

# Predicting Power Relations between Participants in Written Dialog from a Single Thread

**Vinodkumar Prabhakaran**

Dept. of Computer Science  
Columbia University, New York, NY  
vinod@cs.columbia.edu

**Owen Rambow**

Cntr. for Comp. Learning Systems  
Columbia University, New York, NY  
rambow@ccls.columbia.edu

## Abstract

We introduce the problem of predicting who has power over whom in pairs of people based on a single written dialog. We propose a new set of structural features. We build a supervised learning system to predict the direction of power; our new features significantly improve the results over using previously proposed features.

## 1 Introduction

Computationally analyzing the social context in which language is used has gathered great interest within the NLP community recently. One of the areas that has generated substantial research is the study of how social power relations between people affect and/or are revealed in their interactions with one another. Researchers have proposed systems to detect social power relations between participants of organizational email threads (Bramsen et al., 2011; Gilbert, 2012; Prabhakaran and Rambow, 2013), online forums (Danescu-Niculescu-Mizil et al., 2012; Biran et al., 2012; Danescu-Niculescu-Mizil et al., 2013), chats (Strzalkowski et al., 2012), and off-line interactions such as presidential debates (Prabhakaran et al., 2013; Nguyen et al., 2013). Automatically identifying power and influence from interactions can have many practical applications ranging from law enforcement and intelligence to online marketing.

A significant number of these studies are performed in the domain of organizational email where there is a well defined notion of power (organizational hierarchy). Bramsen et al. (2011) and Gilbert (2012) predict hierarchical power relations between people in the Enron email corpus using lexical features extracted from *all* the messages exchanged between them. However, their approaches primarily apply to situations where large collections of messages exchanged between pairs

of people are available. In (Prabhakaran and Rambow, 2013), we introduced the problem of detecting whether a participant of an email thread has power over someone else in the thread and established the importance of dialog structure in that task. However, in that work we did not detect over whom that person has power.

In this paper, we introduce a new problem formulation. We predict the hierarchical power relation between pairs of participants in an email interaction thread based *solely* on features extracted from that thread. As a second major contribution, we introduce a new set of features to capture aspects of participant behavior such as responsiveness, and we show that these features are significantly correlated with the direction of power. We present a fully automatic system for this task obtaining an accuracy of 73.0%, an improvement of 6.9% over 68.3% by a system using only lexical features. This best-performing system uses our new feature set.

## 2 Motivation

Early NLP-based approaches such as Bramsen et al. (2011) and Gilbert (2012) built systems to predict hierarchical power relations between people in the Enron email corpus using lexical features from all the messages exchanged between them. One limitation of this approach is that it relies solely on lexical cues and hence works best when large collections of messages exchanged between the pairs of people are available. For example, Bramsen et al. (2011) excluded sender-recipient pairs who exchanged fewer than 500 words from their evaluation set, since they found smaller text samples are harder to classify. By taking the message out of the context of the interaction in which it was exchanged, they fail to utilize cues from the structure of interactions, which complements the lexical cues in detecting power relations, as we showed in (Prabhakaran and Rambow, 2013).

We modeled the problem of detecting power relationships differently in (Prabhakaran and Rambow, 2013): we predicted whether a participant in an email thread has a certain type of power or not. However, in that work we did not predict over whom he/she has that power. This may result in noisy features; consider a thread in which participant  $\mathcal{X}$  has power over participant  $\mathcal{Y}$ , who has power over participant  $\mathcal{Z}$ . By aggregating features over all messages sent by  $\mathcal{Y}$ , features salient to a subordinate-superior interaction are incorrectly conflated with those salient to superior-subordinate interaction. Another limitation of (Prabhakaran and Rambow, 2013) is that we used manual annotations for many of our features such as dialog acts and overt displays of power. Relying on manual annotations for features limited our analysis to a small subset of the Enron corpus, which has only 18 instances of hierarchical power. Consequently, our findings with respect to hierarchical power were weak in terms of both correlations of features and system performance.

In this paper, we introduce the problem of predicting who has power over whom in pairs of interacting participants based on a single thread of interactions. From (Bramsen et al., 2011) we retain the idea that we want to predict the power relation between pairs of people. But in contrast to their formulation, we retain the goal from (Prabhakaran and Rambow, 2013) that we want to study communication in the context of an interaction, and that we want to be able to make predictions using only the emails exchanged in a single thread. Like (Prabhakaran and Rambow, 2013), we use features to capture the dialog structure, but we use automatic taggers to generate them and assume no manual annotation at all at training or test time. This allows us to use the entire Enron email corpus for this study.

### 3 Data

In this work, we use the version of Enron email corpus by Yeh and Harnly (2006) which captures the thread structure of email exchanges. The corpus contains 36,615 email threads. We excluded a small subset of 419 threads that was used for previous manual annotation efforts, part of which was also used to train the DA and ODP taggers (Section 5) that generate features for our system. The average number of email messages per thread was around 3. We divided the remaining threads into

*train* (50%), *dev* (25%) and *test* (25%) sets by random sampling. We then applied various basic NLP preprocessing steps such as tokenization, POS tagging and lemmatization to the body of email messages. We use the Enron gold organizational hierarchy released by Agarwal et al. (2012) to model hierarchical power. Their corpus was manually built using information from Enron organizational charts. It includes relations of 1,518 employees and captures dominance relations between 13,724 pairs of them. This is the largest such data set available to the best of our knowledge.

## 4 Problem Formulation

Let  $t$  denote an email thread and  $M_t$  denote the set of all messages in  $t$ . Also, let  $P_t$  be the set of all participants in  $t$ , i.e., the union of senders and recipients (*To* and *CC*) of all messages in  $M_t$ . We are interested in detecting power relations between pairs of participants who interact within a given email thread. Not every pair of participants  $(p_1, p_2) \in P_t \times P_t$  interact with one another within  $t$ . Let  $IM_t(p_1, p_2)$  denote the set of *Interaction Messages* — non-empty messages in  $t$  in which either  $p_1$  is the sender and  $p_2$  is one of the recipients or vice versa. We call the set of  $(p_1, p_2)$  such that  $|IM_t(p_1, p_2)| > 0$  the *interacting participant pairs* of  $t$  ( $IPP_t$ ).

We focus on the manifestations of power in interactions between people across different levels of hierarchy. For every  $(p_1, p_2) \in IPP_t$ , we query the set of dominance relations in the gold hierarchy to determine their hierarchical power relation ( $HP(p_1, p_2)$ ). We exclude pairs that do not exist in the gold hierarchy from our analysis and denote the remaining set of *related interacting participant pairs* as  $RIPP_t$ . We assign  $HP(p_1, p_2)$  to be *superior* if  $p_1$  dominates  $p_2$ , and *subordinate* if  $p_2$  dominates  $p_1$ . Table 1 shows the total number of pairs in  $IPP_t$  and  $RIPP_t$  from all the threads in our corpus and across *train*, *dev* and *test* sets.

Description	Total	Train	Dev	Test
# of threads	36,196	18,079	8,973	9,144
$\sum_t  IPP_t $	355,797	174,892	91,898	89,007
$\sum_t  RIPP_t $	15,048	7,510	3,578	3,960

Table 1: Data Statistics

Row 1 presents the total number of threads in different subsets of the corpus. Row 2 and 3 present the number of interacting participant pairs ( $IPP$ ) and related interacting participant pairs ( $RIPP$ ) in those subsets.

Given a thread  $t$  and a pair of participants  $(p_1, p_2) \in RIPP_t$ , we want to automatically detect  $HP(p_1, p_2)$ . This problem formulation is similar to the ones in (Bramsen et al., 2011) and (Gilbert, 2012). However, the difference is that for us an instance is a pair of participants in a single thread of interaction (which may or may not include other people), whereas for them an instance constitutes all messages exchanged between a pair of people in the entire corpus. Our formulation also differs from (Prabhakaran and Rambow, 2013) in that we detect power relations between pairs of participants, instead of just whether a participant had power over anyone in the thread.

## 5 Structural Analysis

In this section we analyze various features that capture the structure of interaction between the pairs of participants in a thread. Each feature  $f$  is extracted with respect to a person  $p$  over a reference set of messages  $M$  (denoted  $f_M^p$ ). For a pair  $(p_1, p_2)$ , we extract 4 versions of each feature  $f$ :  $f_{IM_t(p_1, p_2)}^{p_1}$ ,  $f_{IM_t(p_1, p_2)}^{p_2}$ ,  $f_{M_t}^{p_1}$  and  $f_{M_t}^{p_2}$ . The first two capture behavior of the pair among themselves, while the third and fourth capture their overall behavior in the entire thread. We group our features into three categories —  $THR^{New}$ ,  $THR^{PR}$  and  $DIA^{PR}$ .  $THR^{New}$  is a set of new features we propose, while  $THR^{PR}$  and  $DIA^{PR}$  incorporate features we proposed in (Prabhakaran and Rambow, 2013).  $THR^{New}$  and  $THR^{PR}$  capture the structure of message exchanges without looking at the content of the emails (e.g., how many emails did a person send), while  $DIA^{PR}$  captures the pragmatics of the dialog and requires an analysis of the content of the emails (e.g., did they issue any requests).

**$THR^{New}$ :** This is a new set of features we introduce in this paper. It includes the average number of recipients (AvgRecipients) and *To* recipients (AvgToRecipients) in emails sent by  $p$ , the percentage of emails  $p$  received in which he/she was in the *To* list (InToList%), boolean features denoting whether  $p$  added or removed people when responding to a message (AddPerson and RemovePerson), average number of replies received per message sent by  $p$  (ReplyRate) and average number of replies received from the other person of the pair to messages where he/she was a *To* recipient (ReplyRateWithinPair). ReplyRateWithinPair applies only to  $IM_t(p_1, p_2)$ .

**$THR^{PR}$ :** This feature set includes two meta-

data based feature sets — positional and verbosity. Positional features include a boolean feature to denote whether  $p$  sent the first message (Initiate), and relative positions of  $p$ 's first and last messages (FirstMsgPos and LastMsgPos) in  $M$ . Verbosity features include  $p$ 's message count (MsgCount), message ratio (MsgRatio), token count (TokenCount), token ratio (TokenRato) and tokens per message (TokenPerMsg), all calculated over  $M$ .

**$DIA^{PR}$ :** In (Prabhakaran and Rambow, 2013), we used dialog features derived from manual annotations — dialog acts (DA) and overt displays of power (ODP) — to model the structure of interactions within the message content. In this work, we obtain DA and ODP tags on the entire corpus using automatic taggers trained on those manual annotations. The DA tagger (Omuya et al., 2013) obtained an accuracy of 92%. The ODP tagger (Prabhakaran et al., 2012) obtained an accuracy of 96% and F-measure of 54%. The DA tagger labels each sentence to be one of the 4 dialog acts: Request Action, Request Information, Inform, and Conventional. The ODP Tagger identifies sentences (mostly requests) that express additional constraints on its response, beyond those introduced by the dialog act. We use 5 features: ReqAction%, ReqInform%, Inform%, Conventional%, and ODP% to capture the percentage of sentences in messages sent by  $p$  that has each of these labels. We also use a feature to capture the number of  $p$ 's messages with a request that did not get a reply, i.e., dangling requests (DanglingReq%), over all messages sent by  $p$ .

We perform an unpaired two-sample two-tailed Student's t-Test comparing mean values of each feature for *subordinates* vs. *superiors*. For our analysis, a data point is a related interacting pair, and not a message. Hence, a message with multiple recipients who have a *superior/subordinate* relation with the sender will contribute to features for multiple data points. We limit our analysis to the related interacting pairs from only our *train* set. Table 2 presents mean values of features for *subordinates* and *superiors* at the interaction level. Thread level versions of these features also obtained similar results overall in terms of direction of difference and significance. We denote three significance levels — \* ( $p < .05$ ), \*\* ( $p < .01$ ), and \*\*\* ( $p < .001$ ). To control false discovery rates in multiple testing, we adjusted the p-values (Benjamini and Hochberg, 1995). We summarize

Feature Name	Mean( $f_{IM_t}^{sub}$ )	Mean( $f_{IM_t}^{sup}$ )
<i>THR<sup>New</sup></i>		
AvgRecipients***	21.14	43.10
AvgToRecipients***	18.19	38.94
InToList%	0.82	0.80
ReplyRate***	0.86	1.23
ReplyRateWithinPair***	0.16	0.10
AddPerson	0.48	0.47
RemovePerson***	0.41	0.37
<i>THR<sup>PR</sup></i>		
Initiate***	0.45	0.56
FirstMsgPos	0.04	0.03
LastMsgPos***	0.15	0.11
MsgCount***	0.64	0.70
MsgRatio***	0.44	0.56
TokenCount	91.22	83.26
TokenRatio***	0.45	0.55
TokenPerMsg*	140.60	120.87
<i>DIA<sup>PR</sup></i>		
Conventional%***	0.15	0.17
Inform%***	0.78	0.72
ReqAction%***	0.02	0.04
ReqInform%***	0.05	0.06
DanglingReq%***	0.12	0.15
ODP%***	0.03	0.06

Table 2: Student’s t-Test Results of  $f_{IM_t}^P$ .  
 $THR^{New}$ : new meta-data features;  $THR^{PR}$ ,  $DIA^{PR}$ : meta-data  
and dialog-act features from previous studies;  
\* ( $p < .05$ ); \*\* ( $p < .01$ ); \*\*\* ( $p < .001$ )

the main findings on the significant features below.

1. Superiors send messages addressed to more people (AvgRecipients and AvgToRecipients). Consequently, they get more replies to their messages (ReplyRate). However, considering messages where the other person of the pair is addressed in the *To* list (ReplyRate-WithinPair), subordinates get more replies.
2. Superiors issue more requests (ReqAction% and ReqInform%) and overt displays of power (ODP%). Subordinates issue more informs (Inform%) and, surprisingly, have fewer unanswered requests (DanglingReq%).
3. Superiors initiate the interactions more often than subordinates (Initiate). They also leave interactions earlier (LastMsgPos).
4. Superiors send shorter messages (TokenPerMsg). They also send more messages (MsgCount & MsgRatio) and even contribute a higher ratio of tokens in the thread (TokenRatio) despite sending shorter messages.

Finding 1 goes in line with findings from studies analyzing social networks that superiors have higher connectivity in the networks that they are part of (Rowe et al., 2007). Intuitively, those who have higher connectivity also send emails to larger number of people, and hence our result. Since superiors address more people in their emails, they also have a higher chance of getting replies. Finding 2 also aligns with the general intuition about how superiors and subordinates behave within interactions (e.g., superiors exhibit more overt displays of power than subordinates).

Findings 3 & 4 are interesting since they reveal special characteristics of threads involving hierarchically related participants. In (Prabhakaran and Rambow, 2013), we had found that persons with hierarchical power rarely initiated threads and contributed less within the threads. But that problem formulation was different — we were identifying whether a person in a given thread had hierarchical power over someone else or not. The data points in that formulation included participants from threads that did not have any hierarchically related people, whereas our current formulation do not. These findings suggest that if a person starts an email thread, he’s likely not to be the one who has power, but if a thread includes a pair of people who are hierarchically related, then it is likely to be initiated by the superior and he/she tends to contribute more in such threads.

## 6 Predicting Direction of Power

We build an SVM-based supervised learning system that can predict  $HP(p_1, p_2)$  to be either *superior* or *subordinate* based on the interaction within a thread  $t$  for any pair of participants  $(p_1, p_2) \in RIPP_t$ . We deterministically fix the order of participants in  $(p_1, p_2)$  such that  $p_1$  is the sender of the first message in  $IM_t(p_1, p_2)$ . We use the ClearTK (Ogren et al., 2008) wrapper for SVMLight (Joachims, 1999) in our experiments. We use the related interacting participant pairs in threads from the *train* set to train our models and optimize our performance on those from the *dev* set. We report results obtained on *dev* and *test* sets.

In our formulation, values of many features are undefined for some instances (e.g., Inform% is undefined when MsgCount = 0). Handling of undefined values for features in SVM is not straightforward. Most SVM implementations assume the value of 0 by default in such cases, conflating them

Description	Accuracy
Baseline (Always Superior)	52.54
Baseline (Word Unigrams + Bigrams)	68.56
THR <sup>New</sup>	55.90
THR <sup>PR</sup>	54.30
DIA <sup>PR</sup>	54.05
THR <sup>PR</sup> + THR <sup>New</sup>	61.49
DIA <sup>PR</sup> + THR <sup>PR</sup> + THR <sup>New</sup>	62.47
LEX	70.74
LEX + DIA <sup>PR</sup> + THR <sup>PR</sup>	67.44
LEX + DIA <sup>PR</sup> + THR <sup>PR</sup> + THR <sup>New</sup>	68.56
BEST (= LEX + THR <sup>New</sup> )	<b>73.03</b>
BEST (Using $p_1$ features only)	72.08
BEST (Using $IM_t$ features only)	72.11
BEST (Using $M_t$ only)	71.27
BEST (No Indicator Variables)	72.44

Table 3: Accuracies on feature subsets (*dev* set). THR<sup>New</sup>: new meta-data features; THR<sup>PR</sup>, DIA<sup>PR</sup>: meta-data and dialog-act features from previous studies; LEX: ngrams; BEST: best subset;  $IM_t$  stands for  $IM_t(p_1, p_2)$

with cases where Inform% is truly 0. In order to mitigate this issue, we use an indicator feature for each structural feature to denote whether or not it is valid. Since we use a quadratic kernel, we expect the SVM to pick up the interaction between each feature and its indicator feature.

Lexical features have already been shown to be valuable in predicting power relations (Bramsen et al., 2011; Gilbert, 2012). We use another feature set LEX to capture word ngrams, POS (part of speech) ngrams and mixed ngrams. A mixed ngram (Prabhakaran et al., 2012) is a special case of word ngram where words belonging to open classes are replaced with their POS tags. We found the best setting to be using both unigrams and bigrams for all three types of ngrams, by tuning in our *dev* set. We then performed experiments using all subsets of {LEX, THR<sup>New</sup>, THR<sup>PR</sup>, DIA<sup>PR</sup>}.

Table 3 presents the results obtained using various feature subsets. We use a majority class baseline assigning  $HP(p_1, p_2)$  to be always *superior*, which obtains 52.5% accuracy. We also use a stronger baseline using word unigrams and bigrams as features, which obtained an accuracy of 68.6%. The performance of the system using each structural feature class on its own is very low. Combining all three of them improves the accuracy to 62.5%. The highest performance obtained without using any message content is for THR<sup>PR</sup> and THR<sup>New</sup> (61.5%). LEX features by

itself obtain a very high accuracy of 70.7%, confirming the importance of lexical patterns in this task. Perplexingly, adding all structural features to LEX reduces the accuracy by around 2.2 percentage points. The best performing system (BEST) uses LEX and THR<sup>New</sup> features and obtains an accuracy of 73.0%, a statistically significant improvement over the LEX-only system (McNemar).

We also performed an ablation study to understand the importance of different slices of our feature sets. If we remove all feature versions with respect to the second person, the accuracy drops to 72.1%. This suggests that features about the other person’s behavior also help the prediction task. If we remove either the thread level versions of features or interaction level versions of features, the accuracy again drops, suggesting that both the pair’s behavior among themselves, and their overall behavior in the thread add value to the prediction task. Removing the indicator feature denoting the structural features’ validity also reduces the performance of the system.

We now discuss evaluation on our blind test set. The majority baseline (Always Superior) for accuracy is 55.0%. The word unigrams and bigrams baseline obtains an accuracy of 68.3%. The LEX system (using other forms of ngrams as well) obtains a slightly lower accuracy of 68.1%. Our BEST system using LEX and THR<sup>New</sup> features obtains an accuracy of 73.0% (coincidentally the same as on the *dev* set), an improvement of 6.9% over the system using only lexical features.

## 7 Conclusion

We introduced the problem of predicting who has power over whom based on a single thread of written interactions. We introduced a new set of features which describe the structure of the dialog. Using this feature set, we obtain an accuracy of 73.0% on a blind test. In future work, we will tackle the problem of three-way classification of pairs of participants, which will cover cases in which they are not in a power relation at all.

## Acknowledgments

This paper is based upon work supported by the DARPA DEFT Program. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government. We also thank several anonymous reviewers for their feedback.

## References

- Apoorv Agarwal, Adinoyi Omuya, Aaron Harnly, and Owen Rambow. 2012. A Comprehensive Gold Standard for the Enron Organizational Hierarchy. In *Proceedings of the 50th Annual Meeting of the ACL (Short Papers)*, pages 161–165, Jeju Island, Korea, July. Association for Computational Linguistics.
- Yoav Benjamini and Yocef Hochberg. 1995. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 289–300.
- Or Biran, Sara Rosenthal, Jacob Andreas, Kathleen McKeown, and Owen Rambow. 2012. Detecting influencers in written online conversations. In *Proceedings of the Second Workshop on Language in Social Media*, pages 37–45, Montréal, Canada, June. Association for Computational Linguistics.
- Philip Bramsen, Martha Escobar-Molano, Ami Patel, and Rafael Alonso. 2011. Extracting social power relationships from natural language. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 773–782, Portland, Oregon, USA, June. Association for Computational Linguistics.
- Cristian Danescu-Niculescu-Mizil, Lillian Lee, Bo Pang, and Jon Kleinberg. 2012. Echoes of power: language effects and power differences in social interaction. In *Proceedings of the 21st international conference on World Wide Web, WWW '12*, New York, NY, USA. ACM.
- Cristian Danescu-Niculescu-Mizil, Moritz Sudhof, Dan Jurafsky, Jure Leskovec, and Christopher Potts. 2013. A computational approach to politeness with application to social factors. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 250–259, Sofia, Bulgaria, August. Association for Computational Linguistics.
- Eric Gilbert. 2012. Phrases that signal workplace hierarchy. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work, CSCW '12*, pages 1037–1046, New York, NY, USA. ACM.
- Thorsten Joachims. 1999. Making Large-Scale SVM Learning Practical. In Bernhard Schölkopf, Christopher J.C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, Cambridge, MA, USA. MIT Press.
- Viet-An Nguyen, Jordan Boyd-Graber, Philip Resnik, Deborah A. Cai, Jennifer E. Midberry, and Yuanxin Wang. 2013. Modeling topic control to detect influence in conversations using nonparametric topic models. *Machine Learning*, pages 1–41.
- Philip V. Ogren, Philipp G. Wetzler, and Steven Bethard. 2008. ClearTK: A UIMA toolkit for statistical natural language processing. In *Towards Enhanced Interoperability for Large HLT Systems: UIMA for NLP workshop at Language Resources and Evaluation Conference (LREC)*.
- Adinoyi Omuya, Vinodkumar Prabhakaran, and Owen Rambow. 2013. Improving the quality of minority class identification in dialog act tagging. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 802–807, Atlanta, Georgia, June. Association for Computational Linguistics.
- Vinodkumar Prabhakaran and Owen Rambow. 2013. Written dialog and social power: Manifestations of different types of power in dialog behavior. In *Proceedings of the IJCNLP*, pages 216–224, Nagoya, Japan, October. Asian Federation of Natural Language Processing.
- Vinodkumar Prabhakaran, Owen Rambow, and Mona Diab. 2012. Predicting Overt Display of Power in Written Dialogs. In *Human Language Technologies: The 2012 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Montreal, Canada, June. Association for Computational Linguistics.
- Vinodkumar Prabhakaran, Ajita John, and Dorée D. Seligmann. 2013. Who had the upper hand? ranking participants of interactions based on their relative power. In *Proceedings of the IJCNLP*, pages 365–373, Nagoya, Japan, October. Asian Federation of Natural Language Processing.
- Ryan Rowe, German Creamer, Shlomo Hershkop, and Salvatore J. Stolfo. 2007. Automated social hierarchy detection through email network analysis. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web Mining and Social Network Anal.* ACM.
- Tomek Strzalkowski, Samira Shaikh, Ting Liu, George Aaron Broadwell, Jenny Stromer-Galley, Sarah Taylor, Umit Boz, Veena Ravishankar, and Xiaoai Ren. 2012. Modeling leadership and influence in multi-party online discourse. In *Proceedings of COLING*, pages 2535–2552, Mumbai, India, December. The COLING 2012 Organizing Committee.
- Jen-Yuan Yeh and Aaron Harnly. 2006. Email thread reassembly using similarity matching. In *CEAS 2006 - The Third Conference on Email and Anti-Spam, July 27-28, 2006, Mountain View, California, USA*, Mountain View, California, USA, July.