Extracting Social Power Relationships from Natural Language

Philip Bramsen
Louisville, KY
bramsen@alum.mit.edu*

Martha Escobar-Molano
San Diego, CA
mescobar@asgard.com*

Ami Patel
Massachusetts Institute of Technology
Cambridge, MA
ampatel@mit.edu*

Rafael Alonso
SET Corporation
Arlington, VA
ralonso@setcorp.com

Abstract

Sociolinguists have long argued that social context influences language use in all manner of ways, resulting in *lects*[1]. This paper explores a text classification problem we will call *lect modeling*, an example of what has been termed computational sociolinguistics. In particular, we use machine learning techniques to identify *social power relationships* between members of a social network, based purely on the content of their interpersonal communication. We rely on statistical methods, as opposed to language-specific engineering, to extract features which represent vocabulary and grammar usage indicative of social power lect. We then apply support vector machines to model the social power lects representing superior-subordinate communication in the Enron email corpus. Our results validate the treatment of lect modeling as a text classification problem – albeit a hard one – and constitute a case for future research in computational sociolinguistics.

1 Introduction

Linguists in sociolinguistics, pragmatics and related fields have analyzed the influence of social context on language and have catalogued countless phenomena that are influenced by it, confirming many with qualitative and quantitative studies. Indeed, social context and function influence language at every level – morphologically, lexically, syntactically, and semantically, through discourse structure, and through higher-level abstractions such as pragmatics.

Considered together, the extent to which speakers modify their language for a social context amounts to an identifiable variation on language, which we call a *lect*. *Lect* is a backformation from words such as *dialect* (geographically defined language) and e*thnolect* (language defined by ethnic context).

In this paper, we describe *lect classifiers* for *social power relationships*. We refer to these lects as:

- *UpSpeak*: Communication directed to someone with greater social authority.
- *DownSpeak*: Communication directed to someone with less social authority.
- *PeerSpeak*: Communication to someone of equal social authority.

We call the problem of modeling these lects *Social Power Modeling* (SPM). The experiments reported in this paper focused primarily on modeling UpSpeak and DownSpeak.

Manually constructing tools that effectively model specific linguistic phenomena suggested by sociolinguistics would be a Herculean effort. Moreover, it would be necessary to repeat the effort in every language! Our approach first identifies statistically salient phrases of words and parts of speech – known as *n-grams* – in training texts generated in conditions where the social power

* This work was done while these authors were at SET Corporation, an SAIC Company.

[1] Fields that deal with society and language have inconsistent terminology; “lect” is chosen here because “lect” has no other English definitions and the etymology of the word gives it the sense we consider most relevant.

Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, pages 773–782,
Portland, Oregon, June 19-24, 2011.

relationship is known. Then, we apply machine learning to train classifiers with groups of these n-grams as features. The classifiers assign the UpSpeak and DownSpeak labels to unseen text. This methodology is a cost-effective approach to modeling social information and requires no language- or culture-specific feature engineering, although we believe sociolinguistics-inspired features hold promise.

When applied to the corpus of emails sent and received by Enron employees (CALO Project 2009), this approach produced solid results, despite a limited number of training and test instances.

This has many implications. Since manually determining the power structure of social networks is a time-consuming process, even for an expert, effective SPM could support data driven socio-cultural research and greatly aid analysts doing national intelligence work. Social network analysis (SNA) presupposes a collection of individuals, whereas a social power lect classifier, once trained, would provide useful information about individual author-recipient links. On networks where SNA already has traction, SPM could provide complementary information based on the content of communications.

If SPM were yoked with sentiment analysis, we might identify which opinions belong to respected members of online communities or lay the groundwork for understanding how respect is earned in social networks.

More broadly, computational sociolinguistics is a nascent field with significant potential to aid in modeling and understanding human relationships. The results in this paper suggest that successes to date modeling authorship, sentiment, emotion, and personality extend to social power modeling, and our approach may well be applicable to other dimensions of social meaning.

In the coming sections, we first establish the **Related Work**, primarily from Statistical NLP. We then cover our **Approach,** the **Evaluation**, and, finally, the **Conclusions and Future Research**.

2 Related Work

The feasibility of Social Power Modeling is supported by sociolinguistic research identifying specific ways in which a person's language reflects his relative power over others. Fairclough's classic work *Language and Power* explores how "sociolinguistic conventions . . . arise out of -- and give rise to – particular relations of power" (Fairclough, 1989). Brown and Levinson created a theory of politeness, articulating a set of strategies which people employ to demonstrate different levels of politeness (Brown & Levinson, 1987). Morand drew upon this theory in his analysis of emails sent within a corporate hierarchy; in it, he quantitatively showed that emails from subordinates to superiors are, in fact, perceived as more polite, and that this perceived politeness is correlated with specific linguistic tactics, including ones set out by Brown and Levinson (Morand, 2000). Similarly, Erikson et al identified measurable characteristics of the speech of witnesses in a courtroom setting which were directly associated with the witness's level of social power (Erikson, 1978). Given, then, that there are distinct differences among what we term UpSpeak and DownSpeak, we treat Social Power Modeling as an instance of *text classification* (or *categorization*): we seek to assign a class (UpSpeak or DownSpeak) to a text sample. Closely related natural language processing problems are authorship attribution, sentiment analysis, emotion detection, and personality classification: all aim to extract higher-level information from language.

Authorship attribution in computational linguistics is the task of identifying the author of a text. The earliest modern authorship attribution work was (Mosteller & Wallace, 1964), although forensic authorship analysis has been around much longer. Mosteller and Wallace used statistical language-modeling techniques to measure the similarity of disputed Federalist Papers to samples of known authorship. Since then, authorship identification has become a mature area productively exploring a broad spectrum of features (stylistic, lexical, syntactic, and semantic) and many generative and discriminative modeling approaches (Stamatatos, 2009). The generative models of authorship identification motivated our statistically extracted lexical and grammatical features, and future work should consider these language modeling (a.k.a. compression) approaches.

Sentiment analysis, which strives to determine the attitude of an author from text, has recently garnered much attention (e.g. Pang, Lee, & Vaithyanathan, 2002; Kim & Hovy, 2004; Breck, Choi

& Cardie, 2007). For example, one problem is classifying user reviews as positive, negative or neutral. Typically, polarity lexicons (each term is labeled as positive, negative or neutral) help determine attitudes in text (Hiroya & Takamura, 2005, Ravichandran 2009, Choi & Cardie 2009).

The polarity of an expression can be determined based on the polarity of its component lexical items (Choi & Cardie 2008). For example, the polarity of the expression is determined by the majority polarity of its lexical items or by rules applied to syntactic patterns of expressions on how to determine the polarity from its lexical components. McDonald et al studied models that classify sentiment on multiple levels of granularity: sentence and document-level (McDonald, 2007). Their work jointly classifies sentiment at both levels instead of using independent classifiers for each level or cascaded classifiers. Similar to our techniques, these studies determine the polarity of text based on its component lexical and grammatical sequences. Unlike their works, our text classification techniques take into account the frequency of occurrence of word n-grams and part-of-speech (POS) tag sequences, and other measures of statistical salience in training data.

Text-based emotion prediction is another instance of text classification, where the goal is to detect the emotion appropriate to a text (Alm, Roth & Sproat, 2005) or provoked by an author, for example (Strapparava & Mihalcea, 2008). Alm, Roth, and Sproat explored a broad array of lexical and syntactic features, reminiscent of those of authorship attribution, as well as features related to story structure. A Winnow-based learning algorithm trained on these features convincingly predicted an appropriate emotion for individual sentences of narrative text. Strapparava and Mihalcea try to predict the emotion the author of a headline intends to provoke by leveraging words with known affective sense and by expanding those words' synonyms. They used a Naïve Bayes classifier trained on short blogposts of known emotive sense. The knowledge engineering approaches were generally superior to the Naïve Bayes approach. Our approach is corpus-driven like the Naïve Bayes approach, but we interject statistically driven feature selection between the corpus and the machine learning classifiers.

In personality classification, a person's language is used to classify him on different personality dimensions, such as extraversion or neuroticism (Oberlander & Nowson, 2006; Mairesse & Walker; 2006). The goal is to recover the more permanent traits of a person, rather than fleeting characteristics such as sentiment or emotion. Oberlander and Nowson explore using a Naïve Bayes and an SVM classifier to perform binary classification of text on each personality dimension. For example, one classifier might determine if a person displays a high or low level of extraversion. Their attempt to classify each personality trait as either "high" or "low" echoes early sentiment analysis work that reduced sentiments to either positive or negative (Pang, Lee, & Vaithyanathan, 2002), and supports initially treating Social Power Modeling as a binary classification task. Personality classification seems to be the application of text classification which is the most relevant to Social Power Modeling. As Mairesse and Walker note, certain personality traits are indicative of leaders. Thus, the ability to model personality suggests an ability to model social power lects as well.

Apart from text classification, work from the topic modeling community is also closely related to Social Power Modeling. Andrew McCallum extended Latent Dirichlet Allocation to model the author and recipient dependencies of per-message topic distributions with an Author-Recipient-Topic (ART) model (McCallum, Wang, & Corrada-Emmanuel, 2007). This was the first significant work to model the content and relationships of communication in a social network. McCallum et al applied ART to the Enron email corpus to show that the resulting topics are strongly tied to role. They suggest that clustering these topic distributions would yield roles and argue that the person-to-person similarity matrix yielded by this approach has advantages over those of canonical social network analysis. The same authors proposed several Role-Author-Recipient-Topic (RART) models to model authors, roles and words simultaneously. With a RART modeling roles-per-word, they produced per-author distributions of generated roles that appeared reasonable (e.g. they labeled Role 10 as 'grant issues' and Role 2 as 'natural language researcher').

We have a similar emphasis on statistically modeling language and interpersonal communication. However, we model social power relationships, not roles or topics, and our approach produces discriminative classifiers, not generative models, which enables more concrete evaluation.

Namata, Getoor, and Diehl effectively applied role modeling to the Enron email corpus, allowing them to infer the social hierarchy structure of Enron (Namata et al., 2006). They applied machine learning classifiers to map individuals to their roles in the hierarchy based on features related to email traffic patterns. They also attempt to identify cases of manager-subordinate relationships within the email domain by ranking emails using traffic-based and content-based features (Diehl et al., 2007). While their task is similar to ours, our goal is to classify any case in which one person has more social power than the other, not just identify instances of direct reporting.

3 Approach

3.1 Feature Set-Up

Previous work in traditional text classification and its variants – such as sentiment analysis – has achieved successful results by using the bag-of-words representation; that is, by treating text as a collection of words with no interdependencies, training a classifier on a large feature set of word unigrams which appear in the corpus. However, our hypothesis was that this approach would not be the best for SPM. Morand's study, for instance, identified specific features that correlate with the direction of communication within a social hierarchy (Morand, 2000). Few of these tactics would be effectively encapsulated by word unigrams. Many would be better modeled by POS tag unigrams (with no word information) or by longer n-grams consisting of either words, POS tags, or a combination of the two. "Uses subjunctive" and "Uses past tense" are examples. Because considering such features would increase the size of the feature space, we suspected that including these features would also benefit from algorithmic means of selecting n-grams that are indicative of particular lects, and even from *binning* these relevant n-grams into sets to be used as features.

Therefore, we focused on an approach where each feature is associated with a set of one or more n-grams. Each n-gram is a sequence of words, POS tags or a combination of words and POS tags ("mixed" n-grams). Let *S* represent a set $\{n_1, ..., n_k\}$ of n-grams. The feature associated with S on text T would be:

$$f(S,T) = \sum_{i=1}^{k} freq(n_i, T)$$

where $freq(n_i, T)$ is the relative frequency (defined later) of n_i in text *T*. Let n_i represent the sequence $s_1 \ldots s_m$ where s_j specifies either a word or a POS tag. Let *T* represent the text consisting of the sequence of tagged-word tokens $t_1 \ldots t_l$. $freq(n_i, T)$ is then defined as follows:

$$freq(n_i, T) = freq(s_1 \ldots s_m, T)$$
$$= \frac{\left|\left\{t_{b+1} \ldots t_{b+m} : \forall_{1 \le p \le m}\left(t_{b+p} = s_p\right)\right\}\right|}{l - m + 1}$$

where:

$$t_i = s_j \leftrightarrow \begin{cases} word(t_i) = s_j \; if \; s_j \; is \, a \, word \\ tag(t_i) = s_j \; if \; s_j \; is \, a \, tag \end{cases}$$

To illustrate, consider the following feature set, a bigram and a trigram (each term in the n-gram either has the form *word* or *^tag*):

> *{please ^VB, please ^'comma' ^VB}*[2]

The tag "VB" denotes a verb. Suppose T consists of the following tokenized and tagged text (sentence initial and final tokens are not shown):

> *please^RB bring^VB the^DET report^NN to^TO our^PRP$ next^JJ weekly^JJ meeting^NN .^.*

The first n-gram of the set, *please ^VB*, would match *please^RB bring^VB* from the text. The frequency of this n-gram in *T* would then be 1/9, where 1 is the number of substrings in *T* that match

[2] To distinguish a comma separating elements of a set with a comma as part of an ngram, we use 'comma' to denote the punctuation mark ',' as part of the ngram.

please ^VB and 9 is the number of bigrams in *T*, excluding sentence initial and final markers. The other n-gram, the trigram *please ^'comma' ^VB,* does not have any match, so the final value of the feature is 1/9.

Defining features in this manner allows us to both explore the bag-of-words representation as well as use groups of n-grams as features, which we believed would be a better fit for this problem.

3.2 N-Gram Selection

To identify n-grams which would be useful features, frequencies of n-grams in only the training set are considered. Different types of frequency measures were explored to capture different types of information about an n-gram's usage. These are:

- *Absolute frequency*: The total number of times a particular n-gram occurs in the text of a given class (social power lect).
- *Relative frequency*: The total number of times a particular n-gram occurs in a given class, divided by the total number of n-grams in that class. Normalization by the size of the class makes relative frequency a better metric for comparing n-gram usage across classes.

We then used the following frequency-based metrics to select n-grams:

- We set a minimum threshold for the absolute frequency of the n-gram in a class. This helps weed out extremely infrequent words and spelling errors.
- We require that the ratio of the relative frequency of the n-gram in one class to its relative frequency in the other class is also greater than a threshold. This is a simple means of selecting n-grams indicative of lect.

In experiments based on the bag-of-words model, we only consider an absolute frequency threshold, whereas in later experiments, we also take into account the relative frequency ratio threshold.

3.3 N-gram Binning

In experiments in which we bin n-grams, selected n-grams are assigned to the class in which their relative frequency is highest. For example, an n-gram whose relative frequency in UpSpeak text is twice that in DownSpeak text would be assigned to the class UpSpeak.

N-grams assigned to a class are then partitioned into sets of n-grams. Each of these sets of n-grams is associated with a feature. This partition is based on the n-gram type, the length of n-grams and the relative frequency ratio of the n-grams. While the n-grams composing a set may themselves be indicative of social power lects, this method of grouping them makes no guarantees as to how indicative the overall set is. Therefore, we experimented with filtering out sets which had a negligible information gain. Information gain is an information theoretic concept measuring how much the probability distributions for a feature differ among the different classes. A small information gain suggests that a feature may not be effective at discriminating between classes.

Although this approach to partitioning is simple and worthy of improvement, it effectively reduced the dimensionality of the feature space.

3.4 Classification

Once features are selected, a classifier is trained on these features. Many features are weak on their own; they either occur rarely or occur frequently but only hint weakly at social information. Therefore, we experimented with classifiers friendly to weak features, such as Adaboost and Logistic Regression (MaxEnt). However, we generally achieved the best results using support vector machines, a machine learning classifier which has been successfully applied to many previous text classification problems. We used Weka's optimized SVMs (SMO) (Witten 2005, Platt 1998) and default parameters, except where noted.

4 Evaluation

4.1 Data

To validate our supervised learning approach, we sought an adequately large English corpus of person-to-person communication labeled with the ground truth. For this, we used the publicly available Enron corpus. After filtering for duplicates and removing empty or otherwise unusable emails, the total number of emails is 245K, containing roughly 90 million words. However, this total includes emails to non-Enron employees, such as family members and employees of other corporations, emails to multiple people, and emails received from Enron employees without a known corporate role. Because the author-recipient relationships of these emails could not be established, they were not included in our experiments.

Building upon previous annotation done on the corpus, we were able to ascertain the corporate role (CEO, Manager, Employee, etc.) of many email authors and recipients. From this information, we determined the author-recipient relationship by applying general rules about the structure of a corporate hierarchy (an email from an Employee to a CEO, for instance, is UpSpeak). This annotation method does not take into account promotions over time, secretaries speaking on behalf of their supervisors, or other causes of relationship irregularities. However, this misinformation would, if anything, generally hurt our classifiers.

The emails were pre-processed to eliminate text not written by the author, such as forwarded text and email headers. As our approach requires text to be POS-tagged, we employed Stanford's POS tagger (http://nlp.stanford.edu/software/tagger.shtml). In addition, text was regularized by conversion to lower case and tokenized to improve counts.

To create training and test sets, we partitioned the authors of text from the corpus into two sets: A and B. Then, we used text authored by individuals in A as a training set and text authored by individuals in B as a test set. The training set is used to determine discriminating features upon which classifiers are built and applied to the test set. We found that partitioning by authors was necessary to avoid artificially inflated scores, because the classifiers pick up aspects of particular authors' language (idiolect) in addition to social power lect information. It was not necessary to account for recipients because the emails did not contain text from the recipients. Table 1 summarizes the text partitions.

	UpSpeak		**DownSpeak**	
	Links	Words	Links	Words
Training	431	136K	328	63K
Test	232	74K	148	27K

Table 1. Author-based Training and Test partitions. The number of author-recipient pairs (links) and the number of words in text labeled as UpSpeak and DownSpeak are shown.

Because preliminary experiments suggested that smaller text samples were harder to classify, the classifiers we describe in this paper were both trained and tested on a subset of the Enron corpus where at least 500 words of text was communicated from a specific author to a specific recipient. This subset contained 142 links, 40% of which were used as the test set.

Weighting for Cost-Sensitive Learning: The original corpus was not balanced: the number of UpSpeak links was greater than the number of DownSpeak links. Varying the weight given to training instances is a technique for creating a classifier that is cost-sensitive, since a classifier built on an unbalanced training set can be biased towards avoiding errors on the overrepresented class (Witten, 2005). We wanted misclassifying UpSpeak as DownSpeak to have the same cost as misclassifying DownSpeak as UpSpeak. To do this, we assigned weights to each instance in the training set. UpSpeak instances were weighted less than DownSpeak instances, creating a training set that was balanced between UpSpeak and DownSpeak. Balancing the training set generally improved results.

Weighting the test set in the same manner allowed us to evaluate the performance of the classifier in a situation in which the numbers of UpSpeak and DownSpeak instances were equal. A baseline classifier that always predicted the majority class would, on its own, achieve an accuracy of 74% on UpSpeak/DownSpeak classification of unweighted test set instances with a minimum length of 500 words. However, results on the weighted test set are properly compared to a baseline of 50%. We include both approaches to scoring in this paper.

4.2 UpSpeak/DownSpeak Classifiers

In this section, we describe experiments on classification of interpersonal email communication into UpSpeak and DownSpeak. For these experiments, only emails exchanged between two people related by a superior/subordinate power relationship were

	Features	# of features	# of n-grams	Cross-Validation		Test Set (weighted)		Test Set (unweighted)	
				Acc (%)	F-score	Acc (%)	F-score	Acc (%)	F-score
(1)	Word unigrams	3899	3899	55.4	.481	62.1	.567	78.9	.748
(2)	Word bigrams	3740	3740	54.5	.457	56.4	.498	73.7	.693
(3)	Word unigrams + word bigrams	7639	7639	51.8	.398	63.3	.576	80.7	.762
(4)	(3) + tag unigrams + tag bigrams	9014	9014	51.8	.398	58.8	.515	77.2	.719
(5)	Binned n-grams	8	106	83.0	.830	**78.1**	**.781**	77.2	.783
(6)	N-grams from (5), separated	106	106	83.0	.828	60.5	.587	70.2	.698
(7)	(5) + polite imperatives	9	108	83.9	.839	77.1	.771	**78.9**	**.797**

Table 2. Experiment Results. Accuracies/F-Scores with an SVM classifier for 10-fold cross validation on the weighted training set and evaluation against the weighted and unweighted test sets. Note that the baseline accuracy against the unweighted test set is 74%, but 50% for the weighted test set and cross-validation.

Human-Engineered Features: Before examining the data itself, we identified some features which we thought would be predictive of UpSpeak or DownSpeak, and which could be fairly accurately modeled by mixed n-grams. These features included the use of different types of imperatives.

We also thought that the type of greeting or signature used in the email might be reflective of formality, and therefore of UpSpeak and DownSpeak. For example, subordinates might be more likely to use an honorific when addressing a superior, or to sign an email with “Thanks.” We preformed some preliminary experiments using these features. While the feature set was too small to produce notable results, we identified which features actually were indicative of lect. One such feature was polite imperatives (imperatives preceded by the word “please”). The polite imperative feature was represented by the n-gram set:

{please ^VB, please ^‘comma’ ^VB}.

Unigrams and Bigrams: As a different sort of baseline, we considered the results of a bag-of-words based classifier. Features used in these experiments consist of single words which occurred a minimum of four times in the relevant lects (UpSpeak and DownSpeak) of the training set. The results of the SVM classifier, shown in line (1) of Table 2, were fairly poor. We then performed experiments with word bigrams, selecting as features those which occurred at least seven times in the relevant lects of the training set. This threshold for bigram frequency minimized the difference in the number of features between the unigram and bigram experiments. While the bigrams on their own were less successful than the unigrams, as seen in line (2), adding them to the unigram features improved accuracy against the test set, shown in line (3).

As we had speculated that including surface-level grammar information in the form of tag n-grams would be beneficial to our problem, we performed experiments using all tag unigrams and all tag bigrams occurring in the training set as features. The results are shown in line (4) of Table 2. The results of these experiments were not particularly strong, likely owing to the increased sparsity of the feature vectors.

Binning: Next, we wished to explore longer n-grams of words or POS tags and to reduce the sparsity of the feature vectors. We therefore experimented with our method of binning the individual n-grams to be used as features. We binned features by their relative frequency ratios. In addition to binning, we also reduced the total number of n-grams by setting higher frequency thresholds and relative frequency ratio thresholds.

When selecting n-grams for this experiment, we considered only word n-grams and tag n-grams – not mixed n-grams, which are a combination of words and tags. These mixed n-grams, while useful for specifying human-defined features, largely increased the dimensionality of the feature search space and did not provide significant benefit in preliminary experiments. For the word sequences, we set an absolute frequency threshold that depended on class. The frequency of a word n-gram in a particular class was required to be *0.18 * nrlinks / n*, where *nrlinks* is the number of links in each class (431 for UpSpeak and 328 for DownSpeak), and *n* is the number of words in the class. The relative frequency ratio was required to be at least 1.5. The tag sequences were required to meet an absolute frequency threshold of 20, but the same relative frequency ratio of 1.5.

Binning the n-grams into features was done based on both the length of the n-gram and the relative frequency ratio. For example, one feature might represent the set of all word unigrams which have a relative frequency ratio between 1.5 and 1.6.

We explored possible feature sets with cross validation. Before filtering for low information gain, we used six word n-gram bins per class (relative frequency ratios of 1.5, 1.6 ..., 1.9 and 2.0+), one tag n-gram bin for UpSpeak (2.0+), and three tag n-gram bins for DownSpeak (2.0+, 5.0+, 10.0+). Even with the weighted training set, DownSpeak instances were generally harder to identify and likely benefited from additional representation. Grouping features by length was a simple but arbitrary method for reducing dimensionality, yet sometimes produced small bins of otherwise good features. Therefore, as we explored the feature space, small bins of different n-gram lengths were merged. We then employed Weka's InfoGain feature selection tool to remove those features with a low information gain[3], which removed all but eight features. The results of this experiment are shown in line (5) of Table 2. It far outperforms the bag-of-words baselines, despite significantly fewer features.

To ascertain which feature reduction method had the greatest effect on performance – binning or setting a relative frequency ratio threshold – we performed an experiment in which all the n-grams that we used in the previous experiment were their own features. Line (6) of Table 2 shows that while this approach is an improvement over the basic bag-of-words method, grouping features still improves results.

Our goal was to have successful results using only statistically extracted features; however, we examined the effect of augmenting this feature set with the most indicative of the human-identified feature – polite imperatives. The results, in line (7), show a slight improvement in both the cross validation accuracy, and the accuracy against the unweighted test set increases to **78.9%**[4]. However, among the weighted test sets, the highest accuracy was **78.1%**, with the features in line (5).

We report the scores for cross-validation on the training set for these features; however, because the features were selected with knowledge of their per-class distribution in the training set, these cross-validation scores should not be seen as the classifier's true accuracy.

Self-Training: Besides sparse feature vectors, another factor likely to be hurting our classifier was the limited amount of training data. We attempted to increase the training set size by performing exploratory experiments with self-training, an iterative semi-supervised learning method (Zhu, 2005) with the feature set from (7). On the first iteration, we trained the classifier on the labeled training set, classified the instances of the unlabeled test set, and then added the instances of the test set along with their predicted class to the training set to be used for the next iteration. After three iterations, the accuracy of the classifier when evaluated on the weighted test set improved to **82%**, suggesting that our classifiers would benefit from more data.

Impact of Cost-Sensitive Learning: Without cost-sensitive learning, the classifiers were heavily biased towards UpSpeak, tending to classify both DownSpeak and UpSpeak test instances as UpSpeak. With cost-sensitive training, overall performance improved and classifier performance on DownSpeak instances improved dramatically. In (5) of Table 2, DownSpeak classifier accuracy even edged out the accuracy for UpSpeak. We expect that on a larger dataset behavior with unweighted training and test data would improve.

5 Conclusions and Future Research

We presented a corpus-based statistical learning approach to modeling social power relationships and experimental results for our methods. To our

[3] In Weka, features ('attributes') with a sufficiently low information gain have this value rounded down to "0"; these are the features we removed.

[4] The associated p-value is 6.56E-6.

knowledge, this is the first corpus-based approach to learning social power lects beyond those in direct reporting relationships.

Our work strongly suggests that statistically extracted features are an efficient and effective approach to modeling social information. Our methods exploit many aspects of language use and effectively model social power information while using statistical methods at every stage to tease out the information we seek, significantly reducing language-, culture-, and lect-specific engineering needs. Our feature selection method picks up on indicators suggested by sociolinguistics, and it also allows for the identification of features that are not obviously characteristic of UpSpeak or DownSpeak. Some easily recognizable features include:

Lect	Ngram	Example
UpSpeak	if you	"Let me know *if you* need anything." "Please call me *if you* have any questions."
DownSpeak	give me	"Read this over and *give me* a call." "Please *give me* your comments next week."

On the other hand, other features are less intuitive:

Lect	Ngram	Example
UpSpeak	I'll, we'll	"*I'll* let you know the final results soon" "Everyone is very excited […] and we're confident *we'll* be successful"
DownSpeak	that is, this is	"Neither does any other group but *that is* not my problem" "I think *this is* an excellent letter"

We hope to improve our methods for selecting and binning features with information theoretic selection metrics and clustering algorithms.

We also have begun work on 3-way, UpSpeak/ DownSpeak/PeerSpeak classification. Training a multiclass SVM on the binned n-gram features from (5) produces **51.6%** cross-validation accuracy on training data and **44.4%** accuracy on the weighted test set (both numbers should be compared to a 33% baseline). That classifier contained no n-gram features selected from the PeerSpeak class. Preliminary experiments incorporating PeerSpeak n-grams yield slightly better numbers. However, early results also suggest that the three-way classification problem is made more tractable with cascaded two-way classifiers; feature selection was more manageable with binary problems. For example, one classifier determines whether an instance is UpSpeak; if it is not, a second classifier distinguishes between DownSpeak and PeerSpeak. Our text classification problem is similar to sentiment analysis in that there are class dependencies; for example, DownSpeak is more closely related to PeerSpeak than to UpSpeak. We might attempt to exploit these dependencies in a manner similar to Pang and Lee (2005) to improve three-way classification.

In addition, we had promising early results for classification of author-recipient links with 200 to 500 words, so we plan to explore performance improvements for links of few words.

In early, unpublished work, we had promising results with generative model-based approach to SPM, and we plan to revisit it; language models are a natural fit for lect modeling. Finally, we hope to investigate how SPM and SNA can enhance one another, and explore other lect classification problems for which the ground truth can be found.

Acknowledgments

Dr. Richard Sproat contributed time, valuable insights, and wise counsel on several occasions during the course of the research. Dr. Lillian Lee and her students in *Natural Language Processing and Social Interaction* reviewed the paper, offering valuable feedback and helpful leads.

Our colleague, Diane Bramsen, created an excellent graphical interface for probing and understanding the results. Jeff Lau guided and advised throughout the project.

We thank our anonymous reviewers for prudent advice.

This work was funded by the Army Studies Board and sponsored by Col. Timothy Hill of the United Stated Army Intelligence and Security Command (INSCOM) Futures Directorate under contract W911W4-08-D-0011.

References

Cecilia Ovesdotter Alm, Dan Roth and Richard Sproat. 2005. Emotions from text: machine learning for text-based emotion prediction. *HLT/EMNLP 2005*. October 6-8, 2005, Vancouver.

Penelope Brown and Stephen C. Levinson. 1987. *Politeness: Some universals in language usage*. Cambridge: Cambridge University Press.

Eric Breck, Yejin Choi and Claire Cardie. 2007. Identifying expressions of opinion in context. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI-2007)*

CALO Project. 2009. Enron E-Mail Dataset. http://www.cs.cmu.edu/~enron/.

Yejin Choi and Claire Cardie. 2008. Learning with compositional semantics as structural inference for subsentential sentiment analysis. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Honolulu, Hawaii: ACM. 793-801.

Yejin Choi and Claire Cardie. 2009. Adapting a polarity lexicon using integer linear programming for domain-specific sentiment classification. *Empirical Methods in Natural Language Processing (EMNLP)*.

Christopher P. Diehl, Galileo Namata, and Lise Getoor. 2007. Relationship identification for social network discovery. *AAAI '07: Proceedings of the 22nd National Conference on Artificial Intelligence.*

Bonnie Erickson, et al. 1978. Speech style and impression formation in a court setting: The effects of 'powerful’ and 'powerless' speech. *Journal of Experimental Social Psychology 14*: 266-79.

Norman Fairclough. 1989. *Language and power*. London: Longman.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The WEKA data mining software: An update. *SIGKDD Exploration (1)*: Issue 1.

JHU Center for Imaging Science. 2005. Scan Statistics on Enron Graphs. http://cis.jhu.edu/~parky/Enron/

Soo-min Kim and Eduard Hovy. 2004. Determining the Sentiment of Opinions. *Proceedings of the COLING Conference*. Geneva, Switzerland.

Francois Mairesse and Marilyn Walker. 2006. Automatic recognition of personality in conversation. *Proceedings of HLT-NAACL.* New York City, New York.

Galileo Mark S. Namata Jr., Lise Getoor, and Christopher P. Diehl. 2006. Inferring organizational titles in online communication. *ICML 2006*, 179-181.

Andrew McCallum, Xuerui Wang, and Andres Corrada-Emmanuel. 2007. Topic and role discovery in social networks with experiments on Enron and academic e-Mail. *Journal of Artificial Intelligence Research* 29.

Ryan McDonald, Kerry Hannan, Tyler Neylon, Mike Wells, and Jeff Reynar. 2007. Structured models for fine-to-coarse sentiment analysis. *Proceedings of the ACL.*

David Morand. 2000. Language and power: An empirical analysis of linguistic strategies used in superior/subordinate communication. *Journal of Organizational Behavior*, *21*:235-248.

Frederick Mosteller and David L. Wallace. 1964. *Inference and disputed authorship: The Federalist*. Addison-Wesley, Reading, Mass.

Jon Oberlander and Scott Nowson. 2006. Whose thumb is it anyway? Classifying author personality from weblog text. *Proceedings of CoLing/ACL*. Sydney, Australia.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? Sentiment classification using machine learning techniques. *Proceedings of EMNLP*, 79–86.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. *Proceedings of the ACL.*

John Platt. 1998. Sequential minimal optimization: A fast algorithm for training support vector machines. In *Technical Report MST-TR-98-14*. Microsoft Research.

Delip Rao and Deepak Ravichandran. 2009. Semi-supervised polarity lexicon induction. *European Chapter of the Association for Computational Linguistics*.

Efstathios Stamatatos. 2009. A survey of modern authorship attribution methods. *JASIST 60(3)*: 538-556.

Carol Strapparava and Rada Mihalcea. 2008. Learning to identify emotions in text. *SAC 2008*: 1556-1560

Hiroya Takamura, Takashi Inui, and Manabu Okumura. 2005. Semantic Orientations of Words using Spin Model. *Annual Meeting of the Association for Computational Linguistics*.

Ian H. Witten and Eibe Frank. 2005. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kauffman.

Xiaojin Zhu. 2005. Semi-supervised learning literature survey. *Technical Report 1530*, Department of Computer Sciences, University of Wisconsin, Madison.