A Comparative Study of Human Coding and Context Analysis against Support Vector Machines (SVM) to Differentiate Campaign Emails by Party and Issues

Examining Narrowcasting in the 2004 Presidential Election

By Stephen Purpura (Harvard University), Dustin Hillard (University of Washington), and Dr. Philip Howard (University of Washington)

December 6, 2004 -- WORKING DRAFT
Last Updated: January 7, 2006

1 Introduction

This work investigates classification of emails sent by the political parties during the 2004 presidential election. Given an email without contextual information, we classify it as originating from either the Republican or Democratic Party. While this type of task is well practiced in the political communication literature, we use this exercise to demonstrate a new methodological technique to bridge the qualitative world of coding & context analysis with empirical analysis and methods.[1]

Our experiment involves two parallel studies using the same data set and coding rules. The first study is a traditional context analysis experiment conducted by Dr. Philip Howard at the University of Washington. The second is a computer-assisted context analysis conducted by Dustin Hillard and Stephen Purpura. The focus of this paper is to describe how a skilled computer scientist would approach the problem of categorizing thousands of email messages. Text categorization problems are frequently encountered by political communication analysts, and current methods employ manual techniques or computer software which searches for keywords in the text. While our proposed methods do not replace these techniques, we identify a new tool to arm the social science researcher with the fruits of advancing research in computational linguistic algorithms.

To bridge the gap between qualitative political communication analysis and research from computational linguistics, our work relies heavily on the computational linguistics literature. Context analysis in political communication has similar goals to topic spotting in newswire data. Numerous techniques for topic classification have been well documented. In this work, support vector machines (SVMs) are chosen due to their relatively strong performance on a wide variety of tasks. Support vector machines are a natural fit for topic classification because they deal well with sparse data and large dimensionality. Our only departure from most standard topic classification research is that campaign emails have different language patterns and characteristics from the typical news stories or broadcasts usually used in topic classification. Unlike news stories or broadcasts, campaign emails are repetitive and very similar. This can make the task of

[1] The authors wish to thank our reviewers: Dr. James Purpura (Columbia University), Dan Hopkins (Harvard University), and Dr. John Wilkerson (University of Washington).

topic spotting less complicated. However, we selected this example because it is easily replicated by future researchers and political science graduate students.

The remainder of this document should serve as a demonstration for building a prototypical machine learning system for categorizing political text. An overview of the problem to be solved is presented in Section 2. The procedures and results of Dr. Howard's traditional study are summarized in Section 3. An overview of the related work in topic classification is described in Section 4. The approach to classifier design is developed in Section 5. Experimental results are detailed in Section 6, and the main conclusions of this work are summarized in Section 7.

2 Searching for evidence of narrowcasting within political e-mail

Although we develop a new method for categorizing political text, the expressed purpose of this research is track how political campaigns use the Internet in their communications strategy by reporting on the email messages received from campaigns during the 2004 campaign period.

While there is a significant amount of research into how specific political campaigns collect information on voters and then use this intelligence to customize political messages, there is little research on how they do this using the Internet. In the 2000 campaign, Dr. Philip Howard studied the Presidential Candidate campaigns and observed some of these campaign strategies implemented using television and newspaper content. To address the lack of research related to Internet content, our research question examined how personal profile data collected from voters via the Internet by the Presidential campaigns in different states would be used to construct specific messages. We theorized that Presidential campaigns would choose to send different messages to voters based on their expressed preferences (as collected by the campaign web sites).

University of Washington students created multiple pseudonyms for purposively selected identities, signed up for political information from Presidential Candidates under these identities, and researchers analyzed the content of messages received during the campaign period. Groups of students were assigned to study the Presidential campaign messages from particular states, and the online identities they took were based on the important demographic categories in their assigned state. The campaigns being researched included the Democratic and Republican Presidential campaigns. The data included email content in the form of "personalized" messages sent from the websites.

3 Research Procedures and Results of the Traditional Context Analysis Study

Since the focus of this paper is to review new methods, we will briefly examine the procedures and results of the traditional political communication context analysis study. Within the traditional study, data was collected and archived by students and teaching assistants.

3.1 Research Procedures

The examination was conducted between September and December 2004, at the height of the election campaign period. Approximately 125 students in two UW Department of Communication classes (COM 300 & 417) were asked to sign up to receive email communications from the Democratic and Republican Presidential Candidates. Each student assumed 5 pseudonyms, each with different demographic features and living in one of the 50 states. The source of data for this study was the Presidential Campaign websites, and any email servers they use to send out content to American voters. The demographic features were selected by Dr. Howard with an eye to representing the social diversity within states.

Every Friday morning during lab section, the students checked their email accounts and reviewed the content of campaign email messages. The campaign websites were established after the summer conventions and vice-Presidential Candidates had been named. The students completed a short "Weekly Codesheet". The course teaching assistants collected all reports, removed the names of students and provided only the pseudonyms, and submitted the reports to Dr. Howard for final analysis.

3.3 Results

The results of the studies were published in two articles by Dr. Howard and his students on www.CampaignAudit.org, "Democrats shoot out e-mail faster, more frequently"[2] by Andrew Ralston and Peter Fotheringham and "Political e-mail: comparative table of a Smart-Mail Strategy" by Samantha Gatto and Jill Dalinkus[3]. The studies examined the frequency of email receipt from each of the parties and whether liberal or conservative identities were more likely to receive targeted messages. These results are summarized in Table 1.

Table 1: Average Message Count per identity, All States, Over Three Weeks from "Democrats shoot out e-mail faster, more frequently"[4] by Andrew Ralston and Peter Fotheringham

Personality	**New Messages**	**Week 1 October 1 st to 7 th**	**Week 2 October 8 th to 14 th**	**Week 3 October 15 th to 21 st**
Liberal	From Democrats	3.6	9.4	7.4
	From Republicans	1.0	1.0	1.1
Conservative	From Democrats	2.8	9.1	7.5
	From	1.0	1.0	1.0

[2] http://www.campaignaudit.org/2004/articles/democratsshootoutemail.html
[3] http://www.campaignaudit.org/2004/articles/email.html
[4] http://www.campaignaudit.org/2004/articles/democratsshootoutemail.html

Republicans

4 Topic classification using Support Vector Machines

Instead of relying on humans to correctly analyze the email messages and segment them, Hillard & Purpura used support vector machines, a software tool. The remainder of this section will serve as an introduction to support vector machines, including our use of word processing features.

4.1 Typical Objections to the use of Support Vector Machines

Many linguists and social scientists are skeptical that computer algorithms can analyze a corpus of text. Yet topic classification is a well researched task for which many different approaches have been shown to be effective. Even given its imperfections, let us consider how topic classification is useful for the purpose of categorization.

Dr. James Purpura, a tenured linguist at Teacher's College of Columbia University, frequently reminds me that it is not possible to evaluate the value of a word without the meaning of a word's placement within the linguistic environment.[5] Consider his examples using the word 'black':

> Aretha is black. (race)
> A black spot on her dress (cleanliness)
> A black hole (amount of light)
> A blackboard (a writing device usually green)
> Black and white (color)
> We're in the black (profit)

If a computer were to search for occurrences or frequency of occurrences of the word 'black' in text, for comparison to reference texts, there is no theoretical link between the use of the term and its contextual meaning. Therefore, the theoretical flaw with analyzing the number of occurrences is the question of whether the corpus (in this case the reference texts) is a true and unbiased representation of the language and the political stance of its authors. However, combining word frequency with collocations, two or three words grouped together, and syntactic structures is much more promising. Examples would include searching for word frequency counts of "a tall tale", "a tall building", "a high ceiling", and "a tall man" as distinct syntactic structures. This technique is much more promising because it examines groups of words together and it assumes that these groups of words are much more likely to have equivalent or similar meaning across multiple texts. In the linguistics field, this is sometimes referred to as e-rater automatic rating of writing ability.

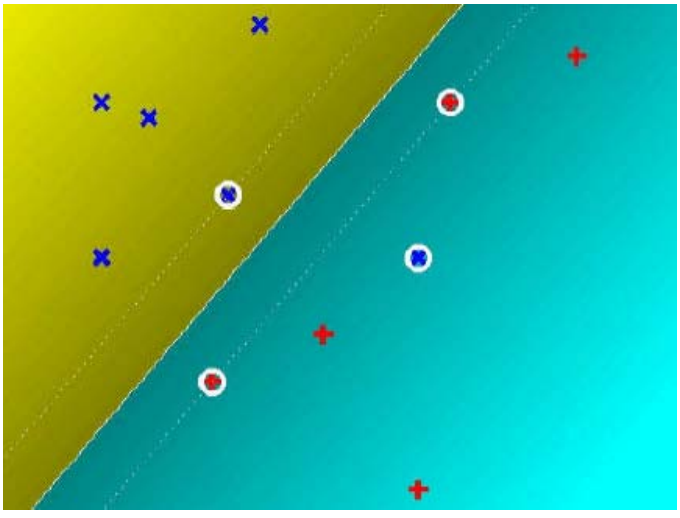
[5] Thanks to Dr. James Purpura for providing this example and support in the accompanying analysis.

Topic classification in computational linguistics has a theoretical basis in this collocation, except the collocation is limited to each training sample. Topic spotting research examines different approaches for classifying reference texts by comparing the groups of collocated words, but since the groups of collocated words aren't limited to just 2 or 3 word phrases the technique is not as theoretically convincing. Instead, collocation is measured as relative to a reference training sample. The effectiveness of different methods using this technique, such as regression models, nearest neighbor classification, Bayesian probabilistic approaches, decision trees, inductive rule learning, neural networks, on-line learning and support vector machines, are reviewed by (Yang and Liu, 1999) and (Sebastiani, 2002). Among these approaches SVMs consistently perform well for various tasks and domains. Due to the broad success of SVMs they are chosen for this work.

Other researchers have investigated similar techniques in political communication. (Laver, Benoit, and Garry, 2003). This "Wordscores" technique has, at its heart, a matrix multiply operation on word frequency. In the future, we will address comparisons of the Wordscores technique with SVMs more directly. But for the short term, the reader will need to settle for a reduction of the differences between their approach and our use of SVMs. The can be summarized by comparing the complexity of our processes. A linguist complains that computational linguistic algorithms do not fully consider whether words are a true and unbiased representation of the language and the political stance of authors. Likewise, a computational linguist will complain that a matrix multiply, as used in Wordscores (Laver, Benoit, and Garry, 2003), is too simple of a reduction. The simple matrix multiply may be effective for certain tasks, but a more robust algorithm may be more effective at working across domains and tackling more significant problems because it can examine many different features and cases.

4.2 Support Vector Machines

Support vector machines were introduced in (Vapnic, 1995) and the technique basically attempts to find the best possible surface to separate positive and negative training samples. The 'best possible' means the surface which produces the greatest possible margin among the boundary points, as in the figure below.



SVMs were developed for topic classification in (Joachims, 1998). Joachims motivates the use of SVMs using the characteristics of the topic classification problem: a high dimensional input space, few irrelevant features, sparse document representation, and that most text categorization problems are linearly separable. All of these factors are conducive to using SVMs because SVMs can train well under these conditions. That work performs feature selection with an information gain criterion and weights word features with a type of inverse document frequency. Various polynomial and RBF kernels are investigated, but most perform at a comparable level to (and sometimes worse than) the simple linear kernel. A software package for training and evaluating SVMs is available and described by (Joachims, 1999). That package is used for these experiments.

4.3 Word feature processing

Text input to topic classification systems is usually preprocessed and then word features given weights depending on importance measures. Most text classification work starts with word stemming to remove variable word endings and reduce words to a canonical form so that different word forms are all mapped to the same token (which is assumed to have essentially equal meaning for all forms). Word features usually consist of stemmed word counts, adjusted by some weighting. Inverse document frequency is commonly used, and has some justification (Papineni, 2001). More complex measures of word importance have shown to provide additional gains though. A weighted inverse document frequency is an extension of inverse document frequency to incorporate term frequency over texts, rather than just term presence (Tokunaga and Iwayama, 1994).
Term selection can also help improve results and many past approaches have found information gain to be good criteria (Yang and Liu, 1999) and (Sebastiani, 2002).

5 Classifier Design

Classifier design for this project implements proven methods for SVM classification. The classifier will be described in the sequence that training and testing data are processed in experiments.

5.1 Word feature processing

The first step in preparing the text for feature extraction is to remove all non-word tokens and map everything to lower case, so that the remaining text consists of only lower case words with no punctuation or extraneous tokens that may exists in the raw text. The following step performs the standard Porter Stemming Algorithm (Porter, 1980).
Although typical word feature weightings such as inverse document frequency have been generally effective, as mentioned previously, more detailed forms of word weighting have also provided further gains. This work adopts a weighting related to mutual information, where each word is given a feature value w_i as shown in Equation 4.

$$w_i = log(\frac{p(w,t)}{p(w)p(t)}) = log(\frac{p(w|t)p(t)}{p(w)p(t)}) \quad (4)$$

In this equation, the top term p(w|t) is the probability of a word in a particular email (the number of occurrences in this email, divided by the number of total words in the email). The denominator term p(w) is the probability of a word across all emails (the number of

occurrences of this word in all emails, divided by the total number of words in all emails). This also reduces to an intuitive form as in Equation 5 where it can be thought of as a ratio of word frequency given an email, divided by the overall frequency in all available emails.

$$w_i = log(\frac{p(w|t)}{p(w)}) \qquad (5)$$

Finally, only words with $w_i > 0$ are placed in the term by conversation matrix (this is all terms with a ratio greater than 1, or in other words those that occur more frequently than the corpus average).

5.2 Support Vector Machine Implementation

A support vector machine classifier is trained with the features vectors of w_i as described above. Linear SVMs are utilized because past work has shown that text categorization is almost always linearly separable, and more complex kernels such as polynomials or RBFs have shown little gains for this task. In addition, more complex kernels introduce additional variability due to the required tuning of additional kernel parameters. The testing condition evaluates a test feature vector against the trained SVM. The software package SVM_{light} is used to train and evaluate all support vector machines.

6 Research Procedures and Results of the SVM Analysis Study

Once the data from the human coding and context analysis techniques had been collected, it was made anonymous and passed on to Purpura and Hillard. Purpura & Hillard then transferred the data into a database and prepared it for processing.

When the traditional context analysis process was conducted by Dr. Howard's students, the students manually counted the emails into piles using rules. Likewise, Purpura & Hillard used these same rules to identify samples to train the SVM system. However, unlike the traditional context analysis process, Hillard and Purpura needed to identify a few samples of emails from each of the groups: Conservative/Liberal and Republican/Democrat.

To achieve this, Purpura & Hillard randomly sampled, with replacement, 10 the emails. When an email was displayed as part of the random sample, they identified it as Conservative/Liberal and Republican/Democrat. The intent was to use these samples were used to "train" the SVM system to divide all the email into piles.

6.1 Conservative/Liberal Differentiation

Selecting samples for the Conservative/Liberal profiles proved to be difficult, because so little variation occurred in the actual emails. In fact, the human coding and context analysis team had suggested that no discernable difference existed between the emails sent to the different profiles. After a random sample of 10 emails, no differences had been spotted.

However, Purpura & Hillard used the "grep" utility to conduct keyword searches on the email subject lines for keywords related to each of the profiles. The keywords were simply "liberal" and "conservative". This method quickly turned up significant examples, demonstrating an observed error rate for human context analysis of 100% misclassification on this article feature. Purpura & Hillard used this non-random sample to select samples to train the SVM system to differentiate emails between liberal and conservative.

The results of using the SVM system to classify the emails as Liberal/Conservative proved to be significantly more impressive than the human analyzers, but they also demonstrated the weakness of using SVMs against small training sets which lack discernable distinguishing features. After extensive analysis of thousands of emails, Table 2 shows that 2000 training samples were required to achieve 99.8% accuracy in email classification. The required processing time was 11 seconds on a Pentium III, ~ 900 MHz computer.

Table 2: Conservative versus Liberal profiles
The number of training samples required to reproduce the results from the original human coding and contextual analysis study.

Training Samples	Accuracy
10	52%
20	63%
50	72%
100	92%
250	98%
500	98.7%
1000	99.2%
1500	99.2%
2000	99.8%

6.1 Republican/Democrat Differentiation

Selecting samples for the Republican/Democrat sources was very easy. The information was contained within the header context of the email.[6] The human coding and context analysis team had easily differentiated between the two and no errors were found in their processing. Purpura & Hillard again used the "grep" utility to conduct keyword searches on the email subject lines for keywords related to each of the profiles to establish a baseline count. Once the baseline counts were generated, the SVM system was given a 10 email training sample and asked to classify all of the emails. The first version of this test kept the emails completely intact, but the results were discarded because we considered allowing the proper names and sender information in the emails to be too easy

[6] Note that all distinguishing information, such as sender address and proper names were stripped from the emails prior to processing by the system.

of a test for the SVM system. So the test was made more meaningful by completely stripping the identity information from the sender and all proper names from the emails. This information was used successfully by the human coders, but we wanted to see if the SVM system could differentiate without the advantage of this context.

The results of using the SVM system to classify the emails as Republican/Democrat proved to be significantly more impressive than we expected. The SVMs achieved a 92% accuracy rate on only 10 training samples. After extensive analysis of thousands of emails, Table 3 shows that only 50 training samples were required to achieve 98% accuracy in email classification. The required processing time was 11 seconds on a Pentium III, ~ 900 MHz computer.

Table 3: Democratic versus Republican sources
The number of training samples required to reproduce the results from the original human coding and contextual analysis study.

Training Samples	Accuracy
10	92%
20	95%
50	98%
100	98%
250	100%

7 Conclusions

When you consider the number of projects using human coding and analysis techniques or computer assisted keyword search analysis techniques, the results of this study are stunning. The Support Vector Machines proved to be significantly more efficient, more reliable, and easier to manage. To put the results in context, hundreds of students spent weeks analyzing email messages to achieve the results in the human study. A single researcher spent a few hours to achieve the results in the SVM study. The estimated reduction in labor cost is more than 800 person hours for the rudimentary questions answered by this study alone. Further, the SVM study proved more sensitive than the human study. The example in topic spotting trends in the liberal/conservative profiles reinforced the advantage of computer assisted techniques.

However, the results also demonstrate a limitation of the technique. Out-of-band information, in the form of the keywords “liberal” and “conservative”, were required to help the Support Vector Machine model make a differentiation when the rules for classification were unclear or ineffective. Dealing with these limitations – recognizing when a SVM system is working well at categorization or not working at all – will be the subject of several future papers. But in the short term, we have shown that the effectiveness of the SVM system can be easily estimated by the ability of a researcher to classify the differences in a small random sample. If the researcher is unable to easily

classify the random samples, then SVMs will not be effective without help or modification of the classification scheme.

Further, this study applied a general purpose solution from topic spotting without any attempt to optimize the algorithms. Future research should examine alternative algorithms for efficiency and effectiveness. In addition, the SVM software needs to be packaged with improvements in usability so that political scientists can easily consume these powerful tools.

8 References

N. Cristianini, J. Shawe-Taylor, and H. Lodhi. 2001. Latent semantic kernels. In C. Brodley and A. Danyluk, editors, *Proceedings of ICML-01, 18th International Conference on Machine Learning*, pages 66–73. Morgan Kaufmann Publishers, San Francisco, US.

S. Deerwester et al. 1990. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407.

T. Joachims. 1998. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of theEuropean Conference on Machine Learning*.

T. Joachims. 1999. Advances in kernel methods - support vector learning. *Making large-Scale SVM Learning Practical*.

Michael Laver, Kenneth Benoit, and John Garry (2003). Extracting policy positions from political texts using words as data. *American Political Science Review* **97**(2).

K. Papineni. 2001. Why inverse document frequency? In *NAACL Proceedings*, pages 25-32.

M. F. Porter. 1980. An algorithm for suffix stripping. *Program*, 16(3):130–137.

F. Sebastiani. 2002. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1).

T. Tokunaga and M. Iwayama, 1994. *Text categorization based on weighted inverse document frequency*. Technical Report 94 TR0001, Department of Computer Science, Tokyo Institute of Technology.

V. Vapnic. 1995. *The Nature of Statistical Learning Theory*. Springer, New York, NY.
Y. Yang and X. Liu. 1999. A re-examination of text categorization methods. In *Proceedings of SIGIR-99*, November.