Extracting Personal Names from Email: Applying Named Entity Recognition to Informal Text

Abstract

There has been little prior work on Named Entity Recognition for ”informal” documents like email. We present two methods for improving performance of person name recognizers for email: email-specific structural features and a recall-enhancing method which exploits name repetition across multiple documents.

1 Introduction

Named entity recognition (NER), the identification of entity names in free text, is a well-studied problem. In most previous work, NER has been applied to news articles (e.g., (Bikel et al., 1999; McCallum and Li, 2003)), scientific articles (e.g., (Craven and Kumlien, 1999; Bunescu and Mooney, 2004)), or web pages (e.g., (Freitag, 1998)). These genres of text share two important properties: documents are written for a fairly broad audience, and writers take care in preparing documents. Important genres that do *not* share these properties include instant messaging logs, newsgroup postings and email messages. We refer to these genres as “informal” text.

Informal text is harder to process automatically. Informal documents do not obey strict grammatical conventions. They contain grammatical and spelling errors. Further, since the audience is more restricted, informal documents often use group- and task-specific abbreviations and are not self-contained. Because of these differences, existing NER methods may require modifications to perform well on informal text.

In this paper, we investigate NER for informal text with an experimental study of the problem of recognizing personal names in email—a task that is both useful and non-trivial. An application of interest is corpus anonymization. Automatic or semi-automatic email anonymization should allow using large amounts of informal text for research purposes, for example, of medical files. Person-name extraction and other NER tasks are helpful for automatic processing of informal text for a large variety of applications (Culotta et al., 2004; Cohen et al., 2005).

We first present four corpora of email text, annotated with personal names, each roughly comparable in size to the MUC-6 corpus[1]. We experimentally evaluate the performance of conditional random fields (CRF) (Lafferty et al., 2001), a state-of-the art machine-learning based NER methods on these corpora. We then turn to examine the special attributes of email text (vs. newswire) and suggest venues for improving extraction performance. One important observation is that email messages often include some structured, easy-to-recognize names, such as names within a header, names appearing in automatically-generated phrases, as well as names in signature files or sign-offs. We therefore suggest a set of specialized *structural* features for email; these features are shown to significantly improve performance on our corpora.

We also present and evaluate a novel method for exploiting *repetition* of names in a test corpus. Techniques for exploiting name repetition within documents have been recently applied to newswire text

[1]Two of these are publicly available. The others can not be distributed due to privacy considerations.

(e.g., (Humphreys et al., 1998)), scientific abstracts (e.g., (Bunescu and Mooney, 2004)) and seminar announcements ((Sutton and Mccallum, 2004)); however, these techniques rely on either NP analysis or capitalization information to pre-identify candidate coreferent name mentions, features which are not reliable in email. Furthermore, we argue that name repetition in email should be inferred by examining *multiple documents* in a corpus, which is not common practice. We therefore present an alternative efficient scheme for increasing recall in email, using the whole corpus. This technique is shown to always improve recall substantially, and to almost always improve F1 performance.

2 Corpora

Two email corpora used in our experiments were extracted from the CSpace email corpus (Kraut et al., 2004), which contains email messages collected from a management course conducted at Carnegie Mellon University in 1997. In this course, MBA students, organized in teams of four to six members, ran simulated companies in different market scenarios. We believe this corpus to be quite similar to the work-oriented mail of employees of a small or medium-sized company. This text corpus contains three header fields: "From", "Subject", and "Time". *Mgmt-Game* is a subcorpora consisting of all emails written over a five-day period. In the experiments, the first day worth of email was used as a training set, the fourth for tuning and the fifth day as a test set. *Mgmt-Teams* forms another split of this data, where the training set contains messages between different teams than in the test set; hence in *Mgmt-Teams*, the person names appearing in the test set are generally different than those that appear in the training set.

The next two collections of email were extracted from the Enron corpus (Klimt and Yang, 2004). The first subset, *Enron-Meetings*, consists of messages in folders named "meetings" or "calendar"[2]. Most but not all of these messages are meeting-related. The second subset, *Enron-Random*, was formed by repeatedly sampling a user name (uniformly at random among 158 users), and then sampling an email from that user (uniformly at random).

Annotators were instructed to include nicknames and misspelled names, but exclude person names that are part of an email address and names that are part of a larger entity name like an organization or location (e.g., "David Tepper School of Business").

The sizes of the corpora are given in Table 1. We limited training size to be relatively small, reflecting a real-world scenario.

Corpus	# Documents			#Words x1000	#Names
	Train	Tune	Test		
Mgmt-Teams	120	82	83	105	2,792
Mgmt-Game	120	216	264	140	2,993
Enron-Meetings	244	242	247	204	2,868
Enron-Random	89	82	83	286	5,059

Table 1: Summary of the corpora used in the experiments. The number of words and names refer to the whole annotated corpora.

3 Existing NER Methods

In our first set of experiments we apply CRF, a machine-learning based probabilistic approach to labeling sequences of examples, and evaluate it on the problem of extracting personal names from email. Learning reduces NER to the task of *tagging* (i.e., classifying) each word in a document. We use a set of five tags, corresponding to (1) a one-token entity, (2) the first token of a multi-token entity, (3) the last token of a multi-token entity, (4) any other token of a multi-token entity and (5) a token that is not part of an entity.

The sets of features used are presented in Figure 3. All features are instantiated for the focus word, as well as for a window of 3 tokens to the left and to the right of the focus word. The *basic features* include the lower-case value of a token t, and its *capitalization pattern*, constructed by replacing all capital letters with the letter "X", all lower-case letters with "x", all digits with "9" and compressing runs of the same letter with a single letter. The *dictionary features* define various categories of words including common words, first names, last names [3] and "roster names" [4] (international names list, where first and

[2] with two exceptions: (a) six very large files were removed, and (b) one very large "calendar" folder was excluded.

[3] We used US Census' lists of the most common first and last names in the US, available from http://www.census.gov/genealogy/www/freqnames.html

[4] A dictionary of 16,623 student names across the country, obtained as part of the RosterFinder project (Sweeney, 2003)

Basic Features
t, lexical value, lowercase (binary form, e.g. $f(t$="hello")=1)
capitalization pattern of t (binary form, e.g. $f(t.cap$=x+)=1)
Dictionary Features
inCommon: t in common words dictionary
inFirst: t in first names dictionary
inLast: t in last names dictionary
inRoster: t in roster names dictionary
First: inFirst $\cap$ $\neg$isLast $\cap$ $\neg$inCommon
Last: $\neg$inFirst $\cap$ inLast $\cap$ $\neg$inCommon
Name: (First $\cup$ Last $\cup$ inRoster) $\cap$ $\neg$ inCommon
Title: t in a personal prefixes/suffixes dict.
Org: t in organization suffixes dic
Loc: t in location suffixes dict
Email Features
t appears in the header
t appears in the "from" field
t is a probable "signoff"
($\approx$ after two line breaks and near end of message)
t is part of an email address (regex)
does the word starts a new sentence
($\approx$ capitalized after a period, question or exclamation mark)
t is a probable initial (X or X.)
t followed by the bigram "and I"
t capitalized and followed by a pronoun within 15 tokens

Table 2: Feature sets

last names are mixed.) In addition, we constructed some composite dictionary features, as specified in Table 3: for example, a word that is in the first-name dictionary and is not in the common-words or last-name dictionaries is designated a "sure first name".

The common-words dictionary used consists of base forms, conjugations and plural forms of common English words, and a relatively small ad-hoc dictionary representing words especially common in email (e.g., "email", "inbox"). We also use small manually created word dictionaries of prefixes and suffixes indicative of persons (e.g., "mr", "jr"), locations (e.g., "ave") and organizations (e.g., "inc").

Email structure features: We perform a simplified document analysis of the email message and use this to construct some additional features. One is an indicator as to whether a token t is equal to some token in the "from" field. Another indicates whether a token t in the email body is equal to some token appearing in the whole header. An indicator feature based on a regular expression is used to mark tokens that are part of a probable "sign-off" (i.e., a name at the end of a message). Finally, since the annotation rules do not consider email addresses to be names, we added an indicator feature for tokens that are inside an email address.

l.2.mr	l.1.president
l.2.mrs	l.2.dr
l.1.jr	r.2.who
l.1.judge	r.2.jr
r.3.staff	l.3.by
l.2.ms	r.3.president
r.2.staff	l.3.by
r.1.family	l.3.rep
l.3.says	l.2.rep
r.3.reporter	r.1.administration

l.1.by	r.2.home
l.2.by	r.1.or
l.3.name	l.1.with
l.2.name	l.1.thanks
l.3.by	r.1.picked
r.3.his	l.3.meet
r.1.ps	r.1.started
r.3.home	r.1.told
r.1.and	l.2.prof
l.1.called	l.2.email

Figure 1: Predictive contexts for personal-name words for MUC-6 (left) and Mgmt-Game (right) corpora. A features is denoted by its direction comparing to the focus word (l/r), offset and lexical value.

We experimented with features derived from POS tags and NP-chunking of the email, but found the POS assignment too noisy to be useful. We did include some features based on approximate linguistic rules. One rule looks for capitalized words that are not common words and are followed by a pronoun within a distance of up to 15 tokens. (As an example, consider "Contact Puck tomorrow. *He* should be around."). Another rule looks for words followed by the bigram "and I". As is common for hand-coded NER rules, both these rules have high precision and low recall.

3.1 Email vs Newswire

In order to explore some of the differences between email and newswire NER problems, we stripped all header fields from the Mgmt-Game messages, and trained a model (using basic features only) from the resulting corpus of email bodies. Figure 1 shows the features most indicative of a token being part of a name in the models trained for the Mgmt-Game and MUC-6 corpora. To make the list easier to interpret, it includes only the features corresponding to tokens surrounding the focus word.

As one might expect, the important features from the MUC-6 dataset are mainly formal name titles such as "mr", "mrs", and "jr", as well as job titles and other pronominal modifiers such as "president" and "judge". However, for the Mgmt-Game corpus, most of the important features are related to email-specific structure. For example, the features "left.1.by" and "left.2.by" are often associated with a quoted excerpt from another email message, which in the Mgmt-Game corpus is often marked by mailers with text like "Excerpts from mail: 7-

Sep-97 Re: paper deadline by Richard Wang". Similarly, features like "left.1.thanks" and "right.1.ps" indicate a "signoff" section of an email, as does "right.2.home" (which often indicates proximity to a home phone number appearing in a signature).

3.2 Experimental Results

We now turn to evaluate the usefulness of the feature sets described above. Table 3 gives entity-level F1 performance [5] for CRF trained models for all datasets, using the basic features alone (B); the basic and email-tailored features (B+E); the basic and dictionary features (B+D); and, all of the feature sets combined (B+D+E). All feature sets were tuned using the Mgmt-Game valudation subset. The given results relate to previously unseen test sets.

Dataset	B	B+E	B+D	B+D+E
Mgmt-Teams	68.1	75.7	82.0	87.9
Mgmt-Game	79.2	84.2	90.7	91.9
Enron-Meetings	59.0	71.5	78.6	76.9
Enron-Random	68.1	70.2	72.9	76.2

Table 3: F1 entity-leavel performance for the sets of features, across all datasets, with CRF training.

The results show that the email-specific features are very informative. In addition, they show that the dictionary features are especially useful. This can be explained by the relatively weak contextual evidence in email. While dictionaries are useful in named entities extraction in general, they are in fact more essential when extracting names from email text, where many name mentions are part of headers, names lists etc. Finally, the results for the combined feature set are superior in most cases to any subset of the features.

Overall the level of performance using all features is encouraging, considering the limited training set size. Performance on Mgmt-Teams is somewhat lower than for Mgmt-Game mainly because (by design) there is less similarity between training and test sets with this split. Enron emails seem to be harder than Mgmt-Game emails, perhaps because they include fewer structured instances of names. Enron-Meetings emails also contain a number of constructs that were not encountered in the Mgmt-Game corpus, notably lists (e.g., of people attending a meeting), and also include many location and organization names, which are rare in Mgmt-Game. A larger set of dictionaries might improve performance for the Enron corpora.

[5] No credit awarded for partially correct entity boundaries.

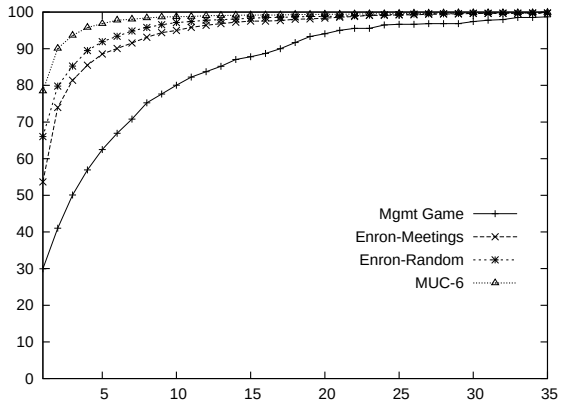


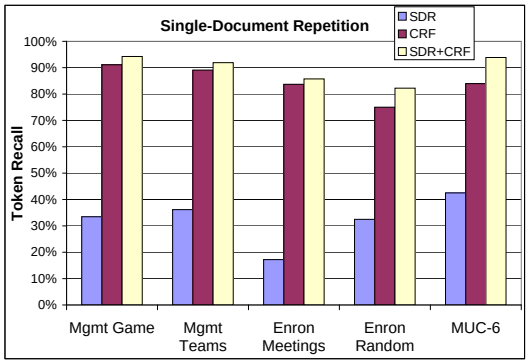
Figure 2: Cumulative percentage of person-name tokens w that appear in at most K distinct documents as a function of K.

4 Repetition of named entities in email

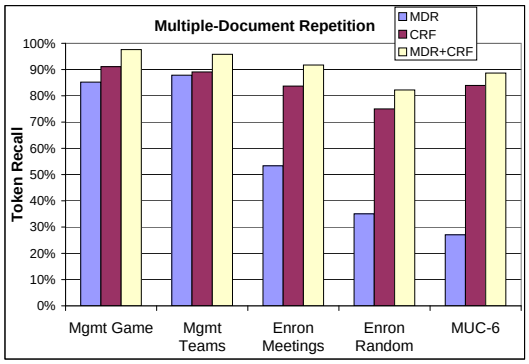
In the experiments described above, the extractors have high precision, but relatively low recall. This typical behavior suggests that some sort of recall-enhancing procedure might improve overall performance.

One family of recall-enhancing techniques are based on looking for multiple occurrences of names in a document, so that names which occur in ambiguous contexts will be more likely to be recognized. It is an intuitive assumption that the ways in which names repeat themselves in a corpus will be different in email and newswire text. In news stories, one would expect repetitions within a *single document* to be common, as a means for an author to establish a shared context with the reader. In an email corpus, one would expect names to repeat more frequently across the corpus, in *multiple documents*—at least when the email corpus is associated with a group that works together closely. In this section we support this conjecture with quantitative analysis.

In a first experiment, we plotted the percentage of person-name tokens w that appear in at most K distinct documents as a function of K. Figure 2 shows this function for the Mgmt-Game, MUC-6, Enron-Meetings, and Enron-Random datasets. There is a large separation between MUC-6 and Mgmt-Game, the most workgroup-oriented email corpus. In MUC-6, for instance, almost 80% of the



(a) SDR



(b) MDR

Figure 3: Upper bounds on recall and recall improvements associated with methods that look for terms that re-occur within a single document (SDR) or across multiple documents (MDR).

names appear in only a single document, while in Mgmt-Game, only 30% of the names appear in only a single document. At the other extreme, in MUC-6, only 1.3% of the names appear in 10 or more documents, while in Mgmt-Game, almost 20% do. The Enron-Random and Enron-Meetings datasets show distributions of names that are intermediate between Mgmt-Game and MUC-6.

As a second experiment, we implemented two very simple extraction rules. The *single document repetition* (SDR) rule marks every token that occurs more than once inside a single document as a name. Adding tokens marked by the SDR rule to the tokens marked by the learned extractor generates a new extractor, which we will denote SDR+CRF. Thus, the recall of SDR+CRF serves as an upper bound on the token recall[6] of any recall-enhancing method that improves the extractor by exploiting repetition within a single document. Analogously, the *multiple document repetition* (MDR) rule marks every token that occurs in more than one document as a name. Again, the token recall of MDR+CRF rule is an upper bound on the token recall of any recall-enhancing method that exploits token repetition across multiple documents.

The left bars in Figure 3 show the recall obtained by the SDR (top) and the MDR rule (bottom). The MDR rule has highest recall for the two Mgmt-Game corpora, and lowest recall for the MUC-6 corpus. Conversely, for the SDR rule, the highest recall level obtained is for MUC-6. The middle bars show the token recall obtained by the CRF extractor, using all features. The right bars show the token recall of the SDR+CRF and MDR+CRF extractors. Comparing them to the other bars, we see that the maximal potential recall gain from a SDR-like method is on MUC-6. For MDR-like methods, there are large potential gains on the Mgmt-Game corpora as well as on Enron-Meetings and Enron-Random to a lesser degree. This probably reflects the fact that the Enron corpora are from a larger and more weakly interacting set of users, compared to the Mgmt-Game datasets.

These results demonstrate the importance of exploiting repetition of names across multiple documents for entity extraction from email.

5 Improving Recall With Inferred Dictionaries

Sequential learners of the sort used here classify tokens from each document independently; moreover, the classification of a word w is independent of the classification of other occurrences of w elsewhere in the document. That is, the fact that a word w has appeared somewhere in a context that clearly indicates that it is name does not increase the probability that it will be classified as a name in other, more ambiguous contexts.

Recently, sequential learning methods have been extended to directly utilize information about name co-occurrence in learning the sequential classifier. This approach provides with an elegant solution to modeling repetition within a single document. However, it requires identifying candidate related en-

[6]Token level recall is recall on the task of classifying tokens as inside or outside an entity name.

tities in advance, applying some heuristic. Thus, (Bunescu and Mooney, 2004) link between similar NPs (requiring their head to be identical), and (Sutton and Mccallum, 2004) connect pairs of identical capitalized words. Given that in email corpora capitalization patterns are not followed to a large extent, there is no adequate heuristic that would link candidate entities prior to extraction. Further, it is not clear if a collective classification approach can scale to modeling multiple-document repetition.

We suggest an alternative approach of recall-enhancing names matching, which is appropriate for email. Our approach has points of similarity to the methods described by Stevenson and Gaizauskas (2000), who suggest matching text against name dictionaries, filtering out names that are also common words or appear as non-names in high proportion in the training data. The approach described here is more systematic and general. In a nutshell, we suggest applying the *noisy* dictionary of predicted names over the test corpus, and use the approximate (predicted) name to non-name proportions over the test set itself to filter out ambiguous names. Therefore, our approach does not require large amount of annotated training data. It also does not require word distribution to be similar between train and test data. We will now describe our approach in detail.

5.1 Matching names from dictionary

First, we construct a dictionary comprised of all spans predicted as names by the learned model. For personal names, we suggest expanding this dictionary further, using a transformation scheme. Such a scheme would construct a family of possible variations of a name n: as an example, Figure 4 shows name variations created for the name span "Benjamin Brown Smith". Once a dictionary is formed, a single pass is made through the corpus, and every longest match to some name-variation is marked as a name[7]. It may be that a partial name span n_1 identified by the extractor is subsumed by the full name span n_2 identified by the dictionary-matching scheme. In this case, entity-level precision is increased, having corrected the entity's boundaries.

[7]Initials-only variants of a name, e.g., "bs" in Figure 4 are marked as a name only if the "inSignoff" feature holds—i.e., if they appear near the end of a message in an apparent signature.

benjamin brown smith	benjamin-brown-s.	b. brown s.	bbs
benjamin-brown smith	benjamin-b. s.	b. b. smith	bs
benjamin brown-smith	benjamin-smith	b. brown-s.	
benjamin-brown-smith	benjamin smith	benjamin	
benjamin brown s.	b. brown smith	brown	
benjamin-b. smith	benjamin b. s.	smith	
benjamin b. smith	b. brown-smith	b. smith	
benjamin brown-s.	benjamin-s.	b. b. s	
benjamin-brown s.	benjamin s.	b. s.	

Figure 4: Names variants created from the name "Benjamin Brown Smith"

5.2 Dictionary-filtering schemes

The noisy dictionary-matching scheme is susceptible to false positives. That is, some words predicted by the extractor to be names are in fact non-names. Presumably, these non-names could be removed by simply eliminating low-confidence predictions of the extractor; however, ambiguous words –that are not exclusively personal names in the corpus– may need to be identified and removed as well. We note that ambiguity better be evaluated in the context of the corpus. For example, "Andrew" is a common first name, and may be confidently (and correctly) recognized as one by the extractor. However, in the Mgmt-Game corpus, "Andrew" is also the name of an email server, and most of the occurrences of this name in this corpus are *not* personal names. The high frequency of the word "Andrew" in the corpus, coupled with the fact that it is only sometimes a name, means that adding this word to the dictionary leads to a substantial drop in precision.

We therefore suggest a measure for filtering the dictionary. This measure combines two metrics. The first metric, *predicted frequency* (PF), estimates the degree to which a word appears to be used consistently as a name throughout the corpus:

$$PF(w) \equiv \frac{cpf(w)}{ctf(w)}$$

where $cpf(w)$ denotes the number of times that a word w is predicted as part of a name by the extractor, and $ctf(w)$ is the number of occurrences of the word w in the entire test corpus (we emphasize that estimating this statistic based on test data is valid, as it is fully automatic "blind" procedure).

Predicted frequency does not assess the likely cost of adding a word to a dictionary: as noted above, ambiguous or false dictionary terms that occur frequently will degrade accuracy. A number of statistics could be used here; for instance, practitioners

sometimes filter a large dictionary by simply discarding all words that occur more than k times in a test corpus. We elected to use the *inverse document frequency* (IDF) of w to measure word frequency:

$$IDF(w) \equiv \frac{\log(\frac{N+0.5}{df(w)})}{\log(N+1)}$$

Here $df(w)$ is the number of documents that contain a word w, and N is the total number of documents in the corpus. Inverse document frequency is often used in the field of information retrieval (Allan et al., 1998), and the formula above has the virtue of being scaled between 0 and 1 (like our PF metric) and of including some smoothing. In addition to bounding the cost of a dictionary entry, the IDF formula is in itself a sensible filter, since personal names will not appear as frequently as common English words.

The joint filter combines these two multiplicatively, with equal weights:

$$PF.IDF(w) : PF(w) \times IDF(w)$$

PF.IDF takes into consideration both the probability of a word being a name, and how common it is in the entire corpus. Words that get low PF.IDF scores are therefore either words that are highly ambiguous in the corpus (as derived from the extractors' predictions) or are common words, which were inaccurately predicted as names by the extractor.

In the MDR method of Figure 3, we imposed an artificial requirement that words must appear in more than one document. In the method described here, there is no such requirement: indeed, words that appear in a small number of documents are given higher weights, due to the IDF factor. Thus this approach exploits both single-document and multiple-document repetitions.

In a set of experiments that are not described here, the PF.IDF measure was found to be robust to parameter settings, and also preferable to its separate components in improving recall at minimal cost in precision. As described, the PF.IDF values per word range between 0 and 1. One can vary the threshold, under which a word is to be removed from the dictionary, to control the precision-recall trade-off. We tuned the PF.IDF threshold using the validation subsets, optimizing entity-level F1 (a value of 0.16 was found optimal).

In summary, our recall-enhancing strategy is as follows:

1. Learn an extractor E from the training corpus C_{train}.
2. Apply the extractor E to a test corpus C_{test} to assign a preliminary labeling.
3. Build a dictionary S_{θ_*} including the names n such that (a) n is extracted somewhere in the preliminary labeling of the test corpus, or is derived from an extracted name applying the name transformation scheme and (b) $PF.IDF(n) > \theta_*$.
4. Apply the dictionary-matching scheme of Section 5.1, using the dictionary S_{θ_*} to augment the preliminary labeling, and output the result.

5.3 Experiments with inferred dictionaries

Table 4 shows results using the method described above. We consider all of the email corpora and the CRF learner, trained with the full feature set. The results are given in terms of relative change, compared to the baseline results generated by the extractors ($score_{result}/score_{baseline} - 1$) and final value.

As expected, recall is always improved. Entity-level F1 is increased as well, as recall is increased more than precision is decreased. The largest improvements are for the Mgmt-Game corpora —the two e-mail datasets shown to have the largest potential improvement from MDR-like methods in Figure 3. Recall improvements are more modest for the Enron datasets, as was anticipated by the MDR analysis. Another reason for the gap is that extractor baseline performance is lower for the Enron datasets, so that the Enron dictionaries are noisier.

As detailed in Section 2, the Mgmt-Teams dataset was constructed so that the names in the training and test set have only minimal overlap. The performance improvement on this dataset shows that repetition of mostly-novel names can be detected using our method. This technique is highly effective when names are novel, or dense, and is optimal when extractor baseline precision is relatively high.

Dataset	Precision	Recall	F1
Mgmt-Teams	-0.9% / 92.9	+8.5% / 89.8	+3.9% / 91.3
Mgmt-Game	-0.8% / 94.5	+8.4% / 96.2	+3.8% / 95.4
Enron-Meetings	-2.5% / 81.1	+4.7% / 74.9	+1.2% / 77.9
Enron-Random	-3.8% / 79.2	+4.9% / 74.3	+0.7% / 76.7

Table 4: Entity-level relative improvement and final result, applying name-matching on models trained with CRF and the full feature set (F1 baseline given in Table 3).

6 Conclusion

This work applies recently-developed sequential learning methods to the task of extraction of named entities from email. This problem is of interest as an example of NER from informal text—text that has been prepared quickly for a narrow audience.

We showed that informal text has different characteristics from formal text such as newswire. Analysis of the highly-weighted features selected by the learners showed that names in informal text have different (and less informative) types of contextual evidence. However, email also has some structural regularities which make it easier to extract personal names. We presented a detailed description of a set of features that address these regularities and significantly improve extraction performance on email.

In the second part of this paper, we analyzed the way in which names repeat in different types of corpora. We showed that repetitions within a single document are more common in newswire text, and that repetitions that span multiple documents are more common in email corpora. Additional analysis confirms that the potential gains in recall from exploiting multiple-document repetition is much higher than the potential gains from exploiting single-document repetition.

Based on this insight, we introduced a simple and effective method for exploiting multiple-document repetition to improve an extractor. One drawback of the recall-enhancing approach is that it requires the entire test set to be available: however, our test sets are of only moderate size (83 to 264 documents), and it is likely that a similar-size sample of unlabeled data would be available in many practical applications. The approach substantially improves recall and often improves F1 performance; furthermore, it can be easily used with any NER method.

Taken together, extraction performance is substantially improved by this approach. The improvements seem to be strongest for email corpora collected from closely interacting groups. On the Mgmt-Teams dataset, which was designed to reduce the value of memorizing specific names appearing in the training set, F1 performance is improved from 68.1% for the out-of-the-box system (or 82.0% for the dictionary-augmented system) to 91.3%. For the less difficult Mgmt-Game dataset, F1 performance is improved from 79.2% for an out-of-the-box CRF-based NER system (or 90.7% for a CRF-based system that uses several large dictionaries) to 95.4%.

References

J. Allan, J. Callan, W.B. Croft, L. Ballesteros, D. Byrd, R. Swan, and J. Xu. 1998. Inquery does battle with trec-6. In *TREC-6*.

D. M. Bikel, R. L. Schwartz, and R. M. Weischedel. 1999. An algorithm that learns what's in a name. *Machine Learning*, 34:211–231.

R. Bunescu and R. J. Mooney. 2004. Relational markov networks for collective information extraction. In *ICML-2004 Workshop on Statistical Relational Learning*.

W. W. Cohen, E. Minkov, and A. Tomasic. 2005. Learning to undertand website update requests. In *IJCAI-05*.

M. Craven and J. Kumlien. 1999. Constructing biological knowledge bases by extracting information from text sources. In *ISMB-99*.

A. Culotta, R. Bekkerman, and A. McCallum. 2004. Extracting social networks and contact information from email and the web. In *CEAS-04*.

D. Freitag. 1998. Information extraction from html: application of a general machine learning approach. In *AAAI-98*.

K. Humphreys, R. Gaizauskas, S. Azzam, C. Huyck, B. Mitchell, H. Cunningham, and Y. Wilks. 1998. Description of the LASIE-II system as used for MUC-7.

B. Klimt and Y. Yang. 2004. Introducing the Enron corpus. In *CEAS-04*.

R. E. Kraut, S. R. Fussell, F. J. Lerch, and J. A. Espinosa. 2004. Coordination in teams: evi-dence from a simulated management game. To appear in the Journal of Organizational Behavior.

J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML-01*.

A. McCallum and W. Li. 2003. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *CoNLL-2003*.

M. Stevenson and R. Gaizauskas. 2000. Using corpus-derived names lists for named entities recognition. In *NAACL-2000*.

C. Sutton and A. Mccallum. 2004. Collective segmentation and labeling of distant entities in information extraction. In *ICML workshop on Statistical Relational Learning*.

L. Sweeney. 2003. Finding lists of people on the web. Technical Report CMU-CS-03-168, CMU-ISRI-03-104. http://privacy.cs.cmu.edu/dataprivacy/ projects/rosterfinder/.