Named Entity Recognition with Extremely Limited Data

John Foley, Sheikh Muhammad Sarwar, and James Allan
Center for Intelligent Information Retrieval
CICS, University of Massachusetts Amherst
[jfoley, smsarwar, allan]@cs.umass.edu

ABSTRACT

Traditional information retrieval treats named entity recognition as a pre-indexing corpus annotation task, allowing entity tags to be indexed and used during search. Named entity taggers themselves are typically trained on thousands or tens of thousands of examples labeled by humans.

However, there is a long tail of named entities classes, and for these cases, labeled data may be impossible to find or justify financially. We propose exploiting named entity recognition as a search task, where the named entity class of interest is a query, and entities of that class are the relevant “documents”. What should that query look like? Can we even perform NER-style labeling with tens of labels? This study presents an exploration of CRF-based NER models with handcrafted features and of how we might transform them into search queries.

1 INTRODUCTION

Consider a reporter, researcher, or policy maker interested in the Syrian refugee crisis, who has found an article discussing how Malta, as a country, has been impacted by the influx of refugees. Suppose this user now wants to find information that details how other island-nations in the Mediterranean have been affected. One technique that could help satisfy this information need is named entity recognition (NER). If a tagger had marked countries and islands or (better yet) Mediterranean islands and they were indexed by the system, the searcher could construct a query targeting documents that had mentions of refugees and islands (or Mediterranean islands). Why, though, would islands have been among the obvious taggers to construct in advance? Would someone bother to build a tagger for Mediterranean islands?

Collecting this kind of data may be difficult or impossible for a domain expert. Sequence labeling tasks like this require fine-grained labels and experienced annotators.

Query-log or knowledge-base analysis would allow one to derive a set of entity classes that are of interest, but entity types would have the same long-tail problem as language: there would always be classes missing. Even without approaching the long-tail, it is fairly simple to come up with entity classes that do not quite fit the traditional four-class paradigm (PER, LOC, ORG, other), such as musical genre, which is obviously none of the above, or cigarette brand which while being related to company or ORG is definitely a distinct entity class. As an experiment, we labeled a number of entities that would have been helpful for TREC QA questions, and found a wide variety of entities, from the common country and city to the rare: black panther members, airlines, sub-atomic particles and comets. Coming up with a comprehensive set of classes a priori seems destined to fail.

Instead, we propose an approach that treats named entity detection as a retrieval task, one performed at query time rather than in advance while indexing. We do not propose this as a replacement for NER, but as something to be used for an ephemeral or contextual class of entity, when it does not make sense to label hundreds or thousands of instances to learn a classifier. We will show that it is possible to achieve reasonable effectiveness using this approach, though it does require a substantially larger index. We will further show that this approach can respond rapidly, an important validation since we are moving a document pre-processing step to query time.

We are interested in transforming the recognition task into a search task and we refer to the task as Named Entity Search (NES). We do that by considering each word occurrence as a document and the words’ features as “terms”, allowing us to use traditional search engines in order to score words in response to a query, where the query is the entity class of interest.

In this work, we explore four core research questions:

(1) How do we translate CRF-based NER models with handcrafted features into a retrieval model where tokens are documents? (See §3 and Table 1.) By focusing on models with handcrafted features, our users could potentially edit or provide feedback on the features in our short queries [25].

(2) Can we produce a reasonable ranking over tokens for the user to interact with sentences when we only have a handful of labels to create an NER-classifier? (See Figure 1.)

(3) Can we make our NER models small enough that they can be efficiently executed by a search engine without loss of effectiveness? (See Figure 2.)

(4) What typical handcrafted NER features are most useful for our ad-hoc variant of NER? (Results in Table 5.)

We present experiments and analysis for these research questions, and begin by discussing related work.

2 RELATED WORK

Named entities have a long history of being important and useful for information retrieval tasks. Recently, work has focused on connecting entities present in text to knowledge bases to allow for improved reasoning [8, 31]. Entities have also been at the center of a large amount of research, including the entity recognition
and disambiguation challenge [5], the TREC temporal summarization track [1], the TREC knowledge-base population track [20], and the entity ranking tracks at INEX [11] and TREC [3]. Entities, specifically people with emails, were also the highlight of the expert-finding task within the TREC enterprise track [7]. Almost all of these problems and domains depend on entity recognition as a document-enrichment step.

In natural language processing, named entity recognition (NER) and part-of-speech tagging (POS) are considered examples of the sequence labeling problem. One of the common evaluations for this task is the CoNLL-2003 shared task [29], which introduced English and German datasets derived from expert-labeled news articles. Successful NER systems use a large number of features, including gazetteers, case information, word cluster ids, and part-of-speech tags. A 2007 survey by Nadeau and Sekine [23] provides more information about the origins of this task and traditional approaches. A recent work that discusses the challenges of generalization for modern NER approaches [2] may be an additional resource in lieu of a more recent survey.

2.1 Recent Approaches in NER

Most state-of-the-art NER approaches are now based on neural networks and deep learning. While using neural networks for NER is not new (one of the CoNLL 2003 submissions used a LSTM network [12, 29]), the availability of more memory and more data has made these approaches the dominant research direction. Recent advances in deep neural networks have led to a reduction in the number of hand-tuned features required for achieving state-of-the-art performance.

However, since we expect our task to have very few labels, and we hope to minimize the complexity of our query models. Our work includes the "distributional similarity features" included in Stanford NLP, which are Brown clusters [4, 16]. In future work, we hope to explore explicitly learning neural representations from unlabeled data, e.g. Lample et al [14].

2.2 Many-classed NER

Some work explores more fine-grained representations of entities, namely a work by Lee et al. which explores a model for identifying named entities as a single step and classification into 147 different categories as a secondary step [15]. More recently, Ling and Weld present FIGER, a system for fine-grained entity recognition of 112 tags, trained using data collected from Wikipedia links used a similar approach: a CRF for segmenting into B (beginning), I (intermediate), and O (outside) tags, and a multiclass classification task on top [18]. Lin et al. assign more than 1000 types to noun-phrases to help classify out-of-knowledge-base entities [17]. In order to be as general as possible, we do not consider filtering to noun-phrases or classified spans in this work, but that is a promising approach to increasing efficiency in future work.

2.3 List Completion and QA

Two existing tasks are similar to NES in terms of their goal: generating a list of entities based on user intent. However, neither explores a limited-data setting.

2.3.1 Question Answering Systems

First, question answering systems often leverage named entity recognition as a method of collecting potential answers [15, 22], for both factoid QA as well as list QA [10, 30]. However, historically, some of the best systems required intensive data resources that do not make sense for our ad-hoc task. The best system from 2007 used coverage from the lists available in Wikipedia, ahead-of-time construction of target NER classes, and sophisticated textual analysis based on logic systems [21]. Another consistently high-ranked system used a tiered response system that first checked a database built from "info boxes," then Wikipedia lists and tables, then a set of data from lexicons, and even web search hit counts [13]. Often, systems targeting factoid queries merely repeatedly executed their factoid method. Because of our reluctance to limit potential answers to those in external resources and our desire to incorporate interactivity we do not compare directly to any TREC-QA systems in this work. However, we do make use of the list questions in the TREC 2005 and 2006 Question Answering track [10, 30].

2.3.2 List Completion

List completion exists in various forms, with some work inspired by the no-longer-running Google Sets [9]. However, this work presumes that data mining methods will be effective: that entities of interest occur in HTML or Wikipedia lists with some regularity. A similar task was explored at the INEX list completion task [11], which was based on selecting the best answers from a fixed Wikipedia corpus. In the language of existing List Completion approaches, we could call our task list completion over unstructured text, but we are not aware of any similar work.

2.4 Active Learning

Active Learning is a broad field that arose from the desire to have machine learning algorithms select a subset of the training data to learn the best classification function, but the tools and techniques in this field are applicable to interactive labeling tasks as well. Settles presents a comprehensive survey of this subfield [26].

In the realm of NER, active learning has specifically been used to minimize human annotation efforts. Shen et al. show that only 20% of labels are truly required to achieve strong performance with a good selection function [27]. Rather than minimizing human interactions to achieve a final high-accuracy classifier, our named entity search task focuses on extremely limited interaction and high-precision effectiveness.

3 METHODS

The core of our interactive Named Entity Search is the ability to leverage user-feedback in order to quickly and interactively construct useful named entity tags. We briefly discuss conditional random field, as the dominant approach to non-neural NER as it is more suitable for our fewer label, interactive setting. And then we talk about the user interaction model.

3.1 CRF as a Retrieval Model

The retrieval model specifies how retrievable items (tokens represented by features) are ranked in response to the query (weights of features). Based on the success of CRFs, we start with its calculation of the probability that a token \( t \) has a label \( y_t \). We use the notation of Sutton and McCallum [28]:
\[
p(y_t|x) = \frac{1}{Z} \exp \left( \sum_{k=1}^{K} w_k f_k(y_t, y_{t-1}, x_t) \right)
\]

To make this useful for search, we first apply some standard IR transformations to this model. We intend to rank tokens consistent with this probability rather than calculating it directly.

Because we are interested in ranking, rather than exact probabilities, we can perform a number of typical rank-safe transformations: removing the \(\frac{1}{Z}\) normalization, and taking the log of each side. We cannot remove the dependency of \(y_t\) on \(y_{t-1}\) with pure mathematics, so we turn to experimentation to determine whether we can simplify this expression further and to quantify the loss of accuracy.

We also validate empirically that with linear CRF models, especially with fewer labels, the transition probabilities are of little benefit. Table 1 shows the results of downgrading a full CRF to doing a single class at a time and then removing transition probabilities on a standard NER dataset. By ignoring the transition ans sequence passes of learning and training classes independently, we have achieved a linear query model and measured the difference between that and the more typical CRF.

\[
p(Q|x) = \sum_{k=1}^{K} Q_k f_k(x) = \tilde{Q} \cdot \tilde{f}(x)
\]

This means that we treat each token in a collection as a document, and it is represented by a bag of traditional NER features (as extracted by Stanford NLP [19]). This means that we can effectively execute our NER models with an IR system that supports this sort of vector-space model.

Table 1: Classification Results on our NER Dataset. Measures are the \(F_1\) of token classification experiments to explore the effect of our new assumptions.

<table>
<thead>
<tr>
<th>System</th>
<th>PER</th>
<th>LOC</th>
<th>ORG</th>
<th>MISC</th>
<th>Micro</th>
<th>Macro</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full CRF</td>
<td>86.0</td>
<td>85.6</td>
<td>91.1</td>
<td>61.9</td>
<td>86.5</td>
<td>81.1</td>
</tr>
<tr>
<td>Query independence</td>
<td>85.2</td>
<td>87.3</td>
<td>90.5</td>
<td>59.2</td>
<td>86.3</td>
<td>80.6</td>
</tr>
<tr>
<td>Token independence</td>
<td>84.5</td>
<td>86.4</td>
<td>89.8</td>
<td>63.0</td>
<td>85.9</td>
<td>80.9</td>
</tr>
</tbody>
</table>

3.2 User Interaction

In early feedback, we discovered that returning tokens or spans to the user was not sufficient to understand whether the entities being discovered were correct, so we built our system around a sentence-feedback model, where users are presented a sentence at a time (which contains the highest ranked tokens) and users label the entire sentence before the system re-ranks and presents them with the next best sentence. In this way, a user works to solve their problem (collecting a list of entities) while generating a small amount of training data suitable for NER software [24].

4 DATASETS

We explore questions related to NES using a well-labeled NER dataset on which a classification task makes sense. Our fundamental research questions regarding whether NES is practical and effective are done using a novel QA-List dataset.

4.1 NER

We leverage the English dataset from the CoNLL-2003 shared task. Table 2 provides some statistics about that collection. It consists of 946 documents for training, 231 documents for validation, and 231 documents for testing. Our approach does not need to train parameters, so we did not use the validation set. In order to compare to previously published results, we choose to ignore that portion of the data rather than mix it into the training or test sets.

4.2 QA-List

In order to explore research questions that surround sparse entity types, we have adapted a dataset from the TREC Question Answering Track. We select data from the 2005 and 2006 challenges because they operate on the same dataset (AQUAINT) and the full set of annotator judgments are available for all surface forms reported by all participants in the challenge. Some statistics about AQUAINT in comparison to the smaller NER dataset are presented in Table 2.

Table 2: Comparison and Summary of Datasets.

<table>
<thead>
<tr>
<th>Collection</th>
<th>NER</th>
<th>QA-List</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source</td>
<td>CoNLL 03</td>
<td>AQUAINT</td>
</tr>
<tr>
<td>#docs</td>
<td>1408</td>
<td>1,088,791</td>
</tr>
<tr>
<td>#words</td>
<td>203,621</td>
<td>488,789,688</td>
</tr>
<tr>
<td>#uniq words</td>
<td>20,386</td>
<td>1,113,614</td>
</tr>
<tr>
<td>Inv &amp; Fwd Index</td>
<td>0.002G</td>
<td>2.8G</td>
</tr>
<tr>
<td>NER Features</td>
<td>0.276G</td>
<td>495G</td>
</tr>
<tr>
<td>NER Features Index</td>
<td>0.095G</td>
<td>158G</td>
</tr>
<tr>
<td>CRFSuite Tagging Time</td>
<td>2 seconds</td>
<td>2 hours</td>
</tr>
</tbody>
</table>

Since we are explicitly curating this data to use less-common entity types, we preprocess it to skip the 25 queries that are explicitly seeking lists of countries and the 4 queries that are seeking lists of cities. We also drop queries that have 3 or fewer answers because we want to show learning curves for this task. After preprocessing, we have 66 queries from 2005 and 60 queries from 2006. Some examples can be found in Table 3.

We limit our processing to the provided ranking created by the track organizers. While this limits our recall slightly (there are positive judgments outside these rankings), this prevents our technique from pulling up large numbers of unjudged tokens, and makes our experiments more computationally efficient. In the future, we hope to fully adapt this dataset and collect the additional judgments.

4.3 Unique Average Precision (uAP)

For our person query (PER), we do not want to claim high effectiveness if the only person it tags in the top ranks is "Barack Obama", even if our system is technically correct about the class of all those repeated instances. We define unique average precision (uAP) to be standard average precision, but only after removing all subsequent mentions of any entity. Unlike evaluations of retrieving novel
We explore three interaction models for the NES problem of labeling high-ranking relevant documents.

Changes via interaction. Selecting random documents is useless and these instances will be most informative to the learning algorithm.

Figure 1a. It shows the change in effectiveness (uAP) as the query changes via interaction. Selecting random documents is useless and is easily out-performed by a strategy that labels sentences from high-ranking relevant documents.

We studied the top-10 features selected by each model (weighted by their occurrence across models and by those models’ uAP score), and found the top 5 most important features by domain. These results are presented in Table 5.

The most useful features for the NER domain were word clusters [4, 16] trained on the RCV1 corpus provided pre-trained by Stanford NLP [19]. We briefly explored training1 our own word clusters on the AQUAINT corpus but we observed no significant difference in learning curves. We hope to explore more distributional similarity features in the future.

1https://github.com/percyliang/brown-cluster
Figure 1: Interactive effectiveness on our two datasets.
The NER dataset (right) is deeply labeled, but is less informative for our desired problem as a result of the entity classes being too frequent and too simple. On our new QA-List dataset, however, we can see that the unsure and interactive baselines are equally competitive, which validates our new, interactive approach to NER for more difficult datasets and sparse labels.

Table 5: The top features in our analysis.

<table>
<thead>
<tr>
<th>Rank</th>
<th>NER</th>
<th>QA-List</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Brown Cluster Curr</td>
<td>Words to the Right</td>
</tr>
<tr>
<td>2</td>
<td>POS Tag</td>
<td>Words to the Left</td>
</tr>
<tr>
<td>3</td>
<td>Char N-grams</td>
<td>Char N-grams</td>
</tr>
<tr>
<td>4</td>
<td>Brown Cluster Next</td>
<td>Word Shape Next</td>
</tr>
<tr>
<td>5</td>
<td>Words to the Right</td>
<td>Word Shape Prev,Curr</td>
</tr>
</tbody>
</table>

Figure 2: Effectiveness on QA-List with limited features for efficiency. On this dataset, the learning curve for models limited to 1000 features is nearly indistinguishable from the curve with full models. Limiting model size does not result in a significant performance drop within a few hundred interactions.

6 CONCLUSION
In this study, we explore the possibility of viewing a traditionally offline pre-processing and token classification task as an exploratory search task to deal with extremely limited data. In so doing, we introduce a novel problem that could help analysts, reporters, and other expert users understand and solve their information needs.

We explore effectiveness, modeling, and efficiency while focusing on handcrafted features rather than neural models so that we can explore human-edited queries in the future. We conclude that our approach will be feasible for expert use. In the future, we hope to fuse more research from the active learning, NLP, and IR fields to develop new approaches to this task.

ACKNOWLEDGEMENTS
This work was supported in part by the Center for Intelligent Information Retrieval and in part by NSF grant #IIS-1617408. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsors.

The authors would additionally like to thank Jiepu Jiang, who helped us to extract the NLP features at the core of these experiments.

REFERENCES