Email Annotation: Extracting Names Using Named Entity Recognition

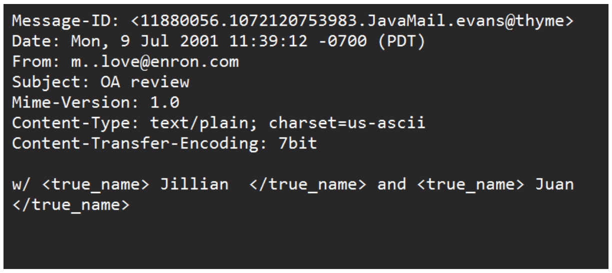
Iman Adetunji [*] **Beamlak Bekele** [*]

1. Introduction

For this project, we aim to extract names from a dataset of informal text using Named Entity Recognition. Named Entity Recognition (NER) is the task of identifying and categorizing key information (entities) in text. An entity can be any word or series of words that consistently refer to the same thing, such as a person, organization, time, or location. Examples of use cases of NER include resume filtering for jobs, as NER is used to identify the necessary skills of a candidate, and customer support, where NER is used to classify incoming requests based on the customer's needs.

1.1. Problem Definition

The problem for this project is to train our model to recognize names in informal text, such as emails. To accomplish this, we used the Enron corpus data, which contains over 700 emails, consisting of the subject line, body, and sender information, as shown below. Our goal is to replicate and improve the results from previous experiments (Minkov et al., 2005). The data was both annotated and in XML format. We also used a framework called Conditional Random Field (CRF) to help us, which is a probabilistic framework for labeling and segmenting structured data. A linear chain CRF's input data in this project included, the set of input vectors X, the position i of the data point we are predicting, the label of the data point i - 1 in X, and the label of the data point i in X. The purpose of this function is to express some kind of characteristic of the sequence that the data point represents. In the case of our project, the feature function represents the words in the emails within our dataset. To build the conditional field, we need the probability distribution, which the equation for is shown below. To estimate the parameters, Maximum Likelihood Estimation is used.



```
Message-ID: <11880056.1072120753983.JavaMail.evans@thyme>
Date: Mon, 9 Jul 2001 11:39:12 -0700 (PDT)
From: m..love@enron.com
Subject: OA review
Mime-Version: 1.0
Content-Type: text/plain; charset=us-ascii
Content-Transfer-Encoding: 7bit

w/ <true_name> Jillian  </true_name> and <true_name> Juan
</true_name>
```

Figure 1. Enron dataset example

$$P(y, X, \lambda) = \frac{1}{Z(X)} exp\{\sum_{i=1}^{n} \sum_{j} \lambda_j f_i(X, i, y_{i-1}, y_i)\}$$

$$\text{Where: } Z(x) = \sum_{y' \in y} \sum_{i=1}^{n} \sum_{j} \lambda_j f_i(X, i, y'_{i-1}, y'_i)$$

Figure 2. Probability Distribution of CRF

2. Proposed Methods

(Minkov et al., 2005) broke the problem in to two pieces: evaluate the performance of CRF and exploiting the repetition of names. The later method was implemented to take advantage of the repetition of names in email corpus. Our experiments concentrated on tracking and improving the implementation of CRF. The paper run experiments on 4 databases.

You can train Named Entity Recognition with two different methods: through Conditional Random Field (CRF) or Hidden Markov Model (HMM). Conditional Random Fields are a discriminative model that can be used for predicting sequences. They do so by using the contextual information from previous labels to make their predictions. Their underlying principle is that they apply logistic regression on sequential inputs.

Hidden Markov Models (HMMs) are probabilistic frameworks where the observed data are modeled as a series of outputs generated by one of several hidden internal states. HMMs are built on the assumptions that output observation is conditionally independent of all other hidden states and all other observations when given the current hidden state, and that the probability of seeing a specific observable given a hidden state (Fosler-Lussier).

[*] Equal contribution . Correspondence to: Iman Adetunji <imana4@umbc.edu>, Beamlak Bekele <pg41777@umbc.edu>.

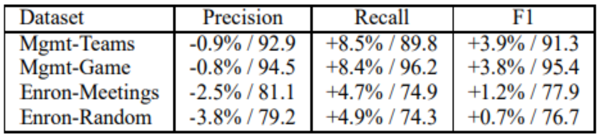
Accepted to the *1st Workshop on Generative AI and Law*, co-located with the *International Conference on Machine Learning*, Honolulu, Hawaii, USA. 2023. Copyright 2023 by the author(s).

Even though Conditional Random Fields and Hidden Markov Models are both used to model sequential data, they are different algorithms. Hidden Markov Models are generative, and give output by modeling the joint probability distribution (Fosler-Lussier). But Conditional Random Fields are discrimnative, and model the conditional probability distribution. HMMs are based on Naive Bayes, and CRFs are derived from logistic regression.

We chose to use Conditional Random Field for this project because of its conditional nature. This means that there is a relaxation of the independence assumptions that is required by HMMs (John Lafferty, 2001). They also avoid the label bias problem, a problem shown in conditional Markov models.

3. Intuition

We think that Conditional Random Field is the best method to work for our project based on experiments done with Named Entity Recognition with another paper. In our reference paper, CRF was flexible enough to substantially improve the performance of extracting names from informal text . In their experiments, they we able to improve the F1 performance in both the Mgmt-Teams dataset and the Mgmt-Game dataset (Minkov et al., 2005), as shown in the image below.



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

Figure 3. Table of results from Minkov paper

Given these results, we are expecting a similar or improved result with this paper.

4. Experiments

4.1. Data

Our reference papers (Minkov et al., 2005) used 4 email corpus: Mgmt-Teams, Mgmt-Games, Enron-Meetings, and Enron-random. Mgmt-Teams and Mgmt-Games are behind a paywall. We used two types of corpus Enron Meeting and Enron Random. The annotated data downloaded contained 2 folders, train, which obtained 80% of the data, test, containing 20% of the data. The Enron Meeting training data contains almost 800 names and the test contained more than 200 names. The Enron Random Corpus training data contained almost 1087 annotated names and 387 true names.

In the Enron meeting corpus, the most common word included articles like “the” and helping verbs like “is”, “meeting”, and “please”, proving that the corpus contained communication about meetings. Given the large amount, we decided not to use all the data, since it would take ample time to run multiple experiments and we don’t have computer resources. Table 1 shows the number of files used to train and test the model, each file is an email.

Table 1. The corpora used in the experiment

CORPUS	TRAINING	TESTING	TOTAL
ENRON MEETING	244	247	729+247
ENRON RANDOM	360	110	516+164

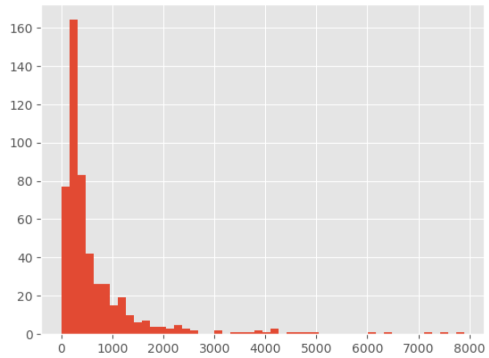


Figure 4. The histogram provides a visual representation of the distribution of string lengths in the Enron-Random. The x-axis shows the ranges of string lengths, and the y-axis shows how many strings have lengths within each range.

The first step taken was to clear the data. An Enron email file (*Figure 1*) contains message-id, from, subject, and date. We used the Panda library to create a data frame that has 10 columns: Employee, email folder, message-id, date, from, subject, body, chain, signature, and email path. The main column used to train the model is the body, with the plan to add a signature and subject as an additional feature in the future.

4.2. Features

There are 3 types of features we worked on: Basic, Dictionary, and Email features. In the beginning, our extended basic features included a part-of-speech (POS) tag. That resulted in our F1 score being 100%, creating too much noise. Agreeing with the (Minkov et al., 2005) paper, we removed it. In the section below, the variable *t* represents the focus token. We also extract a similar feature of the next token, *t+1*. If the word is at the end of the document, we tag it as **EOF**. The basic feature has two parts.

Experiment 1: Basic Feature extracted using just the training data without additional resources.

Part 1:

1. token t in lowercase
2. Boolean to indicate if t is fully capitalized

Extended features - Part 2:

1. Token t istitle(): Boolean is set to 1 if the first letter of t is capitalized
2. The last three letters of t: example "hello" word[-3:] = "'llo"
3. The last two letters of t: example "hello" word[-2:] = "'lo"

Experiment 2: Dictionary Features

1. Name Title: We compiled a list of possible titles common in the workplace that could be placed before a name. Eg. "Mr.", "CEO", "Mrs"
2. First name List (4000 fNa): Most common first name in the US
3. Last name List (4000 last name): Most common Last name in the US

We weren't able to access the link in the paper that referred to a file containing the most common names in the US published by the US Census Bureau. Therefore, we found a GitHub repository containing a total of 8000 most common first and last names in the US.

Experiment 3: Email Features

1. t appears in the header: Boolean that will be 1 if the t is in the Subject of the email
2. t appears in the "from" field: Boolean set to one if the t has been mentioned from filed
3. t is a probable "signoff": Boolean ser ro one if t has occur in a signoff

Subject, from, and sign-off sections of emails have a high possibility of containing the true names of receivers or senders. We hope the features can improve true name recognition.

Data	Enron-M	Enron-R
Precision	0.81	0.88
Recall	0.45	0.53
F1-score	0.58	0.66
F1 from paper	0.59	0.68

Table 2. Basic Feature result: for Enron Meeting and Enron Random data set. F1 from paper results from reference paper

Features	B	B+B	B+D	B+E
Precision	0.81	0.90	0.87	0.88
Recall	0.45	0.67	0.56	0.54
F1	0.58	0.77	0.68	0.67
F1	0.59	**X**	0.78	0.71

Table 3. All experiment result, Enron-Meeting data set

4.3. Experiment Details and Observations

We evaluated the efficiency of the model using precision, recall, and F1 score. **Precision** is how good the model is at predicting a specific category. **Recall** tells you how many times the model was able to detect a specific category. The **F1 score** is the harmonic mean of precision and recall. It provides a balance between precision and recall, considering both false positives and false negatives.

$$P = \frac{True\ postive}{(True\ postive\ +\ Fasle\ Postive)}$$

$$R = \frac{True\ postive}{(True\ postive\ +\ Fasle\ Negative)}$$

$$F1 = 2 * \frac{Precision \times Recall}{(Precision \times Recall)}$$

Figure 5. Results equation

Experiment 1's initial result was shown during our presentation. Additional experiments were done to improve the results. While training the model, the L1 parameter was 0.1, L2 regularization was set to 0.01, and the maximum number of iterations was constant at 70.

When training the model for our presentation, our maximum iteration parameter was 40. To improve results, the iteration size has increased. As predicted, increasing iteration improved the result by 16%. Table 2 contains recalculated results collected from basic features, experiment 1 part 1.

The table shows the precision, recall and F1 performance of CRF-trained models for both datasets, using the basic features alone (B); basic and extended basic features (B +B); the basic and email features (B+E); the basic and dictionary

Features	B	B+B	B+D	B+E
Precision	0.88	0.91	0.88	0.84
Recall	0.45	0.67	0.57	0.67
F1	0.58	0.76	0.69	0.75
F1	0.59	**X**	0.72	0.70

Table 4. All experiment result, Enron-Random data set

features (B+D). It wasn't possible to do combined B+E+D since it takes a lot of time to extract dictionary features.

The result shows that recall is usually the lowest score and this trend continues throughout the other experiment. It is also evident that the results from the reference paper and our results are close. Another observation is that the Enron-Random data set has better results. This may be because the Enron-Random data set contains more true name instances than the Enron-Meeting.In future experiments to improve our results we can filter our dictionary to reduce false positives.

5. Conclusion

In this paper, we try to train a Conditional Random Field Named Entity Recognition Model with two datasets, Enron-Meeting and Enron-Random. In doing so, our experiments show that the recall for each one was consistently the lowest number, but our results compared to our reference paper were very close. Overall, the Enron-Random dataset had better results than Enron-Random. In our experiment on the Enron-Meeting dataset, using basic and email features, we were able to increase the precision by 0.07, the recall by 0.01, and the F1 score by 0.01. In our experiment on the Enron-Random dataset and using basic and email features, the precision had decreased by 0.04, but we were able to increase the recall by 0.14, and the F1 score by 0.09.

References

Fosler-Lussier, E. Markov models and hidden markov models: A brief tutorial.

John Lafferty, Andrew McCallum, D. P. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *In Proceedings of the Eighteenth International Conference on Machine Learning*, 2001.

Minkov, E., Wang, R. C., and Cohen, W. W. Extracting personal names from email: Applying named entity recognition to informal text. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pp. 443–450, Vancouver, British Columbia, Canada, 2005. Association for Computational Linguistics.