Name Reference Resolution in Organizational Email Archives

Christopher P. Diehl* Lise Getoor† Galileo Namata‡

Abstract

Online communications provide a rich resource for understanding social networks. Information about the actors, and their dynamic roles and relationships, can be inferred from both the communication content and traffic structure. A key component in the analysis of online communications such as email is the resolution of name references within the body of the message. Name reference resolution relies on the context of the message; both the content of the message and the sender and recipients' relationships can help to resolve a reference. Here we investigate a variety of approaches which make use of the email traffic network to disambiguate email name references. The email traffic network serves as a proxy for inferring relationships. These relationships in turn help us infer likely candidates for the name references. Our initial findings suggest that simple temporal models can help us effectively resolve name references. For the class of models proposed, performance is maximized by exploiting long-term traffic statistics to rank candidates.

1 Introduction

Within the networked world, email has become a ubiquitous form of global communication. Whether communicating with friends or colleagues in a local area or halfway around the world, email allows us to maintain or develop relationships with others at any distance. Given email traffic is a reflection of the relationships in an underlying social network, email archives present a potentially rich collection of evidence that can be used to infer the structure, attributes and dynamics of the social network. The challenge is to infer these properties from email data that is often ambiguous, incomplete and context-dependent.

Email collections contain both structured and unstructured data. The structured data or metadata indicates which parties communicated and when the communication occurred. By focusing solely on the metadata, we can identify communication patterns, but we cannot easily ascribe meaning to the underlying relationships. The unstructured data in the body of the email can clarify the roles of individuals and their relationships with others. Yet without the appropriate context, an outside observer may find a message provides little insight.

When communicating with others, people constantly rely on shared context to simplify communication. Shared context is common knowledge among individuals that allows them to use ambiguous references which are clear within the shared context. A common example of this occurs when two people refer to a mutual friend by a first name or a nickname in conversation.

"How's John doing today? Is he feeling better"

Given the topic of conversation along with the name reference, it is clear to both parties who John is. Yet to someone without knowledge of the context, the reference is meaningless.

Consider the problem of exploiting name references in the email body. Name references are an important element in understanding the social network. Before we can process the email content in the archive and associate activities and other attributes with individuals, we need to infer the number and identities of the individuals generating the observed traffic. Each individual has two classes of references: *network references* and *name references*. Network references in the context of email are simply the individual's email addresses. Note that this is potentially a many-many mapping: individuals may have multiple email addresses and a single email address may serve more than one individual. There is also a temporal component; an individual may have one email address for the time they are in one position in the company, but when they change roles within the company, perhaps moving to another division, their email address may change. Name references are the various forms of an individual's given name along with their nicknames that may appear in the email body. In order to define an individual's identity and draw broader connections across emails in the archive, we need to be able to map both name references and network references to the individual.

In this paper, we focus on the problem of mapping ambiguous name references, specifically first name references, to network references. The core challenge in this problem is identifying ways to exploit context from the email archive to effectively resolve the ambiguity. We describe this in the next section. Next, we formally define the general problem of name reference resolution.

*Johns Hopkins Applied Physics Laboratory, 11100 Johns Hopkins Road, Laurel, MD 20723, Chris.Diehl@jhuapl.edu

†Computer Science Department/UMIACS, University of Maryland, College Park, MD 20742, getoor@cs.umd.edu

‡Computer Science Department/UMIACS, University of Maryland, College Park, MD 20742, namatag@cs.umd.edu

Then we discuss the types of context available that can potentially be exploited. We investigate several different approaches, which vary in the context features and temporal models used, and introduce a methodology for evaluating their performance. Finally we present results from our algorithm evaluation on the Enron email archive and conclude with thoughts on future work.

2 Exploiting Context

When reading email, what types of context do we exploit to resolve ambiguous name references? In addition, what context does an email collection offer when analyzing relationships retrospectively? Below is a list of some of the contextual cues available to us for understanding name references.

- The participants in the conversation
- The larger group of people known by the participants in the conversation and the types of relationships among them
- The individuals that the participants in the conversation have recently communicated with, either before or after the email was sent
- The topic of conversation in the email
- Recent topics of conversation among the participants and others outside the current conversation, either before or after the email was sent
- Cues contained within other emails in the thread
- Related name references within the current email
- Prior knowledge linking individuals to topics of conversation

This list of contextual cues is by no means exhaustive. Yet it reminds us of the two broad classes of context that email captures: *social context* (who's talking?) and *topical context* (what are they talking about?). Our long term goal is to exploit both to characterize the underlying social network, as each form of context can help clarify ambiguities in the other. Yet the challenge of capturing and exploiting dynamic topical context is a significant research thrust on its own, as evidenced by the work in the topic detection and tracking community [3].

Our focus in this paper will be to investigate the discriminative power of dynamic social context. We want to first understand the performance of algorithms that leverage the patterns of communication among network references to estimate the mapping between name and network references.

3 Problem Definition

Let $\mathcal{E} = \{e_i\}$ be a set of email addresses observed in the email collection and let $\mathcal{N} = \{n_j\}$ be a set of observed name references in the email bodies. The set $\mathcal{E}$ may be extracted from the email metadata, or the set may come from another source such as an employee directory, which lists individuals together with their emails. The set $\mathcal{N}$ is the result of an entity extraction process that identifies name references within the email bodies.

The objective of name reference resolution is to construct a mapping from a set of observed name references $\mathcal{N} = \{n_j\}$ in the email collection to either ranked subsets of network references, $\mathcal{E}_j$, where $\mathcal{E}_j \subseteq \mathcal{E}$ or the null network reference ϕ, if no network reference is sufficiently probable. The null network reference ϕ serves two purposes. First, it is not a given that there exists a corresponding network reference for each name reference. An email collection may not contain email exchanges between all individuals referenced within the email bodies. Second, the entity extraction process will incorrectly declare some terms in the email collection to be name references, for which there is no network reference. In both cases, the appropriate response is to map the given name reference to ϕ.

For each name reference n_j, the corresponding candidate set $\mathcal{E}_j$ is ranked based on the context of the name reference. A scoring function g is used to compute the strength of association $g(e_c, n_j|C_j)$ between each candidate $e_c \in \mathcal{E}_j$ and n_j, given the context C_j associated with n_j. Once all of the candidates have been scored, they are ranked in descending order and only those candidates with scores $g(e_c, n_j|C_j) > \lambda$ are retained. The most likely network reference $\tilde{e}(n_j)$ is either the candidate with the maximum score greater than the threshold or ϕ otherwise.

In this paper, we explore the use of the email traffic context for ranking the candidate set. We define the *email traffic network* for a set of email messages $\mathcal{M} = \{m_i\}$ as follows: we have a directed hypergraph $\mathcal{G}_\mathcal{M}$ with the set of vertices $\mathcal{E}$ and hyperedges $\mathcal{H} = \{(e_{s_i}, \mathcal{E}_{r_i}, t_i)\}$. For each email message m_i, there is a hyperedge from the sender network reference e_{s_i}, $e_{s_i} \in \mathcal{E}$, to the set of recipients of the message, $\mathcal{E}_{r_i} \subseteq \mathcal{E}$. The attribute t_i is the time at which the email was sent.

4 Name Reference Resolution Process

The general name resolution process is composed of three phases: *candidate set generation*, *candidate ranking* and *candidate rejection* illustrated in figure 1. Given we envision a data analyst reviewing the candidate associations in rank order to identify the true referent, our overall goal is to minimize the number of candidates the user must evaluate while identifying as many true net-

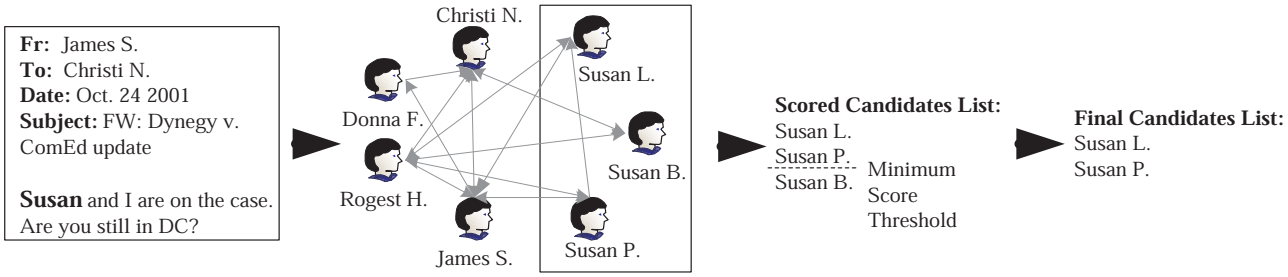

Figure 1: The name reference resolution process

work references as possible. When the true referent is a member of the candidate set, we want the algorithm to rank the true network reference as high as possible. Given the true referent may not be part of the candidate set at all, we also want to reject as many candidates as possible without severely impacting recall.

4.1 Candidate Set Generation The role of the candidate set is to restrict our attention to a small number of likely candidates prior to scoring the candidates. In our initial approach, we use two levels of screening. We begin with the strong assumption that if any communication has occurred between the true referent and the email participants, the sender was involved. Therefore we initially restrict the candidate set to those network references where at least one email communication has been observed with the sender.

Although we expect this assumption will be true in many cases, there are clearly instances where it will break down. For example, not all name references correspond to individuals that the email sender knows personally. Within the context of an organization, references may be made to individuals many levels removed in the management hierarchy. It is also not a given that an active relationship will be observable through email communication. The parties involved may be in close physical proximity allowing direct communication or may use other means of communication. A third possibility is that the email communications are simply not available in the email collection for one of a variety of reasons. Regardless of these factors, as we show in the results section, we are able to achieve suprisingly high recall.

Our second level of screening relies on available name information for the network references. We assume that some name information is initially available either from the name tags attached to email addresses or from the email addresses themselves. In our initial experiments, we examine name references that match exactly at least the first or last name associated with the candidate network reference. Clearly this constraint can be relaxed by employing a string comparison function to look for close name matches.

4.2 Candidate Scoring As mentioned earlier, our main interest is in defining and evaluating candidate scoring functions that leverage dynamic social context. If we begin with the presumption that name reference usage is often connected to events occurring around the time of the reference, the question is what fraction of the name references can we resolve by ranking candidates based on the level of email traffic around the time of the reference? To explore this, we introduce a class of scoring functions and explore the sensitivity of their performance along four general dimensions.

- The relationships examined
- The time scale at which the email traffic is viewed
- The summary statistic used to characterize relationship activity during a given time interval
- The degree of traffic history considered

We consider each of these dimensions next and then describe two temporal models which make use of features defined according to these dimensions.

4.2.1 Relationships Given our assumption of direct communication between at least the email sender and the true referent, our objective is to characterize the degree of communication between the email participants, the sender and recipients $\mathcal{E}_p = e_s \cup \mathcal{E}_{r_i}$, and the candidate network reference e_c.

Specifically we consider models that exploit either solely the traffic between the sender e_s and the candidate e_c (denoted **sender-only**) or models that exploit the pairwise traffic between all the email participants, sender and recipients, and the candidate e_c (denoted **sender+recipients**). When integrating traffic from the sender and recipients' pairwise interactions with the candidate, we want to understand the relative discrimination power offered by each and identify summary

statistics that effectively leverage the relationships for candidate scoring.

4.2.2 Time Scale To examine the email traffic at a given time scale, we first partition the time axis into regular intervals of duration Δt. The phase of the partition is fixed by first selecting a reference time t_0 such that $t_0 \leq t_r < t_0 + \Delta t$ where t_r is the time of the email containing the name reference. The time intervals $\{T_k\}$ are defined as $T_k = \{t' : t_0 + k\Delta t \leq t' < t_0 + (k+1)\Delta t, k \in \mathbb{Z}\}$ so that the time interval T_0 includes the time t_r of the email [1]. In our experiments, we investigate daily and weekly time intervals (denoted **daily** and **weekly**). The weekly time intervals are phased such that they begin on Sunday.

4.2.3 Summary Statistics Once the time axis is partitioned into regular intervals, our next step is to compute a summary statistic or feature $s(\mathcal{E}_p, e_c, T_k, \mathcal{G}_\mathcal{M})$ for each interval T_k that provides an indication of relationship activity among some or all of the email participants $\mathcal{E}_p$ and the candidate e_c. We consider the following variations on computing the statistic:

Binary versus Count. For any pair of network references, for the given interval, we may either have a 0/1 indicator which denotes whether or not there has been an email exchange between the pair (denoted **binary**) or we may want to use the frequency information and keep track of the number of messages exchanged (denoted **count**).

Unidirectional versus Bidirectional. For any network reference, we may be interested in only the messages sent from the network reference to the candidate reference (denoted **unidirectional**) or we may be interested in bidirectional exchanges where the candidate and network references can take on either the sender or recipient roles (denoted **bidirectional**)

As mentioned earlier, we can distinguish models which make use of the sender-only traffic information versus the sender+recipients traffic information. In the latter case, we introduce the parameter β to weight the sender versus recipient contributions. Table 1 summarizes the statistics used in the experiments.

4.2.4 Integrating Traffic History The final step in computing the candidate score $g(e_c, n|C)$ given the context $C = \{\mathcal{E}_p, T_0, \mathcal{G}_\mathcal{M}\}$ involves integrating the summary statistics for time intervals around the time of the email containing the name reference. We compute the local time average of the summary statistics using either a non-causal autoregressive (denoted **AR**) or moving average filter (denoted **MA**) that incorporates both future and past traffic patterns around the time of the name reference. The autoregressive filter is defined as

$$\begin{aligned} g_{\text{AR}}(e_c, n|\mathcal{E}_p, T_k, \mathcal{G}_\mathcal{M}) = \\ \tfrac{(1-\alpha)}{2} g_{\text{AR}}(e_c, n|\mathcal{E}_p, T_{k-1}, \mathcal{G}_\mathcal{M}) + \\ \tfrac{(1-\alpha)}{2} g_{\text{AR}}(e_c, n|\mathcal{E}_p, T_{k+1}, \mathcal{G}_\mathcal{M}) + \\ \alpha s(\mathcal{E}_p, e_c, T_k, \mathcal{G}_\mathcal{M}) \\ = \tfrac{\alpha}{2} \sum_{i=0}^{\infty} (1-\alpha)^i (s(\mathcal{E}_p, e_c, T_{k-i}, \mathcal{G}_\mathcal{M}) \\ + s(\mathcal{E}_p, e_c, T_{k+i}, \mathcal{G}_\mathcal{M})) \end{aligned}$$

while the moving average filter is defined as

$$g_{\text{MA}}(e_c, n|\mathcal{E}_p, T_k, \mathcal{G}_\mathcal{M}) = \frac{1}{2\text{M}+1} \sum_{i=-M}^{M} s(\mathcal{E}_p, e_c, T_{k-i}, \mathcal{G}_\mathcal{M}).$$

In practice, when evaluating the AR filter, we terminate the summation once a convergence criterion is met. The degree of traffic history incorporated into the candidate score $g(e_c, n|\mathcal{E}_p, T_0, \mathcal{G}_\mathcal{M})$ is controlled by the parameters α for the AR model and M for the MA model.

4.3 Candidate Rejection Once the scores have been computed for all network references in the candidate set, the candidates with a score below the specified threshold λ are removed from the candidate set. The objective of candidate rejection is to remove candidates that are deemed unlikely to correspond to name references without rejecting a significant fraction of true referents. The degree of performance achieved is dependent on the ability of the scoring function to separate the true referents from the other candidates.

Within the context of the models proposed above, performance clearly depends on the following two factors. First, it is dependent on the legitimacy of the general assumption that a high degree of communication activity around the time of the name reference is indicative of a potential correspondence between a name and network reference. Second, performance is also dependent on the model's characterization of what qualifies as a high degree of traffic. All relationships are clearly not equivalent. Yet our baseline models do not attempt to capture external factors influencing the relationship activity. We will revisit these issues in later discussion.

[1] Although the definition of T_0 is dependent on the email of interest, we will not explicitly indicate this dependence to avoid additional complexity in the notation.

Table 1: Summary statistic definitions. $m(e_1, e_2, T_k, \mathcal{G}_\mathcal{M})$ is the number of messages sent from network reference e_1 to network reference e_2 over the time interval T_k. $\mathcal{I}(\cdot)$ is the indicator function.

Name	*Definition*
Binary, Sender-Only, Bidirectional	$\mathcal{I}(m(e_s, e_c, T_k, \mathcal{G}_\mathcal{M}) + m(e_c, e_s, T_k, \mathcal{G}_\mathcal{M}))$
Count, Sender-Only, Bidirectional	$m(e_s, e_c, T_k, \mathcal{G}_\mathcal{M}) + m(e_c, e_s, T_k, \mathcal{G}_\mathcal{M})$
Count, Sender-Only, Unidirectional	$m(e_s, e_c, T_k, \mathcal{G}_\mathcal{M})$
Count, Sender+Recipients, Bidirectional(β)	$(1-\beta)(m(e_s, e_c, T_k, \mathcal{G}_\mathcal{M}) + m(e_c, e_s, T_k, \mathcal{G}_\mathcal{M}))+$ $\frac{\beta}{\lvert\mathcal{E}_{r_i}\rvert}\sum_{e_{r_i}\in\mathcal{E}_{r_i}}(m(e_{r_i}, e_c, T_k, \mathcal{G}_\mathcal{M}) + m(e_c, e_{r_i}, T_k, \mathcal{G}_\mathcal{M}))$

5 Experiment Design

With a set of models defined, the next major task at hand is evaluating their performance on a representative dataset. The bulk of our efforts to date have focused on data preparation, ground truth generation and definition of evaluation protocols. A number of subtle but important issues arise as one considers the various elements of the overall experiment. We review all aspects of the approach in the following sections.

5.1 Dataset Preparation

5.1.1 The Data: Enron Email Corpus With the recent release of the Enron email dataset [20], researchers have been given a unique opportunity to glimpse inside a large corporation and observe a subset of email traffic among the employees. The Enron email dataset is the collection of email from the folders of 151 Enron employees. The data is available in several forms. CMU first released the original email data. Since then USC/ISI and more recently UC Berkeley have released normalized forms of the data in a MySQL database. Our results are based on the USC/ISI version of the dataset. There are over 250,000 email messages in the dataset with the majority of the traffic occurring in the 2000-2002 time frame.

We initially chose to resolve name references in only those emails exchanged between the core 151 employees. This was done primarily to reduce confounding effects of observability in our experiments. Given we can only observe pairwise relationships where at least one of the participants is a member of the set of 151 employees, constraining the set of emails in this way guarantees that all relationships we consider in the resolution process are observable in the email collection, assuming emails haven't been lost or deleted.

There are 7644 emails in the ISI database that were exchanged among the 151 employees. A non-trivial number of duplicate emails exist that were removed to avoid skewing the results of the analysis. After deduplication of this set, 6550 emails remain. This is the set of emails from which the name references were extracted.

5.1.2 Extracting Enron Employee Names To support named entity extraction and candidate set generation, we constructed a network reference set $\mathcal{E}$ of 7864 Enron email addresses and a corresponding list of employee names by parsing the email addresses. In total, there are 29,176 *enron.com* email addresses in the collection. This includes employee email addresses along with group mailing lists. Given the most common email address format often corresponding to employees is $<$*name1*$>$. $<$*name2*$>$ @*enron.com*, we parsed these addresses and saved only those where either *name1* or *name2* matched a first or last name in the employeelist table in the ISI database. This reduced the list to 7713 email addresses that are distinct from the email addresses listed for the 151 employees in the ISI database.

As others have noted, some employees have multiple email addresses in the collection. We believe that in most cases this is due to an employee moving within the company. Therefore each email address and its associated relationship structure characterizes the employee's role over a certain time period in the company. We chose not to deduplicate the email addresses in order to preserve this context.

5.1.3 Constructing the Email Traffic Network The hypergraph $\mathcal{G}_\mathcal{M}$ representing the email traffic network captures the observed traffic exchanged between the 7864 Enron email addresses in $\mathcal{E}$. Since the Enron email collection is the union of email folders corresponding to the given 151 Enron email addresses, $\mathcal{G}_\mathcal{M}$ only captures the traffic exchanged between those 151 email addresses and the remaining 7713 email addresses in $\mathcal{E}$. There are 64449 emails in the ISI database that were exchanged among the 7864 email addresses. After deduplication of this set, 55395 emails remain. Therefore $\mathcal{G}_\mathcal{M}$ is composed of 7864 nodes and 55395 hyperedges.

5.1.4 Detecting Name References To detect name references in the email bodies, we initially scan

through the emails searching for words that match exactly one or both first and last names of an employee on the list of 7864 Enron employee names. We also merge adjacent partial name matches, assuming in most cases this results in a full name not listed on the employee list.

For our initial experiments, we chose to focus on resolving first name references to others outside of the email conversation. Therefore to filter out name references not of interest, we saved only partial name detections that matched one of the 151 Enron employee first names. Then we filtered out first name references at the beginning and end of the email text, assuming those are references to the sender and recipients.

5.2 Ground Truth Generation To evaluate algorithm performance, we manually identified the true network references associated with a set of first name references. In some cases, the true referent was obvious from other name references in the sender's message or the attached message. In others, we needed to search through the traffic to find other emails in the thread or previous conversations to clarify the reference. When multiple email addresses appear to correspond to the referenced individual, the email address in use around the time of the name reference is chosen as the true referent.

After this processing, we have 84 labelled first name references with candidate sets of size 2 or greater. Of these, 54 have candidate sets that contain the true referent. A number of first name references with no obvious context in the message could not be resolved after further searching of the email collection.

5.3 Performance Evaluation When evaluating the performance of a given scoring function, we have two objectives. First, we want to understand how well the scoring function ranks the true referent relative to other candidates on average in a candidate set. We refer to this as the *relative ranking performance* of the scoring function. Second, we want to characterize the ability of the scoring function to rank true referents higher than other candidates in general across candidate sets. We refer to this as the *absolute ranking performance* of the scoring function. We consider each evaluation task in the following sections.

5.3.1 Relative Ranking Performance To provide insights into relative ranking performance, three performance metrics are evaluated for each scoring function. First, we compute the *rank 1 rate (R1R)* which is the fraction of candidate sets containing true referents over which the true referent is the top ranked candidate. This is expressed as

$$R1R = \frac{1}{|\mathcal{N}_t|} \sum_{n \in \mathcal{N}_t} \mathcal{I}\left(\tilde{e}(n) = e_{true}(n)\right)$$

where $\mathcal{I}(\cdot)$ is the indicator function, $e_{true}(n)$ is the true network reference associated with the name reference n and $\mathcal{N}_t = \{n : n \in \mathcal{N}, e_{true}(n) \in \mathcal{E}(n)\}$ is the set of name references with the true referent in the corresponding candidate sets. Note the R1R is computed assuming no candidate rejection.

The rank 1 rate provides an intuitive summary of performance, but can be misleading in this context given the variable sized candidate sets. Therefore to establish a relative baseline, we compute the expected value of the *random rank 1 rate (RR1R)* achieved by random selection of the top ranked candidate from each candidate set. This is expressed as

$$RR1R = \frac{1}{|\mathcal{N}_t|} \sum_{n \in \mathcal{N}_t} \frac{1}{|\mathcal{E}(n)|}.$$

Since the rank 1 rate gives no indication of how severe the failure is when the true referent is not rank 1, we also compute a metric we refer to as the *average true referent rank (ATRR)*. The ATRR is the average of the ratio of the true referent rank and the candidate set size. This is expressed as

$$ATRR = \frac{1}{|\mathcal{N}_t|} \sum_{n \in \mathcal{N}_t} \frac{1}{|\mathcal{E}(n)|} \sum_{k=1}^{|\mathcal{E}(n)|} k\mathcal{I}\left(e_{(k)}(n) = e_{true}(n)\right)$$

where $e_{(k)}(n)$ is the network reference with rank k in the candidate set $\mathcal{E}(n)$. Each true referent rank is normalized by the corresponding candidate set size to account for the variation in the number of candidates and reduce the sensitivity of the measure to large candidate sets.

5.3.2 Absolute Ranking Performance Assessing the absolute ranking performance involves evaluating the scoring function's ability to rank true referents higher than other candidates across all candidates nominated for a given set of name references. Our interest in characterizing ranking performance from this perspective stems from our desire to reject as many candidates as possible without a significant loss of true referents. If the scoring function is able to separate the two classes of candidates with reasonable success, we will achieve our aim.

A natural measure of ranking performance advocated in the literature [2, 5, 6, 8] is the *area under the receiver operating characteristic (ROC) curve*. The

ROC curve is a standard depiction of a detector's performance from classical signal detection theory, showing the detector's true positive rate versus false positive rate [22]. The area under the ROC curve (AUC) provides a measure of the separability achieved by the detector between the two classes. More specifically, the empirical AUC is an estimate of the probability that the detector will rank a randomly selected positive example higher than a randomly selected negative example, assuming all ties are broken uniformly at random [2]. When the AUC=1.0, perfect separability is achieved. When the AUC=0.5, the detector performs no better than random chance.

If one defines the true referents to be the positive class and the other candidates to be the negative class, the AUC of the scoring function is the area under the empirical ROC curve generated by sweeping the threshold over the range of scores and computing the (false positive rate,true positive rate) operating points on the curve. This empirical AUC can be directly expressed in the following manner

$$AUC = \frac{1}{N_{TR}N_{OC}} \sum_{n_1 \in \mathcal{N}_t} \sum_{n_2 \in \mathcal{N}} \sum_{e_{oc} \in \mathcal{E}(n_2)/e_{true}(n_2)} \mathcal{I}\left(g(e_{true}(n_1), n_1|C_1) > g(e_{oc}, n_2|C_2)\right) + \frac{1}{2}\mathcal{I}\left(g(e_{true}(n_1), n_1|C_1) = g(e_{oc}, n_2|C_2)\right)$$

where $N_{TR} = |\mathcal{N}_t|$ is the number of true referents and $N_{OC} = \sum_{n \in \mathcal{N}} |\mathcal{E}(n)/e_{true}(n)|$ is the number of other candidates overall [2].

6 Discussion

We now examine the performance of the various scoring functions on the labelled name reference data. Figures 2-4 present a series of summary plots showing the rank 1 rates, average true referent ranks and AUCs of the scoring functions as a function of the amount of traffic history considered.

Consider first the R1R and ATRR metrics measuring relative ranking performance. For all models, as the filter duration is increased [2], incorporating more traffic history into the scoring process, the relative ranking performance generally increases and approaches a maximal level of performance. In terms of rank 1 rate, the performance of these simple models approaches 0.8 in most cases with sufficient history and significantly outperforms the random selection baseline. Finer level distinctions among the models can not yet be made; if one assumes the name reference resolutions are independent, there is no statistically significant difference in performance among the models considered.

At the beginning of this investigation, our expectation was that the traffic patterns around the time of a given name reference would clarify the identity of the true referent. For the models we have proposed, the results suggest quite the opposite: ranking performance is maximized when we consider the long-term traffic patterns over one year or longer. This implies that by simply examining prior relationship strengths, as represented by the volume of communication, we are able to successfully resolve a majority of first name references.

In hindsight, this result is not very surprising. While the use of a name reference in a given email may be related to an ongoing event, it is not clear that we should also expect a notable increase in communication between the true referent and the email sender around that time. For example, it is possible for a single email from the true referent to the email sender to be the impetus for the email containing the name reference. Meanwhile, the email sender may be engaged in longer threads of conversation with other potential candidates, thereby leading to a low rank for the true referent. In these scenarios, examining the content will be critical to correctly resolve the reference.

Now let us consider the AUC metric measuring absolute ranking performance. In contrast to the relative ranking results, we see a significant distinction between the sender-only models and the sender+recipients models. As the influence of the relationships between the recipients and the candidate is increased, the AUC curve continues to shift lower indicating that separability between the true referents and the other candidates is decreasing across all filter durations.

At first glance, it may seem that the trends for the relative and absolute ranking performance measures are inconsistent. Why should the relative ranking performance be fairly insensitive to the influence of the recipients while the absolute ranking performance is much more so? This result suggests that while the relative rankings are not changing significantly, the variances of the true referent and other candidate score distributions are increasing, causing the decrease in separability. This is not surprising for a number of reasons. As we add more pairwise relationships, we introduce more opportunities for the candidate's score to be falsely inflated. Multiple active pairwise relationships do not necessarily signify higher order dependencies between the relationships. Another factor is that not all email users are equivalent. Some users are more prolific email composers than others, leading

[2] The duration of the MA filter is simply the number of time intervals over which the filter averages the summary statistic. We have defined the duration of the AR filter to be twice the number of time intervals required for the impulse response of the filter to decay to 10% of its peak response.

to score inflation once again that is misleading. Further investigation is needed to explore these issues.

Clearly we have only begun to explore the full scope of the name reference resolution problem. In this initial experiment, we have focused on first name references and limited our search for the true referent to those candidates that the email sender has communicated with. Even with first name references, it is possible that a given reference corresponds to an individual that the email sender never communicates with. As discussed earlier, this can occur for a variety of reasons. Most interesting are the cases where the references are to individuals that the email sender knows of but does not have a relationship with. To deal with these challenges, a more sophisticated approach is required. Another important aspect of the resolution task to note is the dependence among name references within a given email and across emails. Our belief about one name reference can certainly impact and inform the resolution of other related references. Therefore it is valuable to incorporate such connections into the resolution process.

To summarize, we see that simple temporal traffic models perform surprisingly well for name reference resolution when considering the long-term traffic patterns. To push the performance beyond a certain level, we expect that both content and traffic patterns will need to be exploited. We are in the process of generating more labelled name references to support experimentation aimed at better understanding the limits of traffic-based approaches.

7 Related Work

This paper uses a social network generated from the email traffic of the Enron data set as a tool for name reference resolution. In this section, we describe some of the relevant related work on social networks, the Enron data set and entity resolution.

7.1 Social Networks There has been a great deal of recent work in social network generation, analysis and mining. Using semantic associations from email communication, for example, McArthur and Bruza [17] propose methods of generating a social network using implicit and explicit connections between people. Liben-Nowell and Kleinberg [15] use co-authorship to create social networks to predict future interactions among members of a given social network. Studies have also been done on creating and mining social networks to identify possible collaborators for a given problem [18, 11] and clustering people of similar interests [19]. Schwartz and Wood generate a social network using the *to* and *from* fields of email messages to discover users of a particular interest and field.

7.2 Enron The release of the Enron data set in 2003 provided an unprecedented collection of emails from a major organization for use in research. Klimt and Yang [14] provides an overview of this corpus including the number of employees, the number of emails and a representative social network derived from the email traffic. Moreover, they used the Enron data to explore methods of email classification [13]. Corrada-Emmanuel [4] created MD5 hashes of the Enron emails and contact information to identify and deduplicate emails. Using the structure of the emails, Keila and Skillicorn [12] found a relationship in the word use pattern with message length as well as relationships among individuals. Skillicorn [21] further demonstrated methods to detect unusual and deceptive email communications. Diesner and Carley [7] used analysis of the email social network patterns over time to explore crisis detection in email. Moreover, a number of useful tools have also been developed in order to navigate and view email archives [9].

7.3 Entity Resolution in Email There has been limited work in named entity resolution in email systems. Abadi [1] uses emails from an online retailer for anaphora resolution within email orders. Abadi's research, however, is designed for the resolution of pronouns referring to product orders rather than individuals and relies mainly on NLP for resolution. Holzer, Malin and Sweeney [10] on the other hand use social networks created from online resources like personal websites to resolve email aliases. Their approach of using social networks derived from relations from other sources, including proximity of references in a given web site, is particularly effective in controlled environments such as the university used in their evaluation. Of note, is Malin's evaluation of methods of disambiguation in relational environments [16]. Although Malin's work used actor collaborations in the Internet Movie Database rather than email, Malin did find that methods which leverage community, in contrast to exact similarity provide more robust disambiguation capability, supporting our approach to the problem.

8 Conclusion

In this paper, we have examined ways in which email traffic can be used to resolve ambiguous name references within the body of the email messages. Our contributions include a formal statement of the problem; the definition of the resolution process in terms of candidate generation, candidate scoring and candidate rejection; and an evaluation methodology that examines both absolute and relative rank. We have presented an initial suite of models for candidate scoring, which exploit both role and temporal information, and evalu-

ated their performance on a real-world corporate email archive, the Enron collection. Surprisingly, such models perform well by simply ranking candidates based on long-term traffic statistics, a natural surrogate for relationship strength. Our overall goal is to develop robust ways of exploiting context information during the resolution process. The email traffic network is just one element of the context information we hope to eventually exploit.

References

[1] D. Abadi. Comparing domain-specific and non-domain-specific anaphora resolution techniques. Master's thesis, Cambridge University MPhil Dissertation, 2003.

[2] S. Agarwal, T. Graepel, R. Herbrich, and D. Roth. Generalization bounds for the area under the ROC curve. *Journal of Machine Learning Research*, 6:393–425, 2005.

[3] J. Allan. *Topic Detection and Tracking.* Kluwer Academic Pub, 2002.

[4] A. Corrada-Emmanuel. Enron email dataset research, 2004. http://ciir.cs.umass.edu/~corrada/enron.

[5] C. Cortes and M. Mohri. AUC optimization versus error rate minimization. *Advances in Neural Information Processing Systems*, 16, 2004.

[6] C. Cortes and M. Mohri. Confidence intervals for the area under the ROC curve. *Advances in Neural Information Processing Systems*, 17, 2005.

[7] J. Diesner and K. Carley. Exploration of communication networks from the Enron email corpus. In *Proc. of Workshop on Link Analysis, Counterterrorism and Security, SIAM International Conference on Data Mining 2005*, Newport Beach, CA, USA, April 21-23 2005.

[8] Y. Freund, R. Iyer, R. Shapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 4:933–969, 2003.

[9] J. Heer. Exploring Enron: Visualizing ANLP results, 2004. http://jheer.org/enron/v1/.

[10] R. Holzer, B. Malin, and L. Sweeney. Email alias detection using social network analysis. In *Proceedings of the ACM SIGKDD Workshop on Link Discovery: Issues, Approaches, and Applications*, Chicago, Illinois, USA, August 2005.

[11] H. Kautz, B. Selman, and M. Shah. Referral web: Combining social networks and collaborative filtering. *Communications of the ACM*, 40(3):63–65, 1997.

[12] P. Keila and D. Skillicorn. Structure in the Enron email dataset. *In Workshop on Link Analysis, Security and Counterterrorism, SIAM International Conference on Data Mining*, pages 55–64, 2005.

[13] B. Klimt and Y. Yang. The Enron corpus: a new dataset for email classification research. In *European Conference on Machine Learning/PKDD*, Pisa, Italy, September 20-24 2004.

[14] B. Klimt and Y. Yang. Introducing the Enron corpus. In *Conference on Email and Anti-Spam*, Mountainview, CA, USA, July 30-31 2004.

[15] D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks. In *Proc. 12th International Conference on Information and Knowledge Management (CIKM)*, New Orleans, Louisiana, USA, November 2-8 2003.

[16] B. Malin. Unsupervised name disambiguation via social network similarity. In *SIAM International Conference on Data Mining*, Newport Beach, CA, USA, April 20-22 2005.

[17] R. McArthur and P. Bruza. Discovery of implicit and explicit connections between people using email utterance. In *Proceedings of the Eighth European Conference of Computer-supported Cooperative Work, Helsinki*, Helsinki, Finland, September 14-17 2003.

[18] H. Ogata and Y. Yano. Collecting organizational memory based on social networks in collaborative learning. In *WebNet*, pages 822–827, 1999.

[19] M. Schwartz and D. Wood. Discovering shared interests among people using graph analysis of global electronic mail traffic. *Communications of the ACM*, 36:78–89, 1992.

[20] J. Shetty and J. Adibi. The Enron email dataset: Database schema and brief statistical report. http://www.isi.edu/~adibi/Enron/Enron_Dataset_Report.pdf.

[21] D. Skillicorn. Detecting unusual and deceptive communication in email. In *Centers for Advanced Studies Conference*, Richmond Hill, Ontario, Canada, October 17-20 2005.

[22] H. V. Trees. *Detection, Estimation, and Modulation Theory.* John Wiley and Sons, 1968.

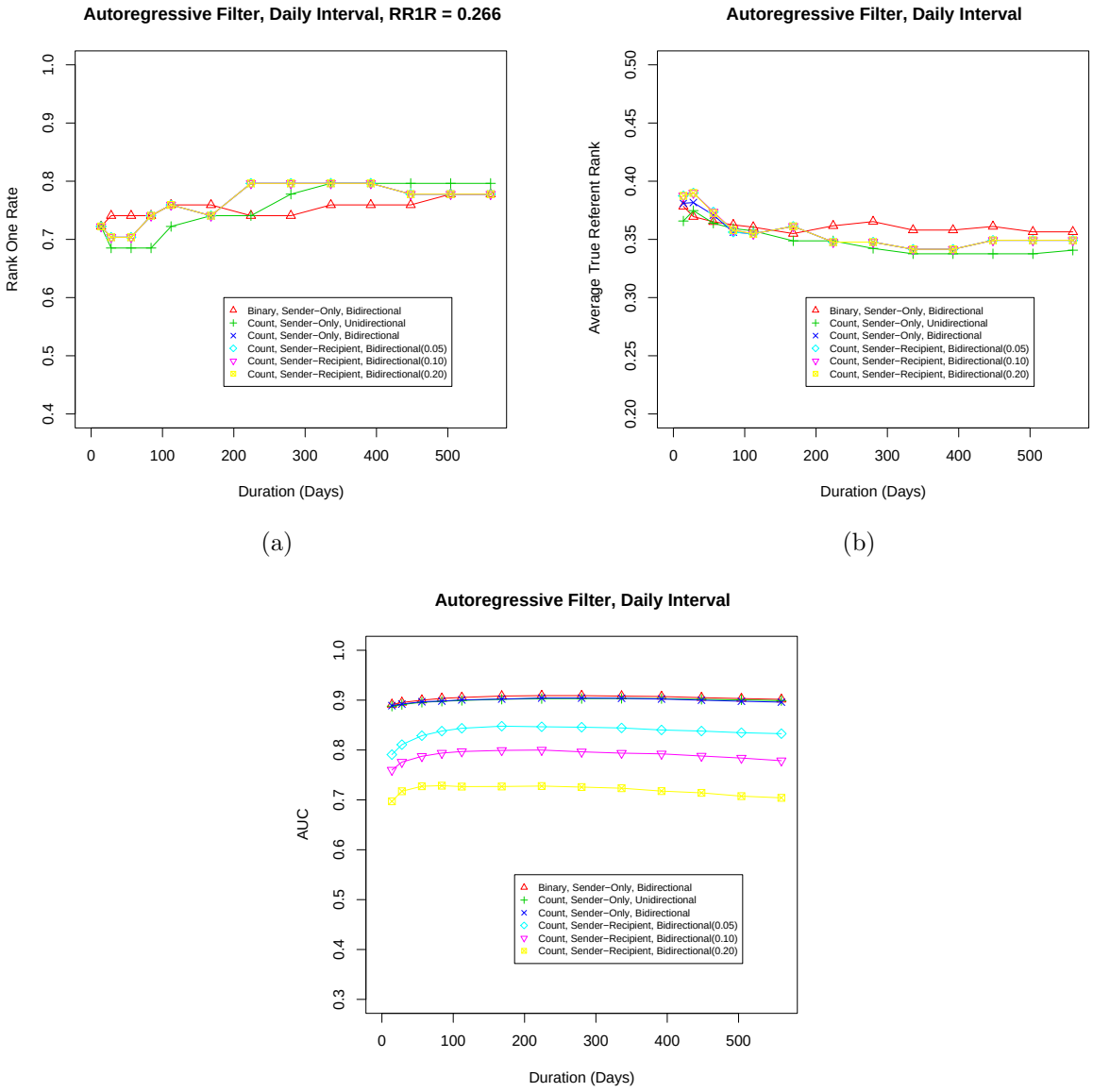

Figure 2: Autoregressive Filter Performance: (a) Daily Interval Rank 1 Rates, (b) Average True Referent Ranks and (c) Areas Under the ROC Curves

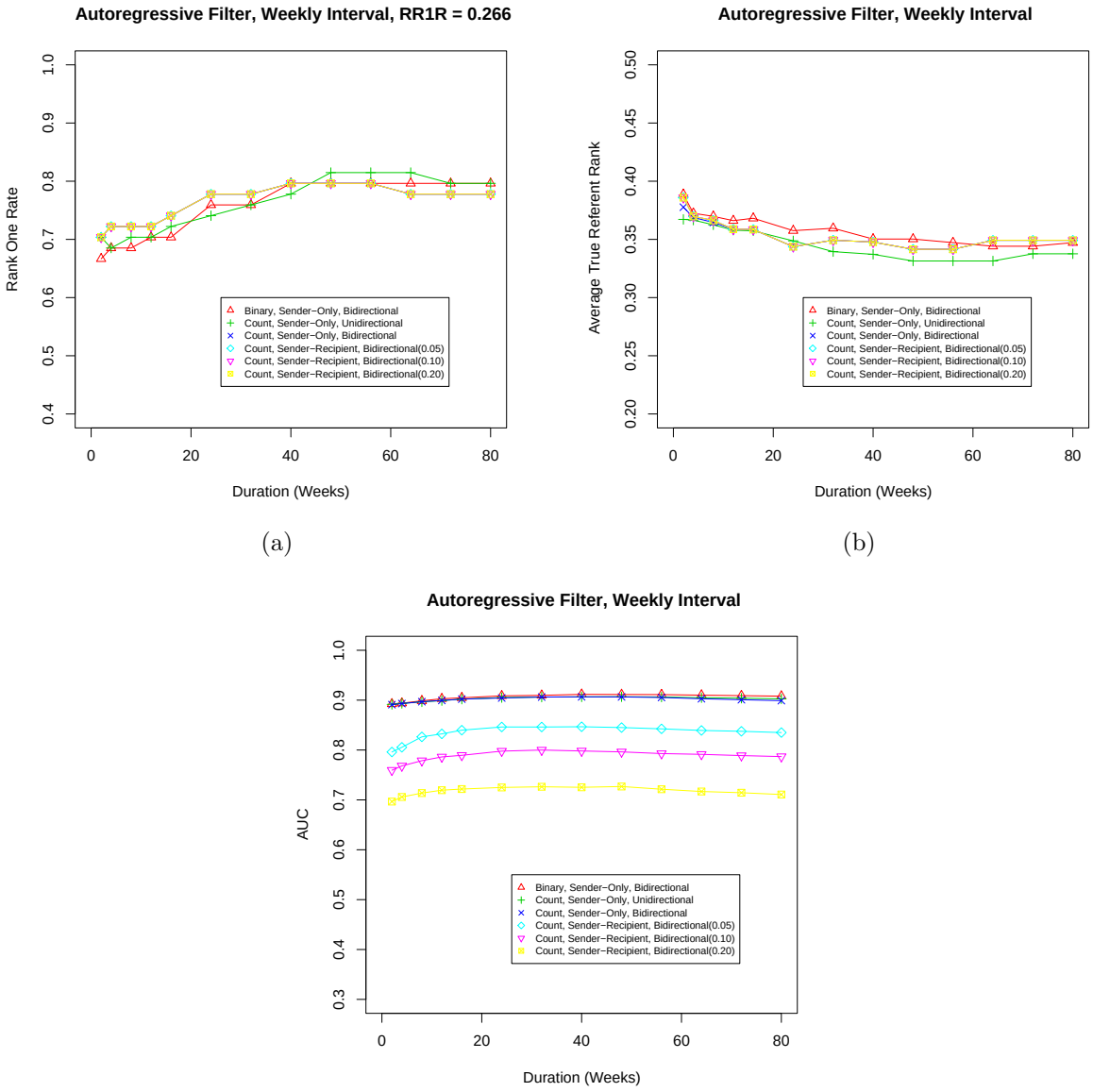

Figure 3: Autoregressive Filter Performance: (a) Weekly Interval Rank 1 Rates, (b) Average True Referent Ranks and (c) Areas Under the ROC Curves

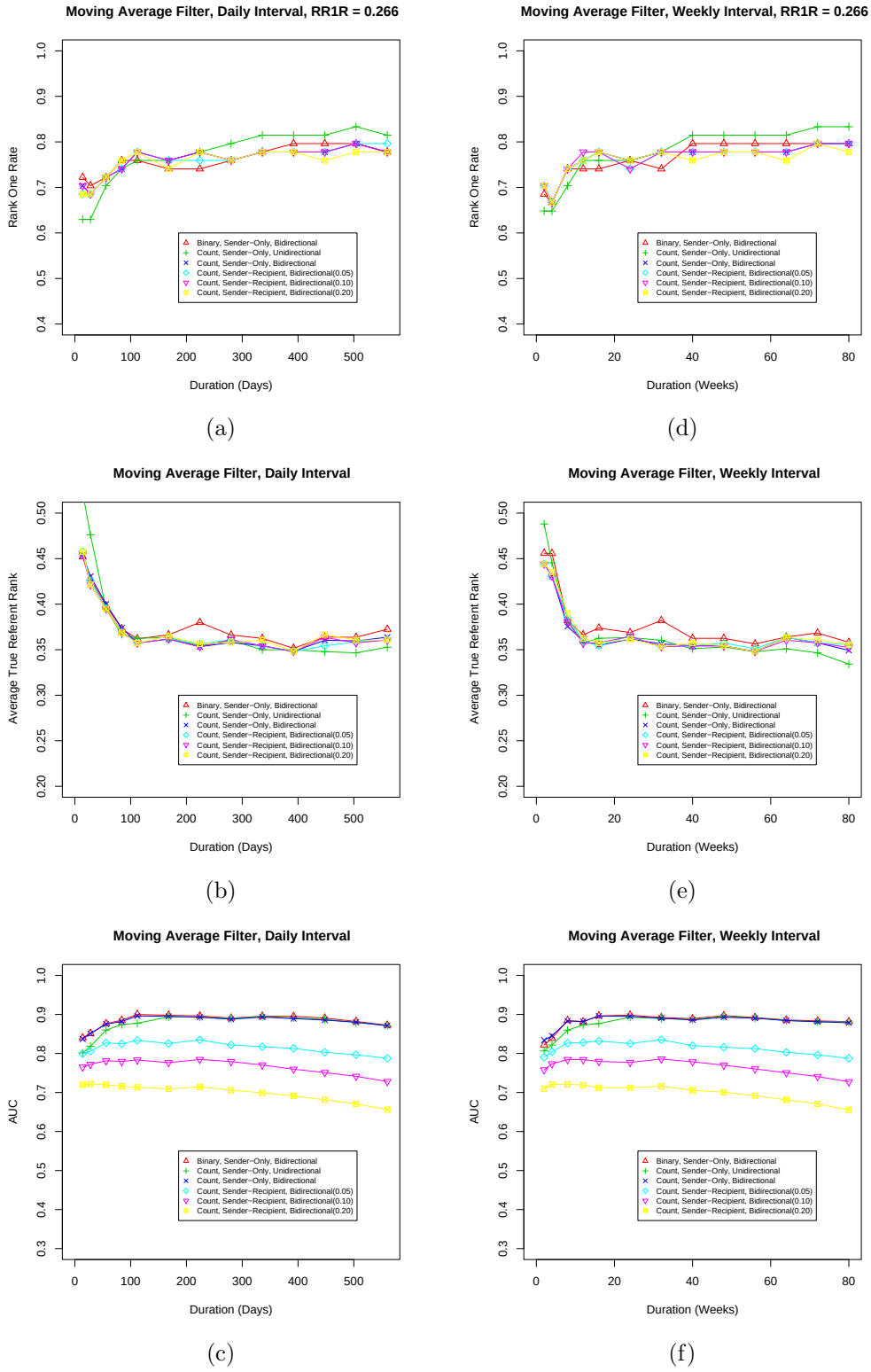

Figure 4: Moving Average Filter Performance: (a) Daily Interval Rank 1 Rates, (b) Average True Referent Ranks and (c) Areas Under the ROC Curves (d) Weekly Interval Rank 1 Rates, (e) Average True Referent Ranks and (f) Areas Under the ROC Curves