Resolving Personal Names in Email Using Context Expansion

Tamer Elsayed,* Douglas W. Oard,† and Galileo Namata*
Human Language Technology Center of Excellence and
UMIACS Laboratory for Computational Linguistics and Information Processing (CLIP)
University of Maryland, College Park, MD 20742
{telsayed, oard, gnamata}@umd.edu

Abstract

This paper describes a computational approach to resolving the true referent of a named mention of a person in the body of an email. A generative model of mention generation is used to guide mention resolution. Results on three relatively small collections indicate that the accuracy of this approach compares favorably to the best known techniques, and results on the full CMU Enron collection indicate that it scales well to larger collections.

1 Introduction

The increasing prevalence of informal text from which a dialog structure can be reconstructed (e.g., email or instant messaging), raises new challenges if we are to help users make sense of this cacophony. Large collections offer greater scope for assembling evidence to help with that task, but they pose additional challenges as well. With well over 100,000 unique email addresses in the CMU version of the Enron collection (Klimt and Yang, 2004), common names (e.g., John) might easily refer to any one of several hundred people. In this paper, we associate named mentions in unstructured text (i.e., the body of an email and/or the subject line) to modeled identities. We see at least two direct applications for this work: (1) helping searchers who are unfamiliar with the contents of an email collection (e.g., historians or lawyers) better understand the context of emails that they find, and (2) augmenting more typical social networks (based on senders and recipients) with additional links based on references found in unstructured text.

Most approaches to resolving identity can be decomposed into four sub-problems: (1) finding a reference that requires resolution, (2) identifying candidates, (3) assembling evidence, and (4) choosing among the candidates based on the evidence. For the work reported in this paper, we rely on the user to designate references requiring resolution (which we model as a predetermined set of mention-queries for which the correct referent is known). Candidate identification is a computational expedient that permits the evidence assembly effort to be efficiently focused; we use only simple techniques for that task. Our principal contributions are the approaches we take to evidence generation (leveraging three ways of linking to other emails where evidence might be found: reply chains, social interaction, and topical similarity) and our approach to choosing among candidates (based on a generative model of reference production). We evaluate the effectiveness of our approach on four collections, three of which have previously reported results for comparison, and one that is considerably larger than the others.

The remainder of this paper is as follows. Section 2 surveys prior work. Section 3 then describes our approach to modeling identity and ranking candidates. Section 4 presents results, and Section 5 concludes.

2 Related Work

The problem of identity resolution in email is a special case of the more general problem referred to as "Entity Resolution." Entity resolution is generically defined as a process of determining the mapping from references (e.g., names, phrases) observed in data to real-world entities (e.g., persons, locations). In our case, the problem is to map *mentions* in emails to the *identities* of the individuals being referred to.

Various approaches have been proposed for entity resolution. In structured data (e.g., databases), approaches have included minimizing the number of "matching" and "merging" operations (Benjelloun et al., 2006), using global relational information(Malin, 2005; Bhattacharya and Getoor, 2007; Reuther, 2006) and using a probabilistic generative

*Department of Computer Science
†College of Information Studies

Proceedings of ACL-08: HLT, pages 941–949,
Columbus, Ohio, USA, June 2008.

model (Bhattacharya and Getoor, 2006). None of these approaches, however, both make use of conversational, topical, and time aspects, shown important in resolving personal names (Reuther, 2006), and take into account global relational information. Similarly, approaches in unstructured data (e.g., text) have involved using clustering techniques over biographical facts (Mann and Yarowsky, 2003), within-document resolution (Blume, 2005), and discriminative unsupervised generative models (Li et al., 2005). These too are insufficient for our problem since they suffer from inability scale or to handle early negotiation.

Specific to the problem of resolving mentions in email collections, Abadi (Abadi, 2003) used email orders from an online retailer to resolve product mentions in orders and Holzer et al. (Holzer et al., 2005) used the Web to acquire information about individuals mentioned in headers of an email collection. Our work is focused on resolving personal name references in the full email including the message body; a problem first explored by Diehl et al. (Diehl et al., 2006) using header-based traffic analysis techniques. Minkov et al.(Minkov et al., 2006) studied the same problem using a lazy graph walk based on both headers and content. Those two recent studies reported results on different test collections, however, making direct comparisons difficult. We have therefore adopted their test collections in order to establish a common point of reference.

3 Mention Resolution Approach

The problem we are interested in is the resolution of a personal-name mention (i.e., a named reference to a person) m, in a specific email e^m in the given collection of emails E, to its true referent. We assume that the user will designate such mention. This can be formulated as a *known-item* retrieval problem (Allen, 1989) since there is always only one right answer. Our goal is to develop a system that provides a list of potential candidates, ranked according to how strongly the system believes that a candidate is the true referent meant by the email author. In this paper, we propose a probabilistic approach that ranks the candidates based on the estimated probability of having been mentioned. Formally, we seek to estimate the probability $p(c|m)$ that a potential candidate c is the one referred to by the given mention m, over all candidates C.

We define a mention m as a tuple $< l^m, e^m >$, where l^m is the "literal" string of characters that represents m and e^m is the email where m is observed.[1] We assume that m can be resolved to a distinguishable participant for whom at least one email address is present in the collection.[2]

The probabilistic approach we propose is motivated by a generative scenario of mentioning people in email. The scenario begins with the author of the email e^m, intending to refer to a person in that email. To do that s/he will:

1. Select a person c to whom s/he will refer
2. Select an appropriate context x_k to mention c
3. Select a specific lexical reference l^m to refer to c given the context x_k.

For example, suppose "John" is sending an email to "Steve" and wants to mention a common friend "Edward." "John" knows that he and Steve know 2 people named Edward, one is a friend of both known by "Ed" and the other is his soccer trainer. If "John" would like to talk about the former, he would use "Ed" but he would likely use "Edward" plus some terms (e.g., "soccer", "team", etc) for the latter. "John" relies on the social context, or the topical context, for "Steve" to disambiguate the mention.

The steps of this scenario impose a certain structure to our solution. First, we need to have a representational model for each candidate identity. Second, we need to reconstruct the context of the queried mention. Third, it requires a computational model of identity that supports reasoning about identities. Finally, it requires a resolution technique that leverages both the identity models and the context to rank the potential candidates. In this section, we will present our resolution approach within that structure. We first discuss how to build both representational and computational models of identity in section 3.1. Next, we introduce a definition of the contextual space and how we can reconstruct it in

[1] The exact position in e^m where l^m is observed should also be included in the definition, but we ignore it assuming that all matched literal mentions in one email refer to the same identity.

[2] Resolving mentions that refer to non-participants is outside the scope of this paper.

section 3.2. Finally, we link those pieces together by the resolution algorithm in section 3.3.

3.1 Computational Model of Identity

Representation: In a collection of emails, individuals often use different email addresses, multiple forms of their proper names, and different nicknames. In order to track references to a person over a large collection, we need to capture as many as possible of these *referential* attributes in one representation. We extend our simple representation of identity proposed in (Elsayed and Oard, 2006) where an identity is represented by a set of pairwise co-occurrence of referential attributes (i.e., co-occurrence "associations"), and each extracted association has a frequency of occurrence. The attributes are extracted from the headers and salutation and signature lines. For example, an "address-nickname" association $< a, n >$ is inferred whenever a nickname n is usually observed in signature lines of emails sent from email address a. Three types of referential attributes were identified in the original representation: email addresses, names, and nicknames. We add usernames as well to account for the absence of any other type of names. Names, nicknames, and usernames are distinguishable based on where each is extracted: email addresses and names from headers, nicknames from salutation and signature lines, and usernames from email addresses. Since (except in rare cases) an email address is bound to one personal identity, the model leverages email addresses as the basis by mandating that at least one email address must appear in any observed association. As an off-line preprocessing step, we extract the referential attributes from the whole collection and build the identity models. The first step in the resolution process is to determine the list of identity models that are viable candidates as the true referent. For the experiments reported in this paper, any identity model with a first name or nickname that exactly matches the mention is considered a candidate.

Labeling Observed Names: For the purpose of resolving name mentions, it is necessary to compute the probability $p(l|c)$ that a person c is referred to by a given "literal" mention l. Intuitively, that probability can be estimated based on the observed "name-type" of l and how often that association occurs in the represented model. We define T as the set of 3 different types of single-token name-types: first, last, and nickname. We did not handle middle names and initials, just for simplicity. Names that are extracted from salutation and signature lines are labeled as nicknames whereas full names extracted from headers are first normalized to "First Last" form and then each single token is labeled based on its relative position as being the first or last name. Usernames are treated similarly to full names if they have more than one token, otherwise they are ignored. Note that the same single-token name may appear as a first name and a nickname.

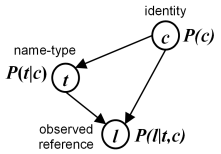

Figure 1: A computational model of identity.

Reasoning: Having tokenized and labeled all names, we propose to model the association of a single-token name l of type t to an identity c by a simple 3-node Bayesian network illustrated in Figure 1. In the network, the observed mention l is distributed conditionally on both the identity c and the name-type t. $p(c)$ is the prior probability of observing the identity c in the collection. $p(t|c)$ is the probability that a name-type t is used to refer to c. $p(l|t,c)$ is the probability of referring to c by l of type t. These probabilities can be inferred from the representational model as follows:

$$p(c) = \frac{|assoc(c)|}{\sum_{c' \in C} |assoc(c')|}$$

$$p(t|c) = \frac{freq(t,c)}{\sum_{t' \in T} freq(t',c)}$$

$$p(l|t,c) = \frac{freq(l,t,c)}{\sum_{l' \in assoc(c)} freq(l',t,c)}$$

where $assoc(c)$ is the set of observed associations of referential attributes in the represented model c.

The probability of observing a mention l given that it belongs to an identity c, without assuming a specific token type, can then be inferred as follows:

$$p(l|c) = \sum_{t \in T} p(t|c)\ p(l|t,c)$$

In the case of a multi-token names (e.g., John Smith), we assume that the first is either a first name

or nickname and the last is a last name, and compute it accordingly as follows:

$$p(l_1 l_2|c) = \{ \sum_{t \in \{f,n\}} p(t|c)\, p(l_1|t,c)\} \cdot p(l_2|last,c)$$

where f and n above denotes first name and nickname respectively.

Email addresses are also handled, but in a different way. Since we assume each of them uniquely identifies the identity, all email addresses for one identity are mapped to just one of them, which then has half of the probability mass (because it appears in every extracted co-occurrence association).

Our computational model of identity can be thought of as a language model over a set of personal references and thus it is important to account for unobserved references. If we know that a specific first name often has a common nickname (by a dictionary of commonly used first to nickname mappings (e.g., Robert to Bob)), but this nickname was not observed in the corpus, we will need to apply smoothing. We achieve that by assuming the nickname would have been observed n times where n is some fraction (0.75 in our experiments) of the frequency of the observed name. We repeat that for each unobserved nickname and then treat them as if they were actually observed.

3.2 Contextual Space

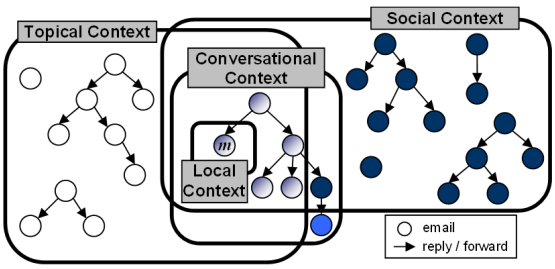

Figure 2: Contextual Space

It is obvious that understanding the context of an ambiguous mention will help with resolving it.

Fortunately, the nature of email as a conversational medium and the link-relationships between emails and people over time can reveal clues that can be exploited to partially reconstruct that context.

We define the contextual space $X(m)$ of a mention m as a *mixture* of 4 types of contexts with λ_k as the mixing coefficient of context x_k. The four contexts (illustrated in Figure 2) are:

(1) *Local Context*: the email e^m where the named person is mentioned.
(2) *Conversational Context*: emails in the broader discussion that includes e^m, typically the thread that contains it.
(3) *Social Context*: discussions that some or all of the participants (sender and receivers) of e^m joined or initiated at around the time of the mention-email. These might bear some otherwise-undetected relationship to the mention-email.
(4) *Topical Context*: discussions that are topically similar to the mention-discussion that took place at around the time of e^m, regardless of whether the discussions share any common participants.

These generally represent a growing (although not strictly nested) contextual space around the queried mention. We assume that all mentions in an email share the same contextual space. Therefore, we can treat the context of a mention as the context of its email. However, each email in the collection has its own contextual space that could overlap with another email's space.

3.2.1 Formal Definition

We define K as the set of the 4 types of contexts. A context x_k is represented by a probability distribution over all emails in the collection. An email e_j belongs to the k^{th} context of another email e_i with probability $p(e_j|x_k(e_i))$. How we actually represent each context and estimate the distribution depends upon the type of the context. We explain that in detail in section 3.2.2.

3.2.2 Context Reconstruction

In this section, we describe how each context is constructed.

Local Context: Since this is simply e^m, all of the probability mass is assigned to it.

Conversational Context: Threads (i.e., reply chains) are imperfect approximations of focused discussions, since people sometimes switch topics within a thread (and indeed sometimes within the same email). We nonetheless expect threads to exhibit a useful degree of focus and we have therefore adopted them as a computational representation of a discussion in our experiments. To reconstruct threads in the collection, we adopted the technique introduced in (Lewis and Knowles, 1997). Thread

reconstruction results in a unique tree containing the mention-email. Although we can distinguish between different paths or subtrees of that tree, we elected to have a uniform distribution over all emails in the same thread. This also applies to threads retrieved in the social and topical contexts as well.

Social Context: Discussions that share common participants may also be useful, though we expect their utility to decay somewhat with time. To reconstruct that context, we temporally rank emails that share at least one participant with e^m in a time period around e^m and then expand each by its thread (with duplicate removal). Emails in each thread are then each assigned a weight that equals the reciprocal of its thread rank. We do that separately for emails that temporally precede or follow e^m. Finally, weights are normalized to produce one distribution for the whole social context.

Topical Context: Identifying topically-similar content is a traditional query-by-example problem that has been well researched in, for example, the TREC routing task (Lewis, 1996) and the Topic Detection and Tracking evaluations (Allan, 2002). Individual emails may be quite terse, but we can exploit the conversational structure to obtain topically related text. In our experiments, we tracked back to the root of the thread in which e^m was found and used the subject line and the body text of that root email as a query to Lucene[3] to identify topically-similar emails. Terms found in the subject line are doubled in the query to emphasize what is sometimes a concise description of the original topic. Subsequent processing is then similar to that used for the social context, except that the emails are first ranked by their topical, rather than temporal, similarity.

The approaches we adopted to reconstruct the social and topical contexts were chosen for their relative simplicity, but there are clearly more sophisticated alternatives. For example, topic modeling techniques (McCallum et al., 2005) could be leveraged in the reconstruction of the topical context.

3.3 Mention Resolution

Given a specific mention m and the set of identity models C, our goal now is to compute $p(c|m)$ for each candidate c and rank them accordingly.

[3] http://lucene.apache.org

3.3.1 Context-Free Mention Resolution

If we resolve m out of its context, then we can compute $p(c|m)$ by applying Bayes' rule as follows:

$$p(c|m) \approx p(c|l^m) = \frac{p(l^m|c)\ p(c)}{\sum_{c' \in C} p(l^m|c')\ p(c')}$$

All the terms above are estimated as discussed earlier in section 3.1. We call this approach "backoff" since it can be used as a fall-back strategy. It is considered the baseline approach in our experiments.

3.3.2 Contextual Mention Resolution

We now discuss the more realistic situation in which we use the context to resolve m. By expanding the mention with its context, we get

$$p(c|m) = p(c|l^m, X(e^m))$$

We then apply Bayes' rule to get

$$p(c|l^m, X(e^m)) = \frac{p(c, l^m, X(e^m))}{p(l^m, X(e^m))}$$

where $p(l^m, X(e^m))$ is the probability of observing l^m in the context. We can ignore this probability since it is constant across all candidates in our ranking. We now restrict our focus to the numerator $p(c, l^m, X(e^m))$, that is the probability that the sender chose to refer to c by l^m in the contextual space. As we discussed in section 3.2, X is defined as a mixture of contexts therefore we can further expand it as follows:

$$p(c, l^m, X(e^m)) = \sum_k \lambda_k\ p(c, l^m, x_k(e^m))$$

Following the intuitive generative scenario we introduced earlier, the context-specific probability can be decomposed as follows:

$$\begin{aligned} p(c, l^m, x_k(e^m)) = & p(c) \\ & * p(x_k(e^m)|c) \\ & * p(l^m|x_k(e^m), c) \end{aligned}$$

where $p(c)$ is the probability of selecting a candidate c, $p(x_k(e^m)|c)$ is the probability of selecting x_k as an appropriate context to mention c, and $p(l^m|x_k(e^m), c)$ is the probability of choosing to mention c by l^m given that x_k is the appropriate context.

***Choosing person to mention*:** $p(c)$ can be estimated as discussed in section 3.1.

***Choosing appropriate context*:** By applying Bayes' rule to compute $p(x_k(e^m)|c)$ we get

$$p(x_k(e^m)|c) = \frac{p(c|x_k(e^m))\ p(x_k(e^m))}{p(c)}$$

$p(x_k(e^m))$ is the probability of choosing x_k to generally mention people. In our experiments, we assumed a uniform distribution over all contexts. $p(c|x_k(e^m))$ is the probability of mentioning c in $x_k(e^m)$. Given that the context is defined as a distribution over emails, this can be expanded to

$$p(c|x_k(e^m)) = \sum_{e_i \in E} p(e_i|x_k(e^m)\ p(c|e_i))$$

where $p(c|e_i)$ is the probability that c is mentioned in the email e_i. This, in turn, can be estimated using the probability of referring to c by at least one unique reference observed in that email. By assuming that all lexical matches in the same email refer to the same person, and that all lexically-unique references are statistically independent, we can compute that probability as follows:

$$\begin{aligned} p(c|e_i) &= 1 - p(c \text{ is not mentioned in } e_i) \\ &= 1 - \prod_{m' \in M(e_i)} (1 - p(c|m')) \end{aligned}$$

where $p(c|m^{'})$ is the probability that c is the true referent of $m^{'}$. This is the same general problem of resolving mentions, but now concerning a related mention $m^{'}$ found in the context of m. To handle this, there are two alternative solutions: (1) break the cycle and compute context-free resolution probabilities for those related mentions, or (2) jointly resolve all mentions. In this paper, we will only consider the first, leaving joint resolution for future work.

***Choosing a name-mention*:** To estimate $p(l^m|x_k(e^m), c)$, we suggest that the email author would choose either to select a reference (or a modified version of a reference) that was previously mentioned in the context or just ignore the context. Hence, we estimate that probability as follows:

$$\begin{aligned} p(l^m|x_k(e^m), c) = \alpha\ & p(l^m \in x_k(e^m)|c) \\ & +(1-\alpha)\ p(l^m|c) \end{aligned}$$

where $\alpha \in [0, 1]$ is a mixing parameter (set at 0.9 in our experiments), and $p(l^m|c)$ is estimated as in section 3.1. $p(l^m \in x_k(e^m)|c)$ can be estimated as follows:

$$p(l^m \in x_k(e^m)|c) = \sum_{m^{'} \in x_k} p(l^m|l^{m^{'}}) p(l^{m^{'}}|x_k)\ p(c|l^{m^{'}})$$

where $p(l^m|l^{m^{'}})$ is the probability of modifying $l^{m^{'}}$ into l^m. We assume all possible mentions of c are equally similar to m and estimate $p(l^m|l^{m^{'}})$ by $\frac{1}{|\text{possible mentions of } c|}$. $p(l^{m^{'}}|x_k)$ is the probability of observing $l^{m^{'}}$ in x_k, which we estimate by its relative frequency in that context. Finally, $p(c|l^{m^{'}})$ is again a mention resolution problem concerning the reference r_i which can be resolved as shown earlier.

The Aho-Corasick linear-time algorithm (Aho and Corasick, 1975) is used to find mentions of names, using a corpus-based dictionary that includes all names, nicknames, and email addresses extracted in the preprocessing step.

4 Experimental Evaluation

We evaluate our mention resolution approach using four test collections, all are based on the CMU version of the Enron collection; each was created by selecting a subset of that collection, selecting a set of query-mentions within emails from that subset, and creating an answer key in which each query-mention is associated with a single email address.

The first two test collections were created by Minkov et al (Minkov et al., 2006). These test collections correspond to two email accounts, "sager-e" (the "Sager" collection) and "shapiro-r" (the "Shapiro" collection). Their mention-queries and answer keys were generated automatically by identifying name mentions that correspond uniquely to individuals referenced in the cc header, and eliminating that cc entry from the header.

The third test collection, which we call the "Enron-subset" is an extended version of the test collection created by Diehl at al (Diehl et al., 2006). Emails from all top-level folders were included in the collection, but only those that were both sent by and received by at least one email address of the form <name1>.<name2>@enron.com were retained. A set of 78 mention-queries were manually selected and manually associated with the email address of the true referent by the third author using an interactive search system developed specifically to support that task. The set of queries was limited to those that resolve to an address of the form <name1>.<name2>@enron.com. Names found in salutation or signature lines or that exactly match <name1> or <name2> of any of the email participants were not selected as query-mentions. Those 78 queries include the 54 used by Diehl et al.

Table 1: Test collections used in the experiments.

Test Coll.	Emails	IDs	Queries	Candidates
Sager	1,628	627	51	4 (1-11)
Shapiro	974	855	49	8 (1-21)
Enron-sub	54,018	27,340	78	152 (1-489)
Enron-all	248,451	123,783	78	518 (3-1785)

For our fourth test collection ("Enron-all"), we used the same 78 mention-queries and the answer key from the Enron-subset collection, but we used the full CMU version of the Enron collection (with duplicates removed). We use this collection to assess the scalability of our techniques.

Some descriptive statistics for each test collection are shown in Table 1. The Sager and Shapiro collections are typical of personal collections, while the other two represent organizational collections. These two types of collections differ markedly in the number of known identities and the candidate list sizes as shown in the table (the candidate list size is presented as an average over that collection's mention-queries and as the full range of values).

4.1 Evaluation Measures

There are two commonly used single-valued evaluation measures for "known item"-retrieval tasks. The "*Success @ 1*" measure characterizes the accuracy of one-best selection, computed as the mean across queries of the precision at the top rank for each query. For a single-valued figure of merit that considers every list position, we use "*Mean Reciprocal Rank*" (MRR), computed as the mean across queries of the inverse of the rank at which the correct referent is found.

4.2 Results

There are four basic questions which we address in our experimental evaluation: (1) How does our approach perform compared to other approaches?, (2) How is it affected by the size of the collection and by increasing the time period?, (3) Which context makes the most important contribution to the resolution task? and (4) Does the mixture help?

In our experiments, we set the mixing coefficients λ_k and the context priors $p(x_k)$ to a uniform distribution over all reconstructed contexts.

To compare our system performance with results previously reported, we experimented with different (symmetric) time periods for selecting threads in the social and topical contexts. Three representative time periods, in days, were arbitrarily chosen: 10 (i.e., +/- 5) days, 100 (i.e., +/- 50) days, and 200 (i.e., +/- 100) days. In each case, the mention-email defines the center of this period.

Table 2: Accuracy results with different time periods.

	Period	MRR		Success @ 1	
	(days)	Prob.	Minkov	Prob.	Minkov
Sager	10	0.899	0.889	0.843	0.804
	100	**0.911**	0.889	**0.863**	0.804
	200	**0.911**	0.889	**0.863**	0.804
Shapiro	10	**0.913**	0.879	0.857	0.779
	100	0.910	0.879	0.837	0.779
	200	0.911	0.837	**0.878**	0.779
Enron-sub	10	0.878	-	0.821	-
	100	**0.911**	-	**0.846**	-
	200	**0.911**	-	**0.846**	-
Enron-all	10	**0.890**	-	**0.821**	-
	100	0.888	-	**0.821**	-
	200	0.888	-	**0.821**	-

A summary of the our results (denoted by "Prob.") are shown in Table 2 with the best results for each test collection highlighted in bold. The table also includes the results reported in Minkov et al (Minkov et al., 2006) for the small collections for comparison purposes.[4] Each score for our system was the best over all combinations of contexts for these collections and time periods. Given these scores, our results compare favorably with the previously reported results for both Sager and Shapiro collections.

Another notable thing about our results is that they seem to be good enough for practical applications. Specifically, our one-best selection (over all tried conditions) is correct at least 82% of the time over all collections, including the largest one. Of course, the Enron-focused selection of mention-queries in every case is an important caveat on these results; we do not yet know how well our techniques will hold up with less evidence, as might be the case for mentions of people from outside Enron.

It is encouraging that testing on the largest col-

[4]For the "Enron-subset" collection, we do not know which 54 mention-queries Diehl et al used in (Diehl et al., 2006)

lection (with all unrelated and thus noisy data) did not hurt the effectiveness much. For the three different time periods we tried, there was no systematic effect.

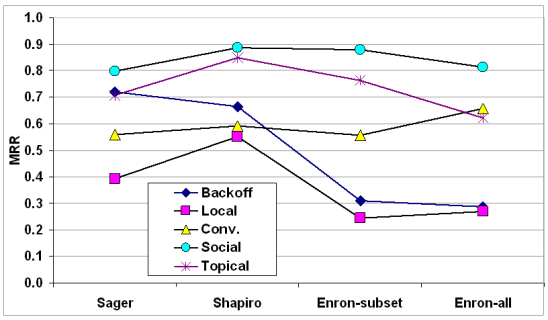

Figure 3: Individual contexts, period set to 100 days.

Individual Contexts: Our choice of contexts was motivated by intuition rather than experiments, so we also took this opportunity to characterize the contribution of each context to the results. We did that by setting some of the context mixing-coefficients to zero and leaving the others equally-weighted. Figure 3 shows the MRR achieved with each context. In that figure, the "backoff" curve indicates how well the simple context-free resolution would do. The difference between the two smallest and the two largest collections is immediately apparent–this backoff is remarkably effective for the smaller collections, and almost useless for the larger ones, suggesting that the two smaller collections are essentially much easier. The social context is clearly quite useful, more so than any other single context, for every collection. This tends to support our expectation that social networks can be as informative as content networks in email collections. The topical context also seems to be useful on its own. The conversational context is moderately useful on its own in the larger collections. The local context alone is not very informative for the larger collections.

Mixture of Contexts: The principal motivation for combining different types of contexts is that different sources may provide complementary evidence. To characterize that effect, we look at combinations of contexts. Figure 4 shows three such context combinations, anchored by the social context alone, with a 100-day window (the results for 10 and 200 day periods are similar). Reassuringly, adding more contexts (hence more evidence) turns out to be a reasonable choice in most cases. For the full combination, we notice a drop in the effectiveness from the addition of the topical context.[5] This suggests that the construction of the topical context may need more careful design, and/or that learned λ_k's could yield better evidence combination (since these results were obtained with equal λ_k's).

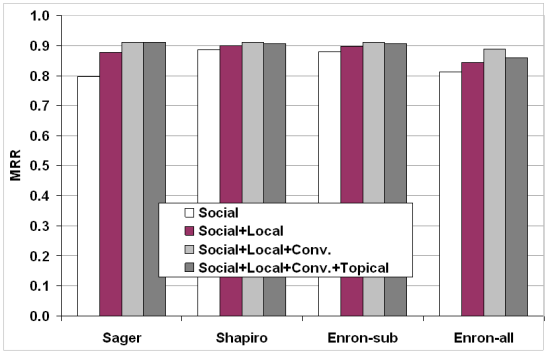

Figure 4: Mixture of contexts, period set to 100 days.

5 Conclusion

We have presented an approach to mention resolution in email that flexibly makes use of expanding contexts to accurately resolve the identity of a given mention. Our approach focuses on four naturally occurring contexts in email, including a message, a thread, other emails with senders and/or recipients in common, and other emails with significant topical content in common. Our approach outperforms previously reported techniques and it scales well to larger collections. Moreover, our results serve to highlight the importance of social context when resolving mentions in social media, which is an idea that deserves more attention generally. In future work, we plan to extend our test collection with mention queries that must be resolved in the "long tail" of the identity distribution where less evidence is available. We are also interested in exploring iterative approaches to jointly resolving mentions.

Acknowledgments

The authors would like to thank Lise Getoor for her helpful advice.

[5]This also occurs even when topical context is combined with only social context.

References

Daniel J. Abadi. 2003. Comparing domain-specific and non-domain-specific anaphora resolution techniques. Cambridge University MPhil Dissertation.

Alfred V. Aho and Margaret J. Corasick. 1975. Efficient string matching: an aid to bibliographic search. In *Communications of the ACM*.

James Allan, editor. 2002. *Topic detection and tracking: event-based information organization*. Kluwer Academic Publishers, Norwell, MA, USA.

Bryce Allen. 1989. Recall cues in known-item retrieval. *JASIS*, 40(4):246–252.

Omar Benjelloun, Hector Garcia-Molina, Hideki Kawai, Tait Eliott Larson, David Menestrina, Qi Su, Sutthipong Thavisomboon, and Jennifer Widom. 2006. Generic entity resolution in the serf project. *IEEE Data Engineering Bulletin*, June.

Indrajit Bhattacharya and Lise Getoor. 2006. A latent dirichlet model for unsupervised entity resolution. In *The SIAM International Conference on Data Mining (SIAM-SDM)*, Bethesda, MD, USA.

Indrajit Bhattacharya and Lise Getoor. 2007. Collective entity resolution in relational data. *ACM Transactions on Knowledge Discovery from Data*, 1(1), March.

Matthias Blume. 2005. Automatic entity disambiguation: Benefits to NER, relation extraction, link analysis, and inference. In *International Conference on Intelligence Analysis*, May.

Chris Diehl, Lise Getoor, and Galileo Namata. 2006. Name reference resolution in organizational email archives. In *Proceddings of SIAM International Conference on Data Mining*, Bethesda, MD , USA, April 20-22.

Tamer Elsayed and Douglas W. Oard. 2006. Modeling identity in archival collections of email: A preliminary study. In *Proceedings of the 2006 Conference on Email and Anti-Spam (CEAS 06)*, pages 95–103, Mountain View, California, July.

Ralf Holzer, Bradley Malin, and Latanya Sweeney. 2005. Email alias detection using social network analysis. In *LinkKDD '05: Proceedings of the 3rd international workshop on Link discovery*, pages 52–57, New York, NY, USA. ACM Press.

Bryan Klimt and Yiming Yang. 2004. Introducing the Enron corpus. In *Conference on Email and Anti-Spam*, Mountain view, CA, USA, July 30-31.

David D. Lewis and Kimberly A. Knowles. 1997. Threading electronic mail: a preliminary study. *Inf. Process. Manage.*, 33(2):209–217.

David D. Lewis. 1996. The trec-4 filtering track. In *The Fourth Text REtrieval Conference (TREC-4)*, pages 165–180, Gaithersburg, Maryland.

Xin Li, Paul Morie, and Dan Roth. 2005. Semantic integration in text: from ambiguous names to identifiable entities. *AI Magazine. Special Issue on Semantic Integration*, 26(1):45–58.

Bradley Malin. 2005. Unsupervised name disambiguation via social network similarity. In *Workshop on Link Analysis, Counter-terrorism, and Security, in conjunction with the SIAM International Conference on Data Mining*, Newport Beach, CA, USA, April 21-23.

Gideon S. Mann and David Yarowsky. 2003. Unsupervised personal name disambiguation. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003*, pages 33–40, Morristown, NJ, USA. Association for Computational Linguistics.

Andrew McCallum, Andres Corrada-Emmanuel, and XueruiWang Wang. 2005. Topic and role discovery in social networks. In *IJCAI*.

Einat Minkov, William W. Cohen, and Andrew Y. Ng. 2006. Contextual search and name disambiguation in email using graphs. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 27–34, New York, NY, USA. ACM Press.

Patric Reuther. 2006. Personal name matching: New test collections and a social network based approach.