common representations for the Enron employees and incorporate that as a stronger prior instead of using randomly initialized embeddings for new users.

Acknowledgements

We are grateful to Marjorie Freedman and Ryan Gabbard for their continual guidance and support in the form of ideas and constructive feedback. We thank our course instructor, Prof. Andrew McCallum for giving us this opportunity to build upon the theoretical concepts learnt in class through this industry mentorship project. We also wish to thank Nicholas Monath and John Lalor, who provided us with invaluable feedback and suggestions regarding how to take our project forward, time after time.

References

[1] Mukkai S. Krishnamoorthy Anrut Chapanond and Bulent Yener. Graph theoretic and spectral analysis of enron email data. 2005.

[2] Apoorv Agarwal, Adinoyi Omuya, Aaron Harnly, and Owen Rambow. A comprehensive gold standard for the enron organizational hierarchy. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*. Association for Computational Linguistics, 2012.

[3] Andrew McCallum, Andrés Corrada-Emmanuel, and Xuerui Wang. The author-recipient-topic model for topic and role discovery in social networks, with application to enron and academic email. In *Workshop on Link Analysis, Counterterrorism and Security*, 2005.

[4] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[5] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *International Conference on Machine Learning*, 2014.

[6] Jitesh Shetty and Jafar Adibi. The enron email dataset database schema and brief statistical report. *Information sciences institute technical report, University of Southern California*, 4(1), 2004.

[7] Yingjie Zhou, Mark Goldberg, Malik Magdon-Ismail, and Al Wallace. Strategies for cleaning organizational emails with an application to enron email dataset. In *5th Conf. of North American Association for Computational Social and Organizational Science*, 2007.

[8] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, 2014.

[9] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2014.