

Applying Name Entity Recognition to Informal Text

Yu-shan Chang
Department of Computer Science
Stanford University
yschang1@stanford.edu

Yun-Hsuan Sung
Department of Electrical Engineering,
Stanford University
yhsung@stanford.edu

Abstract – Although Name Entity Recognition (NER) has been a well-studied problem in recent years, it is seldom applied to informal document, such as E-mail message and Newsgroup postings. Unlike the formal text, which is well-structured and with seldom error, the name entities are much more difficult to recognize in informal one. The key problems for informal text are that it has unstructured properties, more grammatical error, and more spelling error. All of these properties will degrade the performance of the existent classifiers and well-designed features which are suitable for original NER. In this project, we are going to apply two approaches, Maximum Entropy Classifier (MaxEnt) and Conditional Random Field (CRF), which are often used for formal text NER, to informal text NER. We do some experiment to show if they are still good for the informal task. We also focus on how to extract efficient and effective features especially for informal text.

I. INTRODUCTION

Name Entity Recognition (NER) has been a well-studied problem for formal text. The key problem is to extract some special name entities, such as person name, location, and organization, in a text. There are several classification methods which are successful to be applied on this task. Chieu and Ng[1] and Bender et al.[2] used Maximum Entropy approach as the classifier. Conditional Random Field (CRF) was explored by McCallum and Li [3] to NER. Mayfield et al.[4] applied Support Vector Machine (SVM) to classify each name entity. Florian et al. [5] even combined Maximum Entropy and hidden Markov Model (HMM) under different conditions. Some other researches are focused more on extracting some efficient and effective features for NER. Chieu and Ng[1] successful used local features, which are near the word, and global features, which are in the whole document together. Klein et al.[6] and Whitelaw et al.[7] reports that character-based features are useful for recognizing some special structure for the name entity.

All of these works are only applied to formal text, which is a collection of news wire articles from the Reuters Corpus¹. These kinds of texts are well-structured, well organized, and

have seldom grammatical and spelling error. The sentences are well-formed and easily to detect the sentence boundary. There are only few non-word characters in this kind of text, which are easily confused when doing Name Entity Recognition. All of these properties of formal text make NER relatively easier compared with informal text. Until now, there are still seldom works on this task. Minkov[8] has some early research results about informal text NER by using Hidden Markov Model (HMM) and Conditional Random Field (CRF).

In our project, we are going to explore Maximum Entropy Classifier (MaxEnt) and Conditional Random Field (CRF) in informal text NER. The informal text can separate into two subsets, E-mail message and NewsGroups postings. We used some special features which are designed for E-mail combined with those features which are general used in formal text NER. The following report is organized as follows. We first clarify the problem settings and define the performance metrics in Section II. Then, the two different approaches we used are explained in Section III. In section IV, we describe the feature engineering work in the following experiment. Section V is the corpora and experiment setting. The experiment results and error analysis are shown in section VI. Finally, we conclude this report in Section VII.

II. PROBLEM STATEMENT

The main problem of NER is that we want to recognize the name entity, which is general organized into personal name, location, and organization, from a document. In general, we extract some features, either string label or indicator function for each word. Then, we use labeled training data and extracted features to train a classifier by numerical optimization algorithm to get the optimized parameters. Finally, given the features of a word in testing data, we use this classifier to recognize which label it belongs to.

The features we used can separate into two different categories. The first one is string label, which is just the description of the features. For example, “isFirstWord=1” means the word is the first word in a sentence. Another one is using the indicator function, which can be seen as the answer of a true or false question. For example:

¹ <http://about.reuters.com/researchandstandards/corpus/>

$$f(w) = \begin{cases} 1 & , \text{ if } w \text{ is the first word of a sentence} \\ 0 & , \text{ otherwise} \end{cases}$$

Unlike formal text, informal texts have special properties compared with formal text. It makes the original well-designed features not suitable for this new extension any more. We need to observe these properties and find some new suitable features.

The first property is that e-mail and newsgroup posting always begin with some well-defined header, which has special but less syntax structure. Take email as example,

```
Message-ID: <7789106.1072119990022.JavaMail.evans@thyme>
Date: Mon, 4 Feb 2002 09:27:45 -0800 (PST)
From: ed.mcmichael@enron.com
To: ed.mcmichael@enron.com, eric.bass@enron.com,
    jonathan.mckay@enron.com, h.lewis@enron.com,
    mathew.smith@enron.com, frank.ermis@enron.com,
    dutch.quigley@enron.com, keith.considine@enron.com
Subject: Updated: Gas Curves Validation
Cc: shona.wilson@enron.com
Mime-Version: 1.0
Content-Type: text/plain; charset=us-ascii
Content-Transfer-Encoding: 7bit
Bcc: shona.wilson@enron.com
```

If we treat this part as general paragraph, we won't get good result due to its special format. However, we found that all the name entity only exist in some special headers, such as From, to, ...etc. We can do some special process for these headers to extract the name entity.

Another special zone is the "signature", which is general found at the end of email and posting. We always have the name is this signature. However, just like the headers, it lacks sentence structure and is difficult to use general features to find the name entity. Lacking common format makes it even more difficult to deal with than the headers. There is also some research to extract the signature from email (Carvalho and Cohen, 2004). For example:

```
Ex 1.
---
George Patapis -----PAN METRON ARISTON-----
Telecom C.S.S.C Lane Cove---email:gpatapis@cscs-syd.tansu.com.au
P.O.Box A792 Sydney South --fax : (02) 911 3 199-----
NSW, 2000, Australia.-----voice:(02) 911 3 121-----

Ex 2
--
John R. Vanderpool   INTERNET: fish@eosdata.gsfc.nasa.gov
NASA/GSFC/HSTX      VOX: 301-513-1683
"So you run, and you run, to catch up with the sun, but it's sinking,
racing around to come up behind you again." -rw/d
```

Even in the text zone, the structure of those sentences is not well-formed and it has lot of grammar and spelling errors. Further, since the senders and receivers are restricted, they contain more abbreviation and group-specific and task-related jargon. For example

```
Ex 1
When: Monday, October 01, 2001 4:30 PM-5:30 PM (GMT-06:00) Central
Time (US & Canada).
Where: EB-3143B
*~*~*~*~*~*~*~*~*
```

```
Ex 2
Could someone PLEASE give a guess as to why this simple little program
causes a BadPixmap error on the FOURTH (bizarre???) call to XtRe-
laizeWidget()?
```

Here is the code:

```
int stoploop = 0;
static void Callback(Window, XtPointer, XtPointer);
...
```

We have four different performance metrics, accuracy, precision, recall, and F1. Their definitions are as follow:

	Correct	Not correct
Selected	tp	fp
Not selected	fn	tn

$$Accuracy = \frac{tp + tn}{tp + fp + fn + tn}, \quad Precision: P = \frac{tp}{tp + fp}$$

$$Recall \quad R = \frac{tp}{tp + fn}, \quad F1 = \frac{1}{\alpha \frac{1}{P} + (1 - \alpha) \frac{1}{R}}$$

Accuracy is not a good performance metric because most of our labels are not name entity. Since there are more non-name words than true names, the naïve classifier that classifies every word to non-name will still receive high accuracy. Precision is the metric of how many percentages you make good selection. Recall is the metric of how percentages the correct answer you select. Both of them have their meaning, so the general metric is the combination of these two, F1. In our project, we used balanced F1 measure, which sets $\lambda_k = 1/2$.

III. METHOD

Maximum Entropy Classifier

In the Maximum Entropy Classifier, we want to find the distribution with maximum entropy given some feature-based constraints. We use an exponential (log-linear) model to produce a probabilistic model.

$$P(c | d, \lambda) = \frac{\exp \sum_i \lambda_i f_i(c, d)}{\sum_{c'} \exp \sum_i \lambda_i f_i(c', d)}$$

Where C is the class or label, d is the data, and λ is the model parameters. After taking logarithm, we have,

$$\log P(c | d, \lambda) = \log \left[\exp \sum_i \lambda_i f_i(c, d) \right] - \log \left[\sum_{c'} \exp \sum_i \lambda_i f_i(c', d) \right]$$

The conditional likelihood is a function of the i.i.d. data (C,D) and the parameters λ . i.e.

$$\log P(C | D, \lambda) = \log \prod_{(c,d) \in (C,D)} P(c | d, \lambda) = \sum_{(c,d) \in (C,D)} \log P(c | d, \lambda)$$

Our goals are to maximum the conditional likelihood or minimize the minus of the conditional likelihood function.

$$\text{minimize } F(\vec{\lambda}) = -1 \left[\sum_{(c,d) \in (C,D)} \log p(c | d, \vec{\lambda}) \right]$$

$$\text{,where } P(c | d, \lambda) = \frac{\exp \sum_i \lambda_i f_i(c, d)}{\sum_{c'} \exp \sum_i \lambda_i f_i(c', d)}$$

The log likelihood function is concave and has a global maximum. We can use numerical optimization algorithm to find the optimum parameters, such as quasi Newtown method and iterative Scaling algorithm.

In this project, we used the Maximum Entropy Classifier code in assignment3 as mainframe to implement our system.

Conditional Random Field (CRF)

Conditional Random Field (CRF) is a probabilistic framework for labeling and segmenting sequential data, which is a generalization both of maximum entropy Markov Model (MEMM) and hidden Markov Model (HMM). Compare with MEMM and HMM, CRF uses undirected graphical models (see Figure 1) to calculate the conditional probability of a particular label sequence \mathbf{y} given observation \mathbf{x} . i.e.

$$P_{\lambda}(y | x) = \frac{1}{Z_x} \exp \left(\sum_{t=1}^T \sum_k \lambda_k f_k(y_{t-1}, y_t, x, t) \right)$$

,where $f_k(y_{t-1}, y_t, x, t)$ is a transition feature function of the entire observation sequence and the labels at positions t and $t-1$ in the label sequence; and λ_k is the parameter to be optimized from training data. Z_x is a normalization factor.

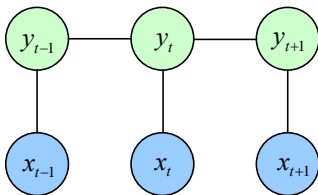


Figure 1, undirected graphical structure in CRF

Each transition feature function takes on the value of one of these real-valued observation features $f(x, i)$ if the previous and current states (in the case of a transition function) take on particular values. All feature functions are therefore real-valued. For example,

$$f_k(y_{t-1}, y_t, x, t) = \begin{cases} 1 & \text{, if } y_{t-1} \text{ is the state } S, y_t \text{ is in} \\ & \text{state } S', \text{ and } x \text{ is a people's name} \\ 0 & \text{, otherwise} \end{cases}$$

The first goal is to estimate the parameters to maximize the likelihood for training data. After taking logarithm, we have

$$\log P_{\lambda}(y | x) = \sum_{t=1}^T \sum_k \lambda_k f_k(y_{t-1}, y_t, x, t) - \log Z_x$$

The log-likelihood function is concave and we have global optimum. Then, it becomes an optimization problem.

$$\text{maximize } \sum_{t=1}^T \sum_k \lambda_k f_k(y_{t-1}, y_t, x, t) - \log Z_x$$

Taking derivative and set it to zero, we get the maximum entropy constraint. The parameters can be found by iterative scaling algorithm.

The second goal is to find the most likely label sequence given an observation sequence. i.e.

$$s^* = \arg \max_s p_{\lambda}(s | o)$$

We can used constrained Viterbi algorithm to find the optimal solution s .

In this project, we used CRF toolbox for Matlab developed by Kevin Murphy².

IV. FEATURE EXTRACTION

Lists derived from training data

The training data is first processed to obtain a number of lists used in later feature extraction. These lists are derived from the training data automatically.

- Frequent Word List (FWL): This word list contains words that occur more than 50 times in all the training data in Enron Emails and Newsgroup corpus.
- Useful Unigram (UUNI): Words that precede true names in training data with occurrences more than 10 times are collected.

² <http://www.cs.ubc.ca/~murphyk/Software/index.html>

- Useful Bigram (UBI): Bigrams of words that precede true names with occurrence more than 10 times are compiled to a list.

Lists from other sources

- First Name List (FNL)/ Last Name List (LNL): US Census report contains the most common first and last names in the US, including 5,494 first names and 88,798 last names. FNL is taken directly from the first name data while only the most common 5,000 last names are included in LNL.³
- Known Name List (KNL): Another name list is obtained from Cognitive Computation Group at University of Illinois at Urbana Champaign.⁴
- Title List (TL): This list contains the possible titles such as mr., jr., chief executive officer, and etc. before a true name.⁵

Basic features

In this paper, w represents the focus token. $w-i$ and $w+i$ refer to the i th words before and after w .

Basic features are extracted without using any external knowledge other than the training data. Therefore, these basic features can also be applied to other alphabetical languages.

Basic features include:

- Capitalized word: If w is capitalized, then this feature is set to 1.
- Hyphenated word: If w has hyphen, then this feature is set to 1.
- CurrentWord, PrevWord, PPrevWord, NextWord: The actual string of w , $w-1$, $w-2$, and $w+1$ are added as features.
- Character-based Unigram, Bi-gram, Tri-gram: We add these features in hopes of recognizing some special prefix or suffix in true names.
- Word length: The length of true names usually falls in certain range. We expect this feature can help us to eliminate some words with extremely long or short length.
- IsFirst: If w is the first word in a sentence, then this feature is set to 1.

Extended Features

- Common word: If w is found in FWL, then this feature is set to 1.
- hasTitle: If $w-1$ is found in TL, then this feature is set to 1.

- Unigrams: Because true name usually contain two words, we check both $w-1$ and $w-2$ to see if either $w-1$ or $w-2$ is in UUNI. If so, then this feature is set to 1.
- Bi-grams: Similar to Unigrams, If $(w-2\ w-1)$ or $(w-3\ w-2)$ is found in UBI, then this feature is set to 1.

We use a Part of Speech (POS) Tagger implemented by Maximum Entropy Classifier and Viterbi decoder. The POSTagger is trained by 500 sentences from the Wall Street Journal. We hope the POS of each word can provide some useful information to identify true names.

- CurrentPos, PrevPos, PPrevPos, NextPos: The POS of $w-2$, $w-1$, w and $w+1$ are added as features.

We apply some heuristics based on the POS of words and the name lists to construct features for names.

- IsNNP: Since true names are tagged as NNP with a very high probability, whether the POS of w is NNP can be a very useful predictor.
- IsAround VB/VBD: We found true names are usually preceded or followed by verbs. For example, please contact/tell/call John. If the POS of w is NNP and there is a VB/VBD/VBG/VBN/VBP/VBZ word within 3 tokens to the left or the right of w , the feature is set to 1.
- AndNNP/PRP: We noticed the structure "... NNP/PRP and <True Name>..." or "...<True Name> and NNP/PRP..." occurs in our training data quite often. If the POS of w is NNP and "NNP/PRP and" or "and NNP/PRP" is directly before or after the w , then the feature is set to 1.
- IsFirstName, IsLastName, IsKnownName: If w occurs in FNL, LNL, and KNL respectively, then the corresponding feature is set to 1.
- IsSureFirstName, IsSureLastName, IsSureKnownName: If w occurs in FNL, LNL, and KNL respectively and w is not in FWL, then the corresponding feature is set to 1. This is because we found some special last names can also be found on FWL such as From, To, Sender, etc. These words are actually used in places other than names more frequently.
- IsPrevName, IsNextName: Names usually appear in a group of two or three words, so the if the adjacent word are names, then the probability of being name for the current word could be increased.

Based on the special fields and structure of Emails and Newsgroup article, we construct some features within a sentence scope.

³ <http://www.census.gov/genealogy/www/freqnames.html>.

⁴ <http://l2r.cs.uiuc.edu/~cogcomp/>

⁵ <http://l2r.cs.uiuc.edu/~cogcomp/>

- **StartsWit-**
From/To/Bcc/Cc/Send/Sender/In-Reply-to/Followup-to/R
eply-to: Lines starting with these function words have a
high probability of containing true names of receivers or
senders. For example, “From:
nstramer@supergas.dazixco.ingr.com (<true_name> Naf-
taly Stramer </true_name>)” and “Sender:
nstramer@supergas (<true_name> Naftaly Stramer)
</true_name>.”
- **StartsWithSubject:** Lines starting with “Subject:” some-
times contain names. Also this feature can indicate the
corresponding zone word w belongs to in an email.
- **AfterForwardedBy:** “Forwarded by” is a format in emails.
As observed, in many cases, words appear after “forwarded
by” are true names. For example, “Forwarded by
<true_name> Christopher McKey </true_name>
/FRA/ECT on 29/03/2001 17:48”.

V. CORPUS AND SETUP

The Email corpus we use were extracted from the Enron cor-
pora by Klimt and Yang[9]. They were further categorized
into two subsets and annotated by Minkov et al.[8]. The first
subset, EnronMeetings, consist most meeting-related email
and messages. The emails in this set are much more relative to
each other. The second subset, EnronRandom, was formed by
repeatedly sampling a user name (uniformly at random among
the 158 users) and then sampling an email from the user
(uniformly at random). Therefore, the emails in this set are
much more independent.

The newsgroup postings were collected from 20 newsgroups
[10]. This collection is a subset of the corpora known to con-
tain complex signature. It was collected by Vitor Carvalho as
part of a related project on learning to extract email signatures
[11].

Corpus	Enron-Meeting	Enron-Random	NewsGroups
# Train	729	516	477
# Test	247	164	120
# Tokens	204,423	285,652	300,177
# Name	2,868	5,059	2,885
# Name Per Email	3.0	7.4	4.8

We first preprocess the Email and Newsgroup corpora by
eliminating sentences where it is impossible for names to
appear in. For example, sentences start with “Message-ID:”,
“Article-ID:”, “Content-Transfer-Encoding”, “Date:”, and etc.
The elimination can be helpful because less noise is introduced
in the training and testing data. Due to the fact that sentences in
emails and newsgroup postings are constructed less formally
and segmentation of sentences is more causal, we use a sen-
tence boundary detector to find the more accurate boundary of
a sentence. Then we feed the preprocessed sentences to POS

tagger to obtain its POS labels. To avoid repeating POS tag-
ging, we output the modified sentences along with its POS
tagges to files. A piece of tagged sentence is shown below:

(From NNS)
(Subject NNP)(Mtg NNP)(w NN)(<true_name> NN)(Neil
NNP)(Hong NNP)(</true_name> NN)(Re NNP)(Role
NNP)(Direction NNP)(or CC)(Organization
NNP)(<true_name> NN)(Tasha NNP)(</true_name>
NN)(x39526 CD)
(requested VBN)(by IN)(<true_name> NN)(Rogers
NNP)(Herndon NNP)(</true_name> NN)(e JJ)(mail NN)(8
CD)(7 CD)(01 CD)

Our experiment is to extract different kind of feature combi-
nations from training data. Then, we use these features to train
Maximum Entropy Classifier and Conditional Random Model.
Finally, we use the testing data to evaluate the performance.

VI. RESULTS & ERROR ANALYSIS

In order to find efficient and effective features used for Email
message and NewsGroups posting. We tried different combi-
nation of the features above. All metrics are taken at a token
level.

Feature 1: include all name list related features.

Feature 2: include all name list related features plus isNNP.

Feature 3: include all POS tag related features plus POS related
heuristics

Feature 4: StartsWithFrom, AfterForwardedBy, isCommon,
hasUnigram, has Bigram, IsNNP, Names list related features,
isFirst, Current Word, character-based features.

Feature 5: all features except character-base n-gram and word
length.

Feature 6: Name list related features, isCommon, charac-
ter-based features, isFirst, isCap, pos tags of w-2, w-1, w, and
w+1, word w-2, w-1, w, w+1.

All: all features

	Acc	P	R	F1	# features
Feature 1	0.9448	0.8406	0.6350	0.7235	15
Feature 2	0.9593	0.8815	0.7421	0.8058	17
Feature 3	0.8929	0.6519	0.1253	0.2102	131
Feature 4	0.9705	0.8991	0.8345	0.8656	7459
Feature 5	0.9714	0.9009	0.8406	0.8697	8224
Feature 6	0.9750	0.9011	0.8759	0.8883	13487
All	0.9736	0.8989	0.8650	0.8816	13524

Table 1. Apply MaxEnt to EnronMeeting

	Acc	P	R	F1	# features
Feature 1	0.9566	0.6677	0.6341	0.6505	15
Feature 2	0.9624	0.7388	0.6327	0.6816	17
Feature 3	0.9394	0.7327	0.0768	0.1391	142
Feature 4	0.9804	0.8511	0.8395	0.8453	16676
Feature 5	0.9844	0.8866	0.8652	0.8758	26201
Feature 6	0.9840	0.8768	0.8705	0.8736	35986
All	0.9856	0.8839	0.8903	0.8871	36036

Table 2. Apply MaxEnt to EnronRandom

	Acc	P	R	F1	# features
Feature 1	0.9640	0.6912	0.2313	0.3466	15
Feature 2	0.8709	0.7317	0.4654	0.5689	17
Feature 3	0.9600	0.5709	0.1374	0.2215	145
Feature 4	0.9811	0.8297	0.6834	0.7495	16052
Feature 5	0.9843	0.8867	0.7118	0.7897	24853
Feature 6	0.9849	0.8722	0.7441	0.8031	34362
All	0.9852	0.8767	0.7479	0.8072	34417

Table 3. Apply MaxEnt to NewsGroups

	Accuracy	Precision	Recall	F1
EnronMeeting	0.9610	0.8771	0.7640	0.8167
EnronRandom	0.9676	0.7527	0.7313	0.7418
NewsGroups	0.9740	0.7481	0.5579	0.6391

Table 4. Apply CRF to three different copra.

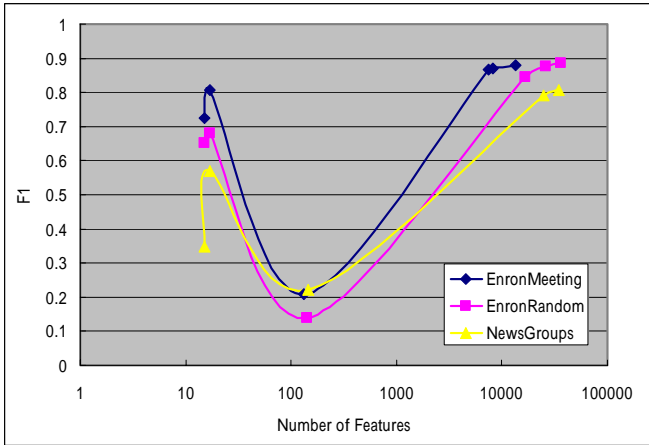


Figure 2, Number of Features v.s. F1

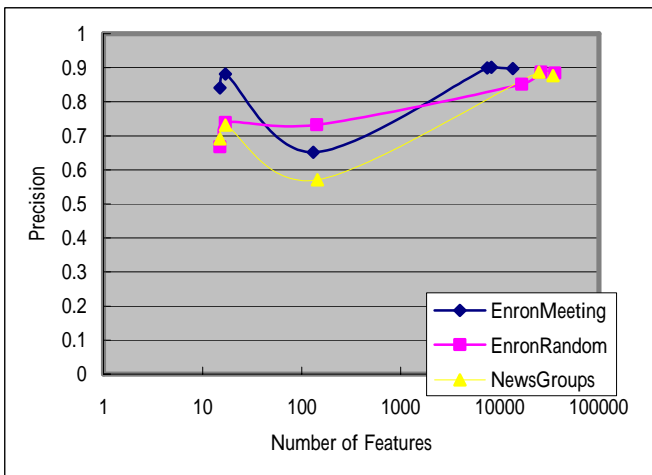


Figure 3, Number of Features v.s. Precision

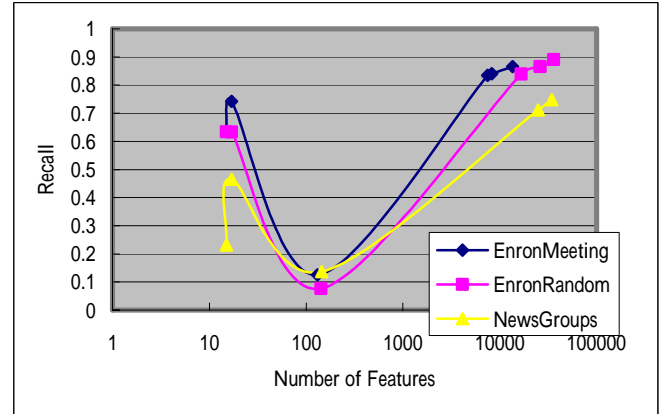


Figure 4, Number of Features v.s. Recall

From Figure 2, 3, and 4, we can find a general trend that the more features included, the better the F1, precision, and recall measures are. The best F1 measures we achieve for EnronMeeting, EnronRandom, and Newsgroup are 88.83%, 88.71%, and 80.72% respectively.

One thing to notice is that feature combination 2 which is composed of feature combination 1 and IsNNP has only 2 more features than feature 1. But, the F1 measure is increased by 22% in NewsGroups, 8% in EnronMeeting, and 2% in EnronRandom. Therefore, we can conclude IsNNP is a very effective and efficient feature to include in name recognition. This result agrees with the fact true names are labeled as NNP in the most cases.

However, the POS related features alone yield a very low F1 measure which mostly comes from low recall. This is observed from the similarity between Figure 2 and Figure 4. Feature combination 3 includes the POS tags for two previous words, current word, and next word and some POS related heuristics such as IsNNP, PRPAnd, and IsVBAround. The extremely low recall can be explained by the fact that in the informal texts, it is very likely that true names do not appear in a complete sentence. For examples, names appear in the sender and receiver fields and the non-trivial signatures. Given only the POS tags and POS related heuristic, names in those fields are hard to be recognized. Therefore, low recall is expected.

Feature combination 4, 5, and 6 are constructed by adding features that we think may be influential to true name recognition. The results from these combinations, despite of fewer features, are comparable to the result by including all features. Especially, feature combination 6 actually has better F1 than including all features by both higher precision and recall for EnronMeeting corpus.

Both Maximum Entropy Classifier and CRF perform the best in EnronMeeting, second in EnronRandom, and as expected worst in Newsgroup. This can be explained by the characteristics of each corpus. Because most emails in En-

ronMeeting are meeting-related, the formats of emails are more consistent than formats of EnronRandom and Newsgroup. Especially in the Newsgroup corpora, complex “signatures” appear more often and the content of each article varies. For example, a portion of newsgroup messages actually contains a piece of program. This makes the true name recognition even more difficult because of the noises those irrelevant parts bring in.

The result of CRF is presented in Table 4. The best F1 measures from CRF for Enron Meeting, Enron Random, and Newsgroup are 81.67%, 74.18%, and 63.91%, respectively. The results from CRF are not better than MaxEnt classifier. In terms of training time, it takes a lot longer to train CRF classifier than MaxEnt classifier. Therefore, MaxEnt does a better job than CRF in terms of performance and training time for the email and newsgroup corpora we experimented on.

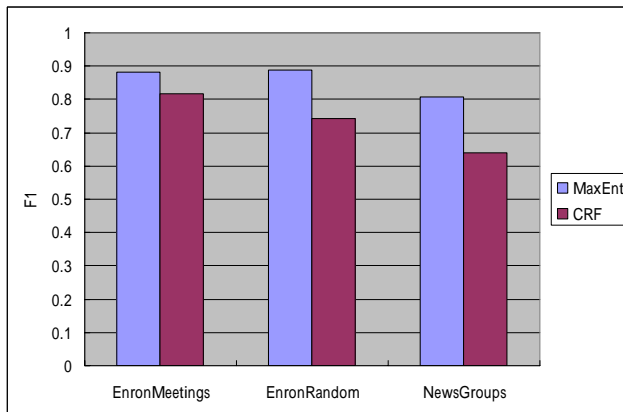


Figure 5, Compare between MaxEnt and CRF

Besides, we found some true names labels in the corpora are incorrect since the annotation is labeled manually. For example, “Attached is an outline of the presentation, including the proposed procedures. -- <true_name> Bob </true_name> Robert E. Bruce Senior Counsel.” Rober E. Bruce should be included inside the true name tags. Also, in some newsgroup articles, punctuations and symbols in complex signature are labeled as true name. From our misclassification log, a portion of misclassified instances actually come from the incorrect true name labels. Therefore, we believe the performance of our classifier could be better if those erroneous labels are corrected.

In addition, since the POS tagger we use to preprocess the emails and newsgroup postings are trained on WSJ, not on informal texts, there is a portion of POS tags are incorrect which definitely degrade the performance of the classification when the POS tags are used as features. Thus, with a POS tagger which is specifically trained on informal text, the performance of our name recognizer can be improved.

From the result of our experiment, we find in the most cases, recall is lower than precision, and thus lower the F1 metric. To improve the relatively low recall, we can try to include more global features that contain the information throughout a single message or across the multiple messages instead local features that is constrained in a sentence. For example, a name may occur multiple times in a corpus, especially in email corpus which is associated with a group that works closely together. Therefore, information of a name’s repetition could be beneficial to recognize the one in an ambiguous context.

VII. CONCLUSION

This project applies MaxEnt classifier and CRF classifier to informal text which is prepared quickly and to a narrow audience. With MaxEnt classifier, we achieve the best F1 measure 88.83% with features built for both informal and formal texts.

Reference:

1. Hai Leong Chieu and Hwee Tou Ng, Named Entity Recognition with a Maximum Entropy Approach. In: *Proceedings of CoNLL-2003*, Edmonton, Canada, 2003, pp. 160-163.
2. Oliver Bender, Franz Josef Och and Hermann Ney, Maximum Entropy Models for Named Entity Recognition In: *Proceedings of CoNLL-2003*, Edmonton, Canada, 2003, pp. 148-151.
3. Andrew McCallum and Wei Li, Early results for Named Entity Recognition with Conditional Random Fields, Feature Induction and Web-Enhanced Lexicons. In: *Proceedings of CoNLL-2003*, Edmonton, Canada, 2003.
4. James Mayfield, Paul McNamee and Christine Piatko, Named Entity Recognition using Hundreds of Thousands of Features. In: *Proceedings of CoNLL-2003*, Edmonton, Canada, 2003, pp. 184-187.
5. Radu Florian, Abe Ittycheriah, Hongyan Jing and Tong Zhang, Named Entity Recognition through Classifier Combination. In: *Proceedings of CoNLL-2003*, Edmonton, Canada, 2003, pp. 168-171.
6. Dan Klein, Joseph Smarr, Huy Nguyen and Christopher D. Manning, Named Entity Recognition with Character-Level Models. In: *Proceedings of CoNLL-2003*, Edmonton, Canada, 2003, pp. 180-183.
7. Casey Whitelaw and Jon Patrick, Named Entity Recognition Using a Character-based Probabilistic Approach. In: *Proceedings of CoNLL-2003*, Edmonton, Canada, 2003, pp. 196-199.
8. Einat Minkov, Richard C. Wang, and William W. Cohen, Extracting Personal Names form Emails: Applying Named Entity Recognition to Informal Text
9. Klimt, Bryan and Yiming Yang, 2004. Introducing the Enron Corpus. In *Proceedings of the Conference on Email and Anti-Spam 2004*, Mountain View, California.
10. Craven, M., D. DiPasquo, D. Freitag, A. McCallum, T.

- Mitchell, K. Nigam, and S. Slattery. 2000. Learning to construct knowledge bases from the world wide web. *Artificial Intelligence*, 118(1-2):69-113.
11. Carvalho, Vitor and William W. Cohen. 2004. Learning to extract signature and reply lines from email. In *Proceedings of the Conference on Email and Anti-Spam 2004*, Mountain View, California.