Text Classification: Uncovering Relationships through Hillary's E-mails

Randolph Hill (rwhill2@illinois.edu), Xiao Li (xli147@illinois.edu)

University of Illinois at Urbana-Champaign
Department of Computer Science

Abstract

This paper explores the use of Linear Regression and Probabilistic learning methods for text classification on Hillary Clinton's publically available e-mails. This project aims to extract meaningful and impactful data from her e-mails with a goal of modeling relationships between Clinton and her contacts based on her conversational patterns. Several techniques are explored for modeling these relationships while the precision is compared over several machine learning algorithm with results supporting the theoretical proposal.

1 Introduction

With the upcoming election, there has been a great deal of controversy surrounding many of the prospective presidential candidates. One of the most prominent ongoing discussions is Hillary Clinton and her use of personal e-mail accounts on a non-government, privately maintained e-mail server when conducting official business during her tenure as United States Secretary of State. In accordance to the National Security Agency, Clinton has agreed to make a wealth of these e-mails public domain. The availability of such content provides a rich set of analysis potential and has opened new doors for modeling public figures analytically in ways that were not previously possible.

Text categorization has become one of the key techniques for handling and organizing text data such as e-mails [6]. In recent years it has been shown that applying Machine learning techniques are efficient means for discovering patterns in data as it does not require specifying and updating any set of rules as opposed to methods such as knowledge engineering. There a several approaches to machine learning such as supervised, semi-supervised, and unsupervised learning - each of which are a construction of algorithms that can learn from and make predictions on data [3].

The fundamental goal of this project is to apply machine learning and text analysis concepts over Clinton's e-mails to extract meaningful and potentially impactful facts that model relationships between Clinton and her e-mail recipients. Given that her personal e-mails are by nature "private" it's expected that the communication tone between her friends and colleges will be different than that of the general public. By analyzing these e-mails, we propose that it's possible to uncover relationship types (eg. Friend or Employee) based on the e-mail content.

This paper is organized as follows: Section 1 gives and introduction and clearly defines the problem that this project is approaching. Section 2 gives a general theoretical description on the feature formalization and machine learning algorithms used for text classification of the relationships. Section 3 gives an overview of the procedures and feature formalizations. Sections 4 and 5 provide an overview and comparison of the results and challenges. The final sections discuss related topics, and a final conclusion.

2 Task Definition and Approach

2.1 Task

Politicians and public figures have important ethical responsibilities when forming relationships, accepting donation and providing favors during their tenure. Any form of relationship between a high standing politician and a benefiting third party (companies, organizations, and other public figures) may pose potential conflicts of interests and biases while making important and impactful decisions. The current problem is that it is difficult to identify close relationships between public figures/politicians and benefiting third parties thus making it hard to identify conflicts of interest.

The overall problem that this project aims to solve is to identify relationships types based on the conversation style between Clinton and e-mail recipients. This project proposes that it's possible to uncover relationship types (eg. Friend or Employee) based on an e-mails content using text classification methods.

We aim to solve the defined issue by analyzing Clinton's e-mails using Text Classification in an attempt to uncover the relationship types based on the e-mails content. The idea is that conversation content changes based on the relationship between individuals. By modeling these conversational patterns, our task is to effectively classify the relationship Hillary has with an individual primarily based on the content of a message using the Naïve Bayes methods and Support Vector Machine classifiers.

2.2 Background

In this section, the formalism of Naïve Bayesian, Bayesian networks, SVM, and the basic methods for their development are reviewed.

2.2.1 Naïve Bayesian

The Naïve Bayes classifier was proposed as an ideal classifier for spam recognition in 1998 [3], which seamlessly transcends into the purposes of this project. The Bayesian classifier deals with working on the dependent events and the probability of an event occurring in the future that can be detected from the previous occurring of the same event [3]. For instance, a friend category and co-worker category may have distinct reoccurring words. The friend category may emphasize the codewords which represent "fun", "memories" and "gossip", while the co-worker categories may emphasize the codewords which represent "meetings" and "schedule". With a given collection of training samples the classifier can learning the probability of an unseen class based on the occurrence of previously seen key words. The categorization decision of a class is made by made by equation 1 [7].

$$c^* = arg\max_c p(c)\prod_{n=1}^{M} p(w_n|c) \quad (1)$$

2.2.2 Bayes Networks

Bayesian networks are directed acyclic graphs that represent joint probability distributions over a set of random variables. They provide a useful representation of the joint distribution of this set of variables, and expose ways to utilize their dependencies to perform statistical inference [7]. Bayesian networks represent ways to utilize their dependencies to perform statistical inference where each node in a Bayesian network corresponds to a random variable in the domain and the directed edges between the nodes each correspond to a node's parents' influence on that node [7]. Unlike Naïve Bayes, Bayes networks makes an assumption of conditional independence as shown in **Equation 2**.

$$X_i: \left(X_i - NonDecendantas_{X_i} \middle| P_a(X_i)\right) \quad (2)$$

With these assumptions, the Bayes Network classifier has the same decision rule as Naïve Bayes show in equation 1.

2.2.3 SVM

Theoretical studies show that SVMs acknowledge the particular properties of text: (1) high dimensional feature spaces, (2) most of the features are relevant (dense concept vector), and (3) sparse instance vectors [5]. As such, SMV is naturally a desired classification method for this project.

Proposed by Vapnik in 1998, Support Vector Machine (SVM) is model representation of examples as points in space for separable and non-separable data [5]. Specifically, SVM finds a hyperplane that separates a set of training data over which has the shortest weight vector while maintaining a maximum margin. This translates into the following optimization problem [4]:

$$Minimize: \quad ||\vec{w}|| \tag{3}$$

$$Such\ that: \quad \forall_i : y_i[\vec{w} \cdot \vec{d}_i + b] \geq 1 \tag{4}$$

The optimization problem from above is difficult to handle numerically. Vapnik proposed Lagrange multipliers to translate the problem into an equivalent quadratic optimization problem [3]:

$$Minimize: \quad -\sum_{i=1}^{n} \alpha_i + \frac{1}{2}\sum_{j=1}^{n} \alpha_i \alpha_i y_i y_j \vec{d}_i \vec{d}_j \tag{5}$$

$$Such\ that: \quad \sum_{i=1}^{n} \alpha_i y_i = 0 : \forall_i : \alpha_i \geq 0 \tag{6}$$

3 Methodology

3.1 Procedures

Using the provided dataset, the first task performed was querying the SQLite database to get e-mails sent from Hillary and count the number of e-mails for each recipient. Because most recipients are famous people, we can categorize them by their roles. The next step performed was cleaning and extract the relevant features from the data. We use the Java NLP library CoreNLP maintained by the Stanford NLP Group to tokenize the e-mail subject and e-mail body. Stop words, irrelevant numbers, and special characters were removed from the test dataset. Relevant features are extracted from the e-mail subject, e-mail body, and the time when the e-mail was sent. Features are formatted in an ARFF (Attribute-Relation File Format) file, an ASCII text file that describes a list of instances sharing a set of attributes. Finally, we used the machine learning tool Weka and apply a series of classifiers, including SVM, Naïve Bayes, and Bayes Net, and analyze the results.

3.2 Training and Test Data

The data set is provided by the Kaggle website, in the format of SQLite database and CSV files. The data is generally pre-cleaned and normalized into a collection of tables including "Persons", "Email Recipients", "Aliases" and "Emails". The data set contains 7945 e-mails, with 2363 e-mails sent from Hillary and 5582 received by Hillary from over 418 people.

To generate the labels, we ranked the recipient s by the number of e-mails they received from Hillary. We selected the top 10 recipients who received more than 25 e-mails and did research to identify the roles of these individuals. Based on the role of these recipients, we labeled them as *FRIEND*, *ASSISTANT*, *ADVISOR*, and *SECRETARY*, as shown in Figure 1.

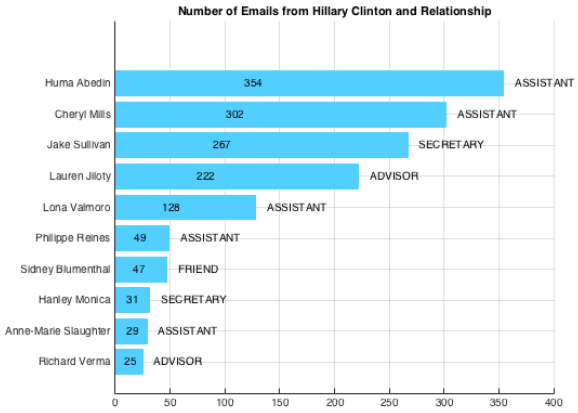

Figure 1: E-mail Counts and Relationship Labels

We selected e-mails that received by recipients listed in Figure 1 from the SQLite database and inserted them into a new table along with the label. E-mails with empty e-mail body were excluded. We used data in this newly created table as our training and test data.

3.3 Feature Selection and Representation

All e-mails were processed and extracted into a set of features stored in the ARFF file format. A feature vector is composed of various words from the dictionaries formed by extracting words from all e-mail subjects and e-mail body and the time when

the e-mail was received. In our case, we only consider words that occur in three or more e-mail subjects or e-mail body as features. This prevents misspelled words and words used rarely from appearing in the dictionary.

3.3.1 Data Cleaning

The provided data contained many characters that can be considered noise. Prior to extracting features, a list of stop words were identified and removed. Stop words are common words in a language such as *the*, *is*, *at*, and *on* [2]. These words do not add any value for the purposes of this project. Additionally, special characters, numbers, and URL links were removed. In all, more than 500 stop words were identified and removed from the evaluated e-mails.

3.3.2 Bag of Words

The bag-of-words (BoW) approach was used in processing the individual string tokens in all e-mails. BoW is a simplified representation used in natural language processing and information retrieval (IR) [2]. In this model, a text (such as a sentence or a document) is represented as the bag (multiset) of its words, disregarding grammar and even word order but keeping multiplicity [2]. This model was applied to the subject and body data sets identified in **Section 3.2.** Specifically, each e-mail subject and e-mail body was processed and tokenized. These tokens were placed into two dictionaries, one containing words occurred in three or more e-mail subjects and the other one containing words occurred in three or more e-mail body. Each word in the dictionary represents a feature.

3.3.3 Term Frequency & IDF Representation

We use Term Frequency-Inverse Document Frequency (TF-IDF) of words in e-mail subject and e-mail body as features. That is, the *i*th component of the feature vector is the TF-IDF word w_i in that e-mail [2]. TF-IDF is the term frequency of word, the number of time w_i appears in an e-mail, multiplied by the inverse document frequency or IDF as defined in **Equation 1**:

$$IDF(w_i) = \log\left(\frac{|D|}{DF(w_i)}\right) \quad (1)$$

, where D is the number of e-mails, and the document frequency DF(w_i) is the number of times that word w_i occurs in all the e-mails [4].

3.3.4 Time as a Feature

One important aspect in modeling a relationship is the time at which communication takes place. Representing a specific timestamp as a feature is not possible, so we use time in 4-hour segments starting from 12:00am instead:

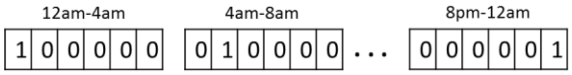

Figure 2: Time Representation

Similarly, we represented the days of weeks as features using 7 bits:

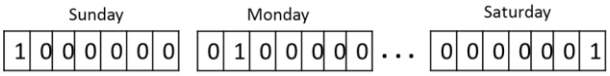

Figure 3: Week Representation

4 Results

Figure 4 shows the comparison of values of the correctly and incorrectly classified instances for each classifier we ran. As can be seen, the SVM classifier has the highest percentage of correctly classified instances of 76.27%. The Bayesian network classifier produces comparable results of 72.35%. The Naïve Bayes classifier had the lowest accuracy of 68.09%.

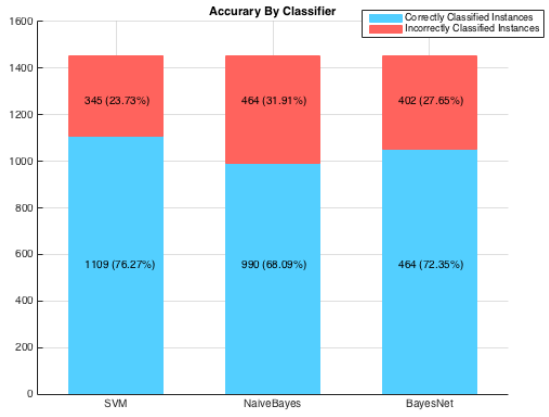

Figure 4: Accuracy by Classifier

To assess classifier accuracy, a confusion matrix is created for each category, as shown in Table 1, Table 2, and Table 3. AS represents the category `ASSISTANT`; *AS* represents the category *`ASSISTANT`*; *S* represents *`SECRETARY`*; *AD* represents *`ADVISOR`*; and *F* represents *`FRIEND`*.

	AS	S	AD	F
ASSIS-TANT	782	71	37	3
SECRE-TARY	149	137	2	4
ADVI-SOR	54	5	162	1
FRIEND	11	5	3	28

Table 1: Confusion Matrix (SVM)

	AS	S	AD	F
ASSIS-TANT	738	79	46	30
SECRE-TARY	194	70	12	16
ADVI-SOR	61	7	152	2
FRIEND	7	10	0	30

Table 2: Confusion Matrix (Naïve Bayes)

	AS	S	AD	F
ASSIS-TANT	838	16	39	0
SECRE-TARY	238	44	5	5
ADVI-SOR	72	0	150	0
FRIEND	9	18	0	20

Table 3: Confusion Matrix (Bayes Net)

Each row is the category, and each column is the number of e-mails misclassified. For instance, 782, the true positives, is the number of e-mails labeled by the SVM classifier to the category ASSISTANT that are correct predictions. Similarly, 71, 37, and 3, the false negatives, are the number of e-mails that have not been labeled by the classifier to the category ASSISTANT, but that should have.

For any category, the classifier *precision* is defined as the ratio of the number of relevant records retrieved to the total number of irrelevant and relevant records retrieved and the *recall* as the ratio of the number of relevant records retrieved to the total number of relevant records in the dataset. To combine these two measures in a single value, the *F-measure* is often used to reflect the relative importance of recall versus precision. When as much importance is granted to precision as it is to recall we have the F1-measure as:

$$F1 = \frac{precision + recall}{2 \cdot precision \cdot recall}$$

The F1-measure is an estimation of the breakeven point where precision and recall meets if classifier parameters are tuned to balance precision and recall.

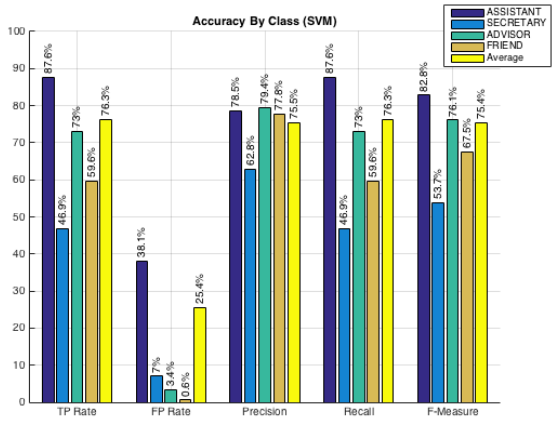

Figure 5: Accuracy By Category (SVM)

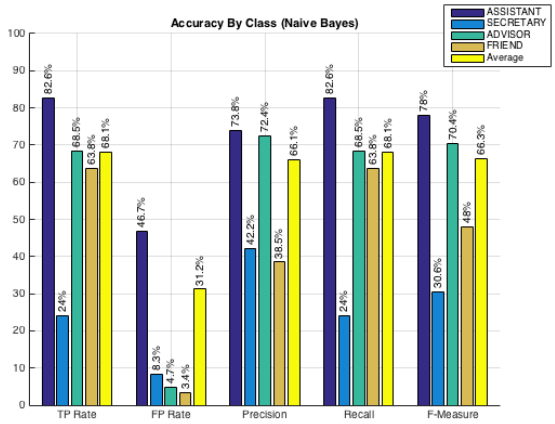

Figure 6: Accuracy By Category (Naïve Bayes)

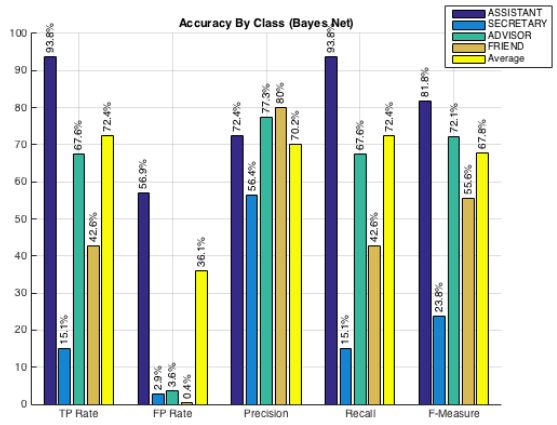

Figure 7: Accuracy By Category (Bayes Net)

Figure 5, Figure 6, and Figure 7 show the detailed accuracy by category for each classifier, including true *positive rate*, *false positive rate*, *precision*, *recall*, and *F-measure*. From these figures, we can see that all three classifiers perform the best for category *ASSISTANT*. One reason is that *ASSISTANT* has more data than the other three categories. All three classifiers perform the worst for category *SECRETARY* because a lot of e-mails in category *SECRETARY* are misclassified as *ASSISTANT*. Categories ADVISOR and FRIEND also have a large portion of e-mails misclassified as ASSISTANT, as shown in the confusion matrix. The reason is that all roles share similar job responsibilities. For example, assistants take care of meeting scheduling for Hillary, but all other roles can schedule meetings with Hillary.

In all, the results clearly show that with a high confidence level, we can achieve good results in learning and classifying relationship types based on e-mail content which is consistent with our initial theory. The confusion between *ASSISTANT* and *SECRETARY* is merely a side effect of their similar features and interchangeable roles. SVM produces the highest accuracy of 76% that is supported by studies which find SVM performs well on high dimensional feature sets [4].

5 Technical Challenges

While the results are promising, we identified a series of technical challenges over the course of this project. For one, it's apparent that the e-mail communications in the provided dataset are somewhat limited to internal operations, employees, and random meetings with peers. Table X shows that Hillary's communications were primarily between the same set of individuals all from within the same e-mail domain. The top 5 individuals made up over 60% of the available useful e-mail data.

Additionally, we found that Hillary isn't very responsive. The available dataset shows that Hillary receives a considerably amount of e-mails but responds to very select group of the same individuals with and overall 1:3 response rate. This greatly reduced the amount of data that could be used for modeling.

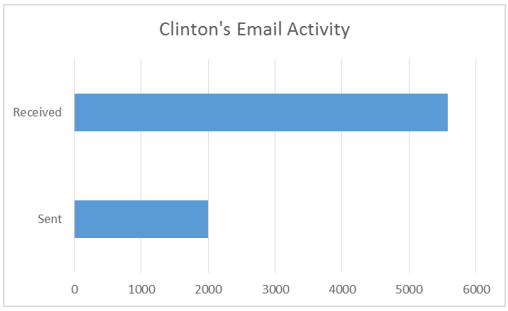

Figure 6: Clintons e-mail activity

Finally, we found that the data is inconsistent and required additional cleaning to ensure accuracy of the classification methods. For instance, some messages contain metadata such as date, time, and subject within the message body, while others simply contain the message. This inconsistency was impossible to completely eliminate thus introducing a non-deterministic set of noise thus reducing the overall accuracy.

6 Related Work

Drucker [4] presented a method for spam categorization with Support Vector Machines. They discussed design choices of feature representation, including term frequency and inverse document frequency. He also discussed performance criteria including recall and precision. They showed that SVM using binary features is a good candidate for spam categorization. Our project improved upon

Drucker's method of design choices of feature representation by expanding the concept to the multi-classification problem.

7 Conclusion

The goal of this project was to apply machine learning and text analysis concepts over Clinton's e-mails to extract meaningful and potentially impactful facts that model relationships between Clinton and her e-mail recipients. We proved that with high confidence and an accuracy of 76%, we can in fact model personal relationships through subtleties in conversations. Furthermore, this project demonstrates the effectiveness of SVM over Bayesian Networks and Naïve Bayes in text classification on high dimensional feature sets [1,3,4,5].

Given the size and focus group of the data set used in this project, only a limited range of trainable relationship classes were identified. Additionally Clinton responds at a ratio of 1:3, which limited the overall data. Non-the less, this project clearly demonstrates how text classification methods can be applied to model relationships through conversations with even a limited data set.

References

[1] Li, Lianghao, Xiaoming Jin, Sinno Jialin Pan, and Jian-Tao Sun. "Multi-domain Active Learning for Text Classification." *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '12*. Print.

[2] Yang, Jun, Yu-Gang Jiang, Alexander G. Hauptmann, and Chong-Wah Ngo. "Evaluating Bag-of-visual-words Representations in Scene Classification." *Proceedings of the International Workshop on Workshop on Multimedia Information Retrieval - MIR '07* (2007). Print.

[3] Awad, W.a. "Machine Learning Methods for Spam E-Mail Classification." *International Journal of Computer Science and Information Technology IJCSIT* (2011): 173-84. Print.

[4] Drucker, H., Donghui Wu, and V.n. Vapnik. "Support Vector Machines for Spam Categorization." *IEEE Trans. Neural Netw. IEEE Transactions on Neural Networks* (1999): 1048-054. Print.

[5] Burges, C. "A Tutorial on Support Vector Machines for Pattern Recondition" *Data Mining and Knowledge Discovery 2* (1998): 121-167. Print.

[6] Li, Lianghao, Jin, Xiaoming, and Sun, Jian-Tao. "Multi-Domain Active Learning for Text Classification."*SIGKDD international conference on Knowledge discovery and data mining* (2012): 1086-094. Print.

[7] Lucas, Peter J.f., Linda C. Van Der Gaag, and Ameen Abu-Hanna. "Bayesian Networks in Biomedicine and Health-care." *Artificial Intelligence in Medicine* (2004): 201-14. Print.