Passage Retrieval for Incorporating Global Evidence in Sequence Labeling

Jeffrey Dalton, James Allan, and David A. Smith
Center for Intelligent Information Retrieval
Department of Computer Science
University of Massachusetts Amherst
Amherst, Massachusetts
{jdalton, allan, dasmith}@cs.umass.edu

ABSTRACT

Many forms of linguistic analysis, such as part of speech tagging, named entity recognition, and other sequence labeling tasks are performed on short spans of text and assume statistical dependence within a window of only a few tokens. We propose using passage retrieval to induce non-local dependencies in structured classification that generalizes earlier work in context aggregation for named-entity recognition. We introduce a new method for feature expansion inspired by psuedo-relevance feedback (PRF). Our results on the CoNLL 2003 task show that features from cross-document feature expansion improves NER effectiveness over previous aggregation models. Utilizing all the tokens in a sentence for query context consistently perform best on both intrinsic and extrinsic evaluations. Tagging models incorporating feature expansion outperform the leading NER system when evaluated on out of domain data, a collection of publicly available scanned books on the topic of historic Deerfield, MA. Finally, the results show that retrieval based feature expansion using an external collection of unlabeled text can result in further effectiveness improvements.

Categories and Subject Descriptors

H.3.3 [**Selection Process**]: [Information Search and Retrieval]

Keywords

Named Entity Recognition, Passage Retrieval, Pseudo-Relevance Feedback, Information Extraction

1. INTRODUCTION

Despite the increased application of Natural Language Processing (NLP) on queries and documents to improve retrieval, there is little work exploring the use of retrieval to improve NLP tasks. In this work, we use passage retrieval to improve the effectiveness of Named Entity Recognition (NER). NER is one of many commonly performed sequence labeling tasks including part of speech tagging, syntactic chunking, and other types of information extraction. In these problems, we are given an input sequence of observed variables, $\mathbf{x}$, which consists of a sequence of words. For each observed variable, $x_i \in \mathbf{x}$ the goal is to infer a corresponding output label.

In most statistical sequence models the decision about the output label of a given token depends only on a small local window of adjacent text. These local features sometimes do not provide enough evidence to accurately infer the output label. This problem is exacerbated by tables, lists, and other structures containing non-grammatical text with little or no contextual clues. To improve NER effectiveness for these tokens, we need methods that utilize non-local dependencies within and across documents.

Specifically, we use a technique inspired by Pseudo-Relevance Feedback (PRF) to aggregate observed features from retrieved passages to more accurately estimate a feature distribution used to label a token. PRF consistently improves retrieval effectiveness by providing a better estimate of the query model [12, 3].

In PRF, the top retrieved documents are assumed to be relevant, and terms are selected from these documents to add to the original query. When labeling in NER, for each x_i in $\mathbf{x}$ meeting specified criteria, we perform passage retrieval using the context of $\mathbf{x}$ to construct a query. We assume that the top retrieved passages containing x_i have the same label as the source word sequence. Given this assumption, we extract features from the retrieved passages and aggregate them to provide a better estimate of the observation sequence. This feature expansion method addresses the problems of feature sparsity and labeling consistency.

PRF based feature expansion has several important properties that make it attractive for handling non-local dependencies in NLP tasks. First, the context of the token is used to rank passages. As we show in our retrieval evaluation, using passage context is highly effective at selecting passages with matching labels in the top ranks, even for ambiguous tokens. Second, the number of dependencies created by the model can be controlled by varying the number of feedback documents. Third, the features extracted from the retrieved passages are weighted by the retrieval model's estimate of their similarity to the source passage. Finally, the number of expansion features can be restricted to those with the highest probability in the retrieved set, reducing the number of features added to the model.

The idea of tying labels and features across tokens has been explored in previous work modeling non-local dependencies, such the skip-chain CRF model [20]. However, efforts to model non-local dependencies directly in the graph structure result in complex graphical models with loopy graphs that require approximate inference methods, such as Loopy BP and Gibbs sampling. The use of approximate inference for NER results in significant slower performance [7]. Consequently, these models are not used often in practice.

CIKM'11, October 24–28, 2011, Glasgow, Scotland, UK.

Another approach to handling non-local dependencies is based on copying and aggregating observed features [23, 19]. Copying features allows the use of simple linear models where efficient exact inference techniques for training and decoding, such as the Viterbi algorithm, can be used. However, results using this approach in the past have been mixed. The recent results of Villain et al. [23] show that feature copying improves the results on the CoNLL 2003 shared task, but not as much as they expect. One cause of error that they highlight is ambiguous tokens that refer to the same entity but take on different labels depending on the context. For example, consider the word *China* which in: "China beat out Finland in the match..." is an ORG and "The Beijing Olympics took place in China." where it is a LOC. Previous models treat all occurrences of a token within a document identically without consideration of the context. Our method addresses this problem by generating a query from the passage context and weighting passage features based on retrieval similarity.

A problem with existing models [20] is that they do not improve effectiveness on tokens that occur infrequently within a document. Our method utilizes features from external documents to aggregate features across documents. We also show that PRF feature expansion can leverage volumes of unlabeled text to improve effectiveness.

One of the stated design goals of NER systems is that they should be robust to unseen text. However, state-of-the-art systems such as the Stanford NER system and the LBJ NER tagger perform poorly when evaluated on out of domain data. Liu et. al. [14] recently demonstrated that the effectiveness of the Stanford NER tagger trained on CoNLL data drops to 45.8% F1 when tagging entities from Twitter microblog documents. In our experiments, we find similar degradation in performance to 51% when tested on the Deerfield collection of historical books. We observe that across multiple out of domain data sets the F1 score of models trained on newswire decreases by approximately 40%. Our experiments show that models incorporating feature based expansion are more robust than previous systems when evaluated on out of domain data. We note that the LBJ tagger also utilizes a greedy form of non-local dependency handling and does better than the other systems tested, but not as well as models using the principled aggregation methods proposed in this paper.

Our main contributions are:

- proposing a new method for incorporating non-local dependencies using passage retrieval;
- demonstrating that retrieval based feature expansion outperforms previous models of feature aggregation and consistently improves effectiveness;
- evaluating the effectiveness of various retrieval models to rank passages based on the likelihood that shared tokens have the same entity label;
- showing that feature expansion using external unlabeled data results in more significant improvement than using only labeled data; and
- demonstrating that models that utilize retrieval features are more robust when evaluated on out of domain data, outperforming the current leading sequence tagging system.

The remainder of the paper is structured as follows. In Section 2 we provide an overview of related work utilizing non-local dependencies in sequence labeling. In Section 3 we outline our approach for NER. Section 4 describes passage retrieval for sequence labeling and pseudo-relevance feedback for feature expansion. In Section 5 we evaluate our methods for passage retrieval and compare the effectiveness of feature expansion approaches to improve NER.

2. RELATED WORK

Named entity recognition, due to its wide array of practical applications, has received substantial attention from researchers for the past twenty years. Nadeau and Sekine [17] survey much of this literature. Here we focus on attempts to augment NER systems with information beyond a candidate token's local neighborhood to improve model consistency and address feature sparsity.

2.1 Label Consistency

Recent efforts have focused on adding dependencies, mostly within a document that penalizes inconsistent labeling and enforce some degree of consistency constraints. Finkel et. al. [7] show that predictions for the same entity are inconsistent within the same document and across the corpus. Sutton and McCallum [20] use a skip-chain CRF with loopy BP inference to enforce consistent decoding among string-identical tokens. Finkel [7] penalizes inconsistent labeling and performs inference using Gibbs sampling. Bunescu and Mooney [4] use a Relational Markov Network (RMN) to explicitly model long-distance dependencies and use loopy BP for inference. Instead of modifying the graph to explicitly encode dependencies our approach aggregates feature information and allows the use of efficient exact inference methods for decoding.

2.2 Two-Pass Systems

A simpler, but still effective, approach to global inference is taken by two-pass or stacked architectures. A token which appears in an unindicative context in one sentence may appear in a very obvious context in another sentence. In a two pass model the predictions of a first-pass system are used as features in a second-pass model that "fixes up" the labeling [10]. The simplest version of this approach enforces consistency in certain labelings by majority vote or other heuristics [16]. Other versions use nearest neighbor classification to incorporate predictions in other parts of the document or corpus [14].

Two pass models fix mistakes on frequent entities with little or no ambiguity. However, the limitations of these models were recently examined by Villain et al.[23]. They fail when the first pass labels the instance incorrectly more often than correctly. Furthermore, for rare tokens the prediction information remains sparse and there may only be weak evidence in each passage considered in isolation. Aggregating feature level information across occurrences can be more effective than coordinating output decisions.

Bendersky at al.[1] tag sparse and ungrammatical web search queries by utilizing labels from top retrieved documents where instances are weighted using pseudo relevance feedback. Instead of aggregating labels, which can be noisy, our method utilizes the underlying features.

2.3 Context Aggregation

Our work on feature expansion is mostly closely related to work on context aggregation, which copies features across token instances. Ratinov and Roth [19] aggregate features for string-identical tokens within a fixed window size of 200, even across document boundaries. The idea of our work is related to that of Villain et. al.[23], who copy "displaced features" across related tokens within the same document. Their method uses information to copy only the most predictive features for related tokens. It requires a preprocessing step over the entire corpus to identify these features over

Feature
words = W_{i-2}, ... W_{i+2}
POS tags = o_{i-1}, o_i, o_{i+1}
W_i capitalization patterns
Char Prefixes = W_{i-1}, W_i, W_{i+1}
Char Suffixes = W_{i-1}, W_i, W_{i+1}

Table 1: Baseline NER features

the corpus before training or decoding. The model suffers from ambiguous token contexts, introducing noise. In contrast, our uses all features weighted based on the retrieved passage's similarity to the source sequence.

2.4 Feature Sparsity

A fundamental cause of inconsistent tagging is that local contexts in isolation may be noisy and contain sparse or contradictory features. Lower-dimensional representations are useful, however, not only as a way of transferring information across domains but for mitigating sparsity within the source domain [22]. Many state-of-the-art systems exploit flat or hierarchical distributional similarity clustering to induce better feature representations [8, 19]. As Turian et al. [22] detail, these models are expensive to train and can take days or weeks on modest sized RCV1 news collection.

3. NER APPROACH

The methods we propose can be incorporated into a variety of models used to infer output values in sequence labeling. For this work we incorporate our feature expansion technique with a state sequence model based on Conditional Random Fields (CRFs) [11]. CRFs are a type of discriminatively trained undirected graphical model trained to maximize the conditional probability of output labels given an input observation sequence. Given an observed sequence of words $\mathbf{x}$, the goal is to predict the values of the unobserved random variables $\mathbf{y}$, which are the corresponding output labels. In this work, we utilize a linear-chain CRF with a first order Markov assumption made on hidden variables in the graph where only adjacent vertices are connected by edges. Just as with first-order HMMs, our model admits efficient inference using the forward-backward and Viterbi algorithms for training and decoding.

CRFs are the state-of-the-art in many sequence modeling tasks [18, 11], and their effectiveness on NER tagging is competitive with the best reported by the LBJ NER tagger [10, 23, 19]. The CRF framework allows flexibility to integrate retrieval based features. Unlike generative models like HMMs, CRFs do not attempt to model the joint distribution $p(\mathbf{x}, \mathbf{y})$. Instead, they estimate $p(\mathbf{y} \mid \mathbf{x})$. We train the CRF model using stochastic gradient descent (SGD). Our system is based on a popular open-source implementation, LingPipe,[1]. This model corresponds roughly to the local Viterbi model described in by Finkel et al[7]. This class of models are widely used because of their efficiency and reliability.

The features used in the model include words within a window size of 4, adjacent word character prefixes and suffixes, part of speech tags, and capitalization patterns. The baseline feature set is summarized in Table 1. For each of these features there is a binary feature function $f_k(x_i, \mathbf{x})$ that indicates the presence of the feature in the observed variables. We also evaluate stronger models that include the Wikipedia gazetteers and hierarchical Brown word clusters [2] used by Ratinov et al [19]. In Section 5 we test the combination of these models with our method for incorporating non-local dependencies using various feature expansion techniques.

[1] `http://www.alias-i.com/lingpipe`

4. PASSAGE RETRIEVAL FOR NER

This section formalizes the use of passage retrieval in the context of a sequence labeling task. We first explore various definitions of the retrieval collection and relate this to previous work. Second, we present methods for generating the query, Q, to retrieval similar passages from the observation sequence. Third, we provide an overview of passage retrieval models. Finally, we present a new method for feature expansion based on pseudo-relevance feedback.

For the purposes of discussion, we define the observation sequence of tokens $\mathbf{x}$ to consist of a single sentence. Likewise, the passages indexed in the collection C are also sentences.

4.1 Corpus Definition

C is a set of passages over which retrieval is performed. The set of passages used is an important factor in the effectiveness of retrieval based feature aggregation. It determines the scope of the non-local dependencies. We now examine several corpus definitions and relate them to previous work.

Within Document.

The within document restriction defines the collection of passages to be the sentences that occur in the same document as $\mathbf{x}$. In previous work this is the most commonly used model [20, 7, 23]. It is simple to implement because an entire document is typically available during labeling. Since documents that mention the same entity multiple times are likely referring to the same entity, there is strong evidence that the entity shares the same label. However, this definition does not consider dependencies between occurrences across documents. This hurts recall and is problematic for short documents and rare entities.

Fixed Token Window.

The fixed window definition restricts the retrieved passages to ones that occur within a specified range of tokens in relation to the observed token, x_i. This is the cross-document context aggregation model utilized by the LBJ NER tagger [19]. The LBJ tagger uses a token window size of 200. The fixed window definition is an ad-hoc model developed based on the observation that documents close together in a newswire stream tend to be topically related. The LBJ results demonstrate that utilizing cross-document feature information improves effectiveness, but the heuristic is highly specific to the CoNLL data format. We include it in our experiments for completeness.

Global.

The global corpus utilizes all passages. The size and scope of the collection can vary significantly. Incorporating global dependencies provides the largest amount of information. This can be useful when labeling very rare entities.

4.2 Query Generation

In this section, we describe methods for generating a query, Q, from an observed input sequence of tokens, $\mathbf{x}$. The process of query generation for sequence labeling has two separate components: query triggering and query context generation.

Feature	Description
notStop	Negative feature indicating presence of x_i in Lemur 418 stopword list
notBos	Negative feature indicating presence x_i at the beginning of a sentence
isFirstCap	Is the first character of x_i capitalized
isCapOnly	Does x_i match the Aa capitalization pattern

Table 2: Query trigger features

4.2.1 Query Triggering

Query triggering is the process of determining the variables for which feature expansion should be performed. In the simplest case for each observed variable x_i in $\mathbf{x}$ we generate a query, Q, to retrieve passages and perform expansion. This results in |C| queries, one for each token in the collection, which is infeasible.

We define a binary decision function, g that determines whether to generate a query, Q for each x_i in $\mathbf{x}$.

$$g(x_i) = \begin{cases} 1 & \text{if } x_i \text{ creates a non-empty query, Q} \\ 0 & \text{if no query is generated} \end{cases}$$

The optimal decision function should minimize overall retrieval time while maximizing the improvement in NER effectiveness. The correct balance of these factors depends on the efficiency vs. effectiveness trade-offs of the application and we do not explore them in detail.

For our experiments we utilize several boolean combinations of the features in Table 2. For the data sets in these experiments the capitalization heuristics work well and have been successfully used in previous work [20, 7]. Beyond capitalization, very common stopword tokens represent a large number of terms and queries generated from them can be slow to execute. Tokens that are short or all capitalized are likely abbreviations and are often ambiguous; not performing expansion for these tokens can avoid errors. The capitalization features at the beginning sentences are also uncertain. Given labeled training data, it is possible to learn feature combination weights using a machine learning technique. Instead, we created several hand-crafted combinations of these features which are evaluated in Section 5. A more thorough investigation of triggering is an area for future work. In practice, the rules are effective for our evaluation data sets.

4.2.2 Query Context

In this section we outline several methods for generating a query, Q, from the input sentence, $\mathbf{x}$. The goal is to generate a query likely to retrieve passages where the target variable x_i shares the same output label. We only utilize the token text, which generalizes across sequence labeling tasks and feature sets.

No context.

In this method, the query consists of only the current observed token, x_i. In previous work cite[19] [20] [7] on modeling non-local dependencies, this is the only method utilized.

Adjacent tokens.

This method makes a first order markov assumption and utilizes only adjacent tokens (x_{i-1}, x_i, x_{i+1}). This is an important feature used in NER classification.

All tokens.

All of the observed tokens in $\mathbf{x}$ are utilized in the query. This utilizes the largest amount of context information. It is also the most expensive to execute because these queries can become quite large for long sequences.

Capitalized Tokens.

Tokens that match the capitalization pattern Aa+ are utilized in the query. These tokens that are likely to be other related named entities.

For retrieving similar passages in NER, an important consideration is how the tokens are normalized. This includes case folding, stemming or lemmatization, and stopword removal. As we show in our experiments, in Section 5, features such as case sensitivity significantly impact the effectiveness of retrieval.

4.3 Passage Retrieval Models

Given a query, Q, generated from our source sentence, we now describe how we rank passages, r, in the collection, C. This defines the similarity function between the query, Q, and other sentences in the collection. For comparison with previous work the basic model is a simple set based retrieval model that performs exact string matching. The remaining models are based on the Markov Random Field retrieval model (MRF-IR) [15] using unigram and sequential dependence models.

4.3.1 Exact match

The simplest model we test is an exact match between a query, Q, consisting of the single token x_i in the source sequence.

$$p(r|Q) = \begin{cases} 1 & \text{if } x_i \text{ is in r} \\ 0 & \text{if } x_i \text{ is not in r} \end{cases}$$

This model performs exact string matching where all passages containing the matching token are returned. All retrieved passages have the same score. This is the method used in previous feature aggregation models [20, 23]. For cross-document dependencies on larger collections the number of passages retrieved can be prohibitively large.

4.3.2 Unigram

The unigram model is equivalent to the Query Likelihood model that ranks documents according to the probability of relevance using a bag of words assumption of term independence. Using Dirichlet smoothing this is defined as:

$$logp(Q|r) = \sum_{i=1}^{|Q|} \log \frac{f(q_i, r) + \mu \frac{c_{q,i}}{|C|}}{|D| + \mu} \quad (1)$$

where $f(q_i, r)$ is the frequency of the query term in the passage, $c_{q,i}$ is the number of times a word occurs in a collection of documents, $|C|$ is the number of words in the collection, and μ is the smoothing parameter that is set empirically.

4.3.3 Sequential Dependence

To model dependencies between terms in the source query we utilize the sequential dependence variant of the Markov Random Field IR model. The model goes beyond individual query terms

and utilizes phrases and word proximity for adjacent terms in the query.

This model can specified using the Indri [2] query language as,

```
#weight( 0.8 #combine(United Arab Emirates)
0.15 #combine( #ow1(United Arab)
#ow1(Arab Emirates) )
0.05 #combine( #uw8(United Arab)
#uw8(Arab Emirates)) )
```

The weighting parameters are set according to those suggested by Metzler and Croft, which were shown to be stable across collections [15].

4.4 PRF Feature Expansion

We now describe our approach to using non-local information. At inference time, we use **weighted feature copying** to provide a better estimate of $p(\mathbf{y} \mid \mathbf{x})$. Specifically, when an input token x_i triggers a query, we generate a query and retrieve r passages. We append NER classifier features from these passages onto the vector of features $\mathbf{f}(x_i, \mathbf{x})$ that fires for x_i. The weights for the appended features are those features' weights learned during CRF training, down-weighted by the passage's score under the retrieval model, as we now describe.

We utilize a pseudo feedback method based on Relevance Models (RM) [12]. RM provides a framework for better estimating a query language model. Given an initial short query, Q, the relevance model result is a distribution $P(w|\theta_Q)$.

$$p(w|\theta_Q) = \sum_{r \in C} P(w|\theta_r) P(\theta_r|Q) \tag{2}$$

Given the set of all passages, r in the collection, the model is an aggregation of term probabilities in the collection weighted by the passage's similarity to the query. We can utilize a similar formulation for generating an expanded feature distribution for our target input passage we are labeling. Given the query, Q_{x_i} generated from $\mathbf{x}$ given the formulation:

$$p(f_k|Q_{x_i}) = \sum_{r \in C} p(f_k|r) P(\theta_r|Q_{x_i}) \tag{3}$$

As discussed in Section 3, each f_k is a binary feature function used to define the features in the CRF. The $p(f_k|r)$ is estimated as a binary indicator function, 1 if the feature occurs in r and 0 otherwise. The result of the model is a distribution over feature values. It is an aggregation of the feature counts in the collection weighted by the similarity of the passage to the source query.

Since for most documents the conditional probability of $p(Q_{x_i}|\theta_r)$ is very small, we can closely approximate the above distribution by using the set of R top retrieved documents in response to the query, giving us:

$$p(f_k|Q_{x_i}) \approx \sum_{r \in R} p(f_k|r) P(\theta_r|Q_{x_i}) \tag{4}$$

We utilize this new feature distribution in place of the distribution extracted from the original observed variable x_i and use the local inference methods for linear chain CRFs.

Note that for exact match retrieval the $P(Q_{x_i}|\theta_r)$ values are all one, so it is simply an average of the feature values across the collection without context based weighting.

[2] http://www.lemurproject.org/indri

	Count
Tokens	10,050
Person	273
Miscellaneous	98
Location	241
Organization	49

Table 3: Historic Deerfield NER collection statistics

A variant of the relevance model that has been shown to be effective when constructing an expanded query is RM3, where the original query is linearly interpolated with the expanded relevance model query [9] using a parameter, λ. We perform a similar form of interpolation by adding the feature values estimated from the relevance model as distinct features in our CRF separate from the feature space of the original passage. This allows the CRF to learn separate weights for expanded features from retrieval. In place of a fixed λ value, each feature has its own weight which is learned by the CRF.

Adding features from the relevance model as separate features significantly expands the number of features used in the model. Over large collections the number of features can become prohibitively expansive. The relevance modeling framework provides parameters to control this: varying the size of R and only using a top-k subset of the highest weighted features instead of the entire distribution.

5. EXPERIMENTS

In this section, we report experimental results utilizing retrieval based feature expansion. First, we evaluate the quality of passage retrieval for various retrieval models. Second, we measure the effectiveness of passage based expansion on NER labeling in the CoNLL03 shared task. Third, we test NER models that utilize external corpora as a source for expansion features. Finally, we assess the robustness of the models by labeling an out of domain a collection of scanned historical texts.

5.1 Datasets

We perform our experiments on two data sets. Our primary data set is the standard CoNLL NER collection. As a secondary data set, we constructed an NER collection using publicly available scanned OCRed books on the history of Deerfield, Massachusetts.

5.1.1 CoNLL 2003

We first use the standard CoNLL 2003 English data set, which was created for the shared task of the Seventh Conference on CoNLL, which focused on entity recognition. The data consists of Reuters newswire documents from 1996. The training set consists of 945 documents from August 1996 containing 14987 sentences and approximately 200,000 tokens. The test (b) set consists of 231 documents from December 1996 with 3584 sentences and approximately 46,000 tokens. The data set is annotated with four entity types: Person (PER), Location, (LOC), Organization (ORG), and Miscellaneous (MISC).

5.1.2 Deerfield Book Collection

Recent book digitization efforts by the Internet Archive [3] and Google Books are making large volumes of public domain books widely available. To simulate the task of a historical researcher, we created a focused topic collection of material relevant to the history of the town of Deerfield, Massachusetts. The books were scanned

[3] http://www.archive.org/details/texts

Trigger	Num Queries	TP	FN	FP	TN	Precision	Recall
isFirstCap	44906	33359	684	11547	158028	74.29	97.99
isFirstCap & notStop	41344	33273	768	8071	161504	80.48	97.74
isFirstCap & notStop & notBos	32552	27413	6628	5139	164438	84.21	80.53
CapOnly & notStop	33429	27912	6132	5517	164060	83.50	81.99
CapOnly & notStop & notBos	27240	24004	10040	3236	166341	88.12	70.51

Table 4: Query Trigger evaluation on the CoNLL training data. It compares boolean combinations of the features from Table 2.

Retrieval	MAP
CaseFold QL NoContext	87.30
CaseFold QL Adjacent	90.10
CaseFold QL All	91.14
CaseFold SD Capitalized	91.43
CaseFold SD All	91.70
CaseSens QL NoContext	90.57
CaseSens QL Adjacent	92.63
CaseSens QL All	93.50
CaseSens SD Capitalized	93.55
CaseSens SD All	**93.92**

Table 6: Evaluation of passage retrieval ranking using Mean Average Precision (in %). Various combinations of case sensitivity, retrieval model, and query generation method are evaluated. QL indicates Query Likelihood retrieval, SD indicates Sequential Dependence. The last word indicates the query generation method from Section 4.2.2.

Approach	F1	% Err. Red.
Local (baseline)	82.16	
FixedWindow	**84.55***	13.4%
Global	83.86*	9.5%
Local (Brown + Wiki)	86.08	
FixedWindow	**86.44***	2.6%
Global	86.11	0.2%

Table 8: F1 scores on CoNLL for feature expansion using exact string matching for varying corpus scopes described in Section 4.2.2. The top is the baseline model with features from Table 1. The bottom results are for a stronger model with Brown clusters and Wikipedia features. Statistically significant over local models where indicated with a * with p $\leq$ .05.

and processed with OCR software, which introduces noise due to OCR errors and page structure recognition issues.

The Historic Deerfield collection has ten books containing 3,311 pages with 98,444 sentences, 2.1 million tokens, and over 60 thousand distinct words. It contains diverse historical resources: biographies, encyclopedias, and historical catalogues of artifacts. To create a evaluation set for NER, we randomly sampled two pages from each book in the collection. The resulting test set contains 20 pages with 481 sentences and approximately 10 thousand tokens. The pages were manually annotated with entities consistent with the CoNLL task [4]. The dataset contains 661 entity mentions. The distribution is shown in Table 3.

5.2 Passage Retrieval Evaluation

We now evaluate the retrieval effectiveness of the retrieval models described in Section 4.3 on the CoNLL data set. All results utilize the entire training corpus. Unlike traditional adhoc retrieval evaluation, the goal is not simply to return passage relevant to a topic. For NER expansion, the aim is to identify passages whose features are useful in the predicting the correct y_i of the observed x_i in the source sequence $\mathbf{x}$.

5.2.1 Query Trigger Evaluation

In the section we test the effectiveness of several combinations of query trigger features described in Section 4.2.1. Query triggering determines which variables are expanded. It should occur when expansion will improve labeling effectiveness; however, this is difficult to estimate directly. For a straightforward evaluation method, feature expansion should be performed only if the token is part of a named entity. This definition ensures that non-local entity information is considered in classification. The triggering evaluation results are shown in Table 4.

From the results, we observe that the heuristic utilizing capitalized letters has high recall. It captures all but 2% of entity tokens, which are mostly stopwords that are part of a longer entity string (e.g. of, the), but has a significant number of false positives. The precision improves by removing stopwords, which are expensive queries to execute and are ambiguous tokens. The *CapOnly* heuristic excludes mixed and all-caps tokens which improves precision over *isFirstCap*. Although recall is reduced significantly, manual inspection shows that many of the missed tokens are abbreviations such as US, UN, and EU. *CapOnly* combined with excluding stopwords reduces the number of queries by 19%, reducing the number of false positives in half compared with the baseline *isFirstCap*. This is a significant savings in the number of queries executed. Most of the remaining false positives are temporal expressions such as month and days which are not labeled as named entities.

The addition of the restriction to exclude tokens at the beginning of sentences, *notBos*, where virtually all tokens are capitalized improved precision but resulted in a significant reduction in recall. Furthermore, capitalized tokens at the beginning of sentences are often ambiguous and expansion can improve effectiveness by providing more features to disambiguate them. We found tagging effectiveness improved by expanding these tokens.

We utilize the *CapOnly & notStop* combination for the remaining experiments. It is simple and provides a satisfactory trade-off between efficiency and recall.

5.2.2 Passage Retrieval Effectiveness

For retrieval based feature expansion the effectiveness of the first pass retrieval is an important factor in expansion quality because the features are weighted by the model probabilities. We therefore evaluate various retrieval methods to determine which is the most effective. For evaluation purposes a retrieved passage r is defined to be relevant with respect to a source query Q_{x_i} for variable x_i as follows:

[4]We make the judgments file available at `http://ciir.cs.umass.edu/~jdalton/deerfield`. It contains the publicly available book ids and annotations

Retrieval	ZeroResults	MAP	Mean Prec.	Relevant Passages	Returned Passages
Case Folding	2491	87.30	84.56	1497893	1844301
Case Sensitive	3112	90.57	88.35	1161370	1297002

Table 5: Evaluation of case normalization in retrieval using the Query Likelihood ranking and no context for the 33429 queries.

Approach	LOC	MISC	ORG	PER	ALL
Baseline	87.02	73.19	78.37	84.53	82.16
Stanford	86.11	77.78	78.50	85.39	82.62
Baseline + Brown	88.79	74.23	79.15	89.73	84.53
Stanford DistSim	89.64	77.35	81.08	90.63	85.88
Baseline + Brown + Wiki	89.59	74.48	81.49	91.91	86.08

Table 7: Phrase level F1 scores for base NER models described in Section 3 compared with the Stanford NER tagger on the CoNLL 2003 Named Entity Recognition test (b) set.

Approach	F1	% Err Red
Local (Brown + Wiki)	86.08	
QL Capitalized	86.36*	2.0%
QL All	86.47*	2.8%
SD Capitalized	86.28	1.4%
SD All	**86.60***	3.7%

Table 9: CoNLL F1 scores for feature expansion using ranked passage retrieval with the Global retrieval scope. (QL) indicates Query Likelihood and (SD) indicates Sequential Dependence retrieval models. The query context was varied, Capitalized includes only capitalized tokens, All has all tokens excluding stopwords. Significant differences over the local model with $p \leq .05$ are indicated with by *.

$$Rel(r) = \begin{cases} 1 & \text{if } x_i = x_j \text{ and } y_i = y_j \\ 0 & \text{Otherwise} \end{cases}$$

where x_j and y_j are the corresponding variables contained in in r. The above definition states that a passage is relevant only if it contains a string-identical observed variable where the output labels have the same entity class.

The CoNLL newswire documents are indexed using the open-source Galago [5] retrieval system. The documents are split into sentences using the boundaries provided and indexed to create a passage level index. We perform stopping using the Lemur 418 stopword list and stemming using the Porter stemmer. Default Dirichlet smoothing was used with μ=2500. For evaluation, the set of 33429 queries resulting from the query triggering method selected in Section 5.2.1 is used. The search index is loaded into memory for fast retrieval during tagging.

We first examine the impact of case folding on effectiveness. As previously discussed, capitalization is an important feature that strongly indicates a token is an entity. To utilize this we test case-folded and case sensitive retrieval. The results are shown in Table 5. Case sensitive matching improves precision but decreases recall, the number of relevant passages decreases by approximately 20%. The number of queries with no results increases by 25%, no expansion can be performed for these queries. It is notable that both models have very high MAP scores. The high MAP score indicates that most tokens in the CoNLL dataset are not ambiguous.

Next, varying combinations of retrieval models described in Section 4.3 and context query generation in Section 4.2.2 are tested. The results of the evaluation on Mean Average Precision (MAP) are shown in in Table 6. The table shows that case sensitive retrieval results in consistent effectiveness improvements across all models. Using the entire sentence as context to generate the query performs the best. Generating the query only using the capitalized words in the sentence performs only slightly worse than using all of the words in the sentence. This is significant because these queries are significantly more efficient to run because they contain fewer terms that occur less frequently in the collection.

The best performing model is the Sequential Dependence model using all words in the source sentence to generate the query. As shown later in Section 5.3.3, this model also performs the best for NER feature expansion. This indicates that our relevance evaluation correlates with real NER improvements in the final combined system.

5.3 CoNLL NER Evaluation

In this section we measure the impact of adding non-local feature information from retrieved passages to our baseline CRF model. We begin by evaluating the local baseline CRF models. For comparison with previous work we also evaluate unweighted exact match boolean retrieval. Finally, we evaluate effectiveness of ranked feature expansion models.

5.3.1 Local NER Models

We now evaluate the baseline local tagging models systems. Table 7 shows various local NER systems and feature combinations on the CoNLL named entity recognition task. We compare the effectiveness of the our baseline tagger with the the Stanford NER system [6]. The base CRF model performance is comparable to the out-of-the-box Stanford system. Although these models are widely used for their efficiency, they are not state-of-the-art.

To the baseline system we add features from external knowledge sources. In particular, they are augmented with gazetteers from Wikipedia and Brown word cluster information. These resources are bundled with the freely available Illinois LBJ Named Entity tagger [7]. Consistent with the findings of Ratinov et. al. [19], the external features provide significant improvement over the baseline model. These local NER models are the baselines we use to assess the impact of feature expansion from retrieval.

5.3.2 Exact Match Feature Expansion

Next, we present the results of cross-document feature expansion using passages with string-identical tokens. Table 8 shows that ex-

[5] http://www.galagosearch.org/

[6] http://nlp.stanford.edu/software/CRF-NER.shtml

[7] http://cogcomp.cs.illinois.edu/page/software_view/4

Approach	LOC	MISC	ORG	PER	ALL	% Error Red
ExactMatch	53.43	57.29	18.90	55.97	51.97	
LBJ (Win 200)	62.10	57.31	11.84	67.12	58.05	12.7%
QL All	64.62	53.47	23.64	63.79	59.31	15.3%
SD All	64.40	58.42	21.71	65.89	**60.15***	17.0%

Table 11: F1 scores for CoNLL models evaluated on the Deerfield collection. The table compares global ranked feature expansion compared with baseline exact string matching. We compare against the state-of-the-art LBJ NER model that uses Fixed Window feature aggregation. All differences are statistically significant over the baseline ExactMatch model with a with p $\leq$ .05, a * indicates significance over LBJ.

Approach	F1	% Err Red
Local (Baseline)	49.86	
Win200	51.41	3.1%
Global	**55.36***	11.0%
Local (Brown+Wiki)	51.58	
Win200	51.06	-1.1%
Global	51.97	0.8%

Table 10: F1 scores of the NER model trained on CoNLL and evaluated on the Deerfield collection. The results show local systems and unweighted feature expansion with varying collection scopes. The top is a tagger model with baseline features. The bottom is a stronger baseline model with word clustering and Wikipedia features. The differences are statistically significant with local models where indicated with a * with p $\leq$ .05.

pansion provides consistent improvements over the local models. The FixedWindow expansion corresponds to the context aggregation method used by the LBJ tagger [19].

For the baseline retrieval system, the FixedWindow expansion method provides a 13.4% reduction in F1 error on the CoNLL dataset. The global expansion model using all passages in the collection provides a smaller 9.5% reduction. FixedWindow outperforms unweighted global feature expansion. FixedWindow restricts the passages to match those near the source sentence in the news stream. It exploits locality in the CoNLL dataset. It does not perform well on collections that do not have this property, as we show later in the Deerfield evaluation. Neither aggregation method applied to the baseline model outperforms a stronger local model that uses Brown word clustering and Wikipedia gazetteers.

The results of adding exact match expansion to a stronger model incorporating Brown and Wikipedia is shown in the bottom of Table 8 there is a small, but significant improvement using the FixedWindow model. The expansion with the global retrieval over all passages provides no significant benefit. The unweighted global aggregation has less topical cohesion and the unweighted expansion contains more noise. The exact match model acts as a type of global prior for a token. This can be problematic for ambiguous tokens. We now explore the use of ranked expansion models that utilize sentence context to address the problem of ambiguity.

5.3.3 Ranked Feature Expansion

The results for feature expansion from ranked retrieval are shown in Table 9. Because the retrieval corpus is small all passages are used for expansion. Unlike the exact match based expansion, the results show that use all expansion models result in significant improvement over the strongest local NER model. The SD AllTok combination provides a 3.7% reduction in error over the best performing local model.

The models with the AllTok context outperform models using only capitalized tokens. The Sequential Dependence model provides a small improvement over Query Likelihood. The models using AllTokens outperform the exact match expansion limited to the 200 token fixed window described in the previous section.

5.4 Deerfield Evaluation

In this section we evaluate the robustness of the models trained on newswire by testing them on the collection of scanned books described in Section 5.1.2. For the Global retrieval scope all the sentences in the 20 books are indexed. Sentence splitting is performed using the OpenNLP MaxEnt classifier.

The results for the evaluation on the Deerfield dataset are shown in Table 10. The results show that the F1 score of the tagger drops by approximately 40% compared with the CoNLL results. We investigated the errors and found that many of errors are due to sparsity in the target domain. A significant number of the entities in the book collection are not present in the newswire training collection. Our error analysis finds that often LOC chunks are confused for PER. The cause of this is that for unseen capitalized tokens the tagger relies heavily on the class prior, which is strongly biased towards PER tags in the newswire data. We now show the impact of feature expansion on addressing these problems.

5.4.1 Exact Match Expansion

Table 10 shows that expansion using Fixed Window of 200 tokens does not improve effectiveness significantly. The Global scope outperforms the Fixed Window method when applied to the baseline model.

It is curious that global retrieval aggregation does not significantly improve the stronger local model that incorporates Wikipedia based gazetteers. In fact, the model performs worse than expansion applied to a weaker model. We believe this is due to the phenomena of model undertraining [21] where the strong Wikipedia features in the newswire domain result in the model underweighting token and context features.

5.4.2 Ranked Feature Expansion

The results for ranked feature expansion are shown in Table 11. The weighted expansion models result in very substantial improvements in NER effectiveness. The Sequential Dependence model using a query generated from the entire sentence results in a 17% reduction in error. It outperforms the LBJ Layer 1 model which is currently the top performing NER tagger on newswire data. The results indicate that non-local dependencies created from retrieval feature expansion create a model that is more robust across domains.

The improvement in model effectiveness from expansion does not address OCR errors. We only copy features for identical observed tokens. Relaxing this constraint to copy features for similar strings could potentially improve accuracy further for these tokens, but we do not focus on this problem.

Approach	F1	% Err Red.
CoNLL SD AllTokens	86.60	
CoNLL QL AllTokens + Ext100	86.66	0.4%
CoNLL SD AllTokens + Ext50	**87.01***	3.1%
Deerfield SD AllTokens	60.15	
Deerfield QL AllTokens + Extl100	60.07	-0.2%
Deerfield SD AllTokens + Ext50	**61.22***	2.7%

Table 12: F1 scores for external feature expansion including a 50k document subset of the RCV1 reuters news collection. Ext100 indicates 100 feedback passages, Ext50 indicates 50 passages. A * indicates significance over non-external model with p $\leq$ .05.

Approach	F1 Score	% Err. Red.
CoNLL Best Local	86.08	
CoNLL Expansion	86.60	3.7%
CoNLL Expansion + External	**87.01**	**6.7**%
Deerfield Best Local	51.58	
Deerfield Expansion	60.15	17.7%
Deerfield Expansion + External	**61.22**	**19.9**%

Table 13: Summary Table comparing the F1 score of the strongest models in each category, a purely local model incorporating word clustering and gazetteers, a model using ranked feature expansion, and feature expansion including an external corpus. All results are statistically significant with p $\leq$ .05.

5.5 Expansion using External Collections

Retrieval based feature expansion can also be used to improve NER effectiveness by using unlabeled data from external collections. The previous experiments utilized small collections. The labeled CoNLL data contains less than 20 thousand sentences. We can create a more general model by incorporating external features from larger collections.

5.5.1 Reuters RCV1 subset

As an external source for PRF feature expansion we use a subset of the Reuters RCV1 collection [13]. RCV1 consists of Reuters newswire data collected in 1996 and 1997. It contains documents from the same source and time period as the CoNLL data set. We use the first 50,000 documents of the collection. The RCV1 subset contains 931,822 sentences and 20.5 million words.

5.5.2 Evaluation

In previous experiments all of the passages in the collection without a cutoff were used because of their limited size. For these experiments, feature expansion only uses top ranked passages. We experimented with the number of retrieved passages and report results using the top 50 and 100 passages.

The results on both the CoNLL and Deerfield collections are shown in Table 12. The results compare against the top performing feature expansion models that does not utilize external data. The Sequential Dependence model with 50 feedback documents results in significant improvement in both the CoNLL and Deerfield evaluations. It provides a 3.1% error reduction in CoNLL and a 2.7% error reduction in Deerfield.

The model using 100 feedback documents and QL retrieval does not significantly improve effectiveness and slightly hurts effectiveness on the Deerfield data. We are unsure why this model does not perform as well, especially on the CoNLL data. More error analysis is needed to fully understand the causes. However, we note that the QL retrieval model is less effective than the Sequential Dependence model. Also, the larger number of feedback documents may introduce noisy features from off-topic passages. For the Deerfield data, the additional newswire data may not contain the topics in the dataset and therefore may not be as useful for expansion.

Despite some mixed results, the external feature expansion models result in the overall best performing system.

6. FUTURE WORK

The most significant area of future work is a better method for determining which tokens in the observation sequence require feature expansion. Using our current heuristics there are over 8000 queries needed on the CoNLL test set. While we used in memory indices for fast retrieval performance, the retrieval time could be significantly reduced with little loss in effectiveness. For sequences with strong evidence feature expansion is unnecessary. Furthermore, a more advanced triggering model could also leverage the training data to identify where expansion hurts effectiveness due to poor retrieval effectiveness.

Another area that could be improved is a more principled approach to selecting the passage collection to use for feature expansion. We would like to utilize strong local evidence within the document and back off to models of similar documents, and finally the entire collection. This could be done using a technique similar to the Mixture of Relevance Models (MoRM) [6]. Diaz and Metzler also investigate the utility of different external corpora for query expansion [6]. They introduce a theory of "concept density" that measures the utility of a collection for expansion.

7. CONCLUSIONS

Many state-of-the-art named entity recognition systems pool information about the context of different entity tokens. This aggregation may be at the level of features or by enforcing consistency in decoding. Context aggregation in documents often exploits discourse constraints [5]; aggregation across adjacent documents in a news feed exploits the local salience of particular stories [19].

We presented a framework that embraces these and other context aggregation methods as forms of passage retrieval. In particular, we can retrieve, and use features from, topically similar passages. A summary of the results is presented in Table 13. In addition to showing that passage retrieval can achieve significant improvements on in-domain accuracy, we showed it surpasses other context aggregation methods when evaluating NER models in new domains.

8. ACKNOWLEDGMENTS

This work was supported in part by the Center for Intelligent Information Retrieval, in part by NSF CLUE IIS-0844226 and in part by NSF grant #IIS-0910884. 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.

9. REFERENCES

[1] M. Bendersky, W. B. Croft, and D. A. Smith. Structural annotation of search queries using pseudo-relevance feedback. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, CIKM '10, pages 1537–1540, New York, NY, USA, 2010. ACM.

[2] P. F. Brown, P. V. deSouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai. Class-based n-gram models of natural language. *Comput. Linguist.*, 18:467–479, December 1992.

[3] C. Buckley. Automatic query expansion using smart : Trec 3. In *In Proceedings of The third Text REtrieval Conference (TREC-3*, pages 69–80.

[4] R. Bunescu and R. J. Mooney. Collective information extraction with relational markov networks. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, ACL '04, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.

[5] H. L. Chieu and H. T. Ng. Named entity recognition: a maximum entropy approach using global information. In *COLING*, pages 1–7, 2002.

[6] F. Diaz and D. Metzler. Improving the estimation of relevance models using large external corpora. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '06, pages 154–161, New York, NY, USA, 2006. ACM.

[7] J. R. Finkel, T. Grenager, and C. Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *ACL*, pages 363–370, 2005.

[8] F. Huang and A. Yates. Distributional representations for handling sparsity in supervised sequence-labeling. In *ACL-IJCNLP*, pages 495–503, 2009.

[9] N. A. Jaleel, J. Allan, W. B. Croft, F. Diaz, L. Larkey, X. Li, M. Smucker, and C. Wade. Umass at trec 2004: Novelty and hard. 2005. Proc. TREC 2004, http://trec.nist.gov/.

[10] V. Krishnan and C. D. Manning. An effective two-stage model for exploiting non-local dependencies in named entity recognition. In *ACL*, pages 1121–1128, 2006.

[11] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, pages 282–289, 2001.

[12] V. Lavrenko and W. B. Croft. Relevance-based language models. In *Proceedings of the ACM SIGIR 01 conference*, pages 120–127, 2001.

[13] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li. Rcv1: A new benchmark collection for text categorization research. *J. Mach. Learn. Res.*, 5:361–397, December 2004.

[14] X. Liu, S. Zhang, F. Wei, and M. Zhou. Recognizing named entities in tweets. In *ACL*, 2011.

[15] D. Metzler and W. B. Croft. A markov random field model for term dependencies. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '05, pages 472–479, New York, NY, USA, 2005. ACM.

[16] A. Mikheev. A knowledge-free method for capitalized word disambiguation. In *ACL*, pages 159–166, 1999.

[17] D. Nadeau and S. Sekine. A survey of named entity recognition and classification. *Lingvisticæ Investigationes*, 30(1):3–26, 2007.

[18] D. Pinto, A. McCallum, X. Wei, and W. B. Croft. Table extraction using conditional random fields. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, SIGIR '03, pages 235–242, New York, NY, USA, 2003. ACM.

[19] L. Ratinov and D. Roth. Design challenges and misconceptions in named entity recognition. In *CoNLL*, pages 147–155, 2009.

[20] C. Sutton and A. McCallum. Collective segmentation and labeling of distant entities in information extraction. In *ICML Workshop on Statistical Relational Learning and its Connections to Other Fields*, 2004.

[21] C. Sutton, M. Sindelar, and A. McCallum. Reducing weight undertraining in structured discriminative learning. In *HLT-NAACL*, pages 89–95, 2006.

[22] J. Turian, L. Ratinov, and Y. Bengio. Word representations: a simple and general method for semi-supervised learning. In *ACL*, pages 384–394, 2010.

[23] M. Vilain, J. Huggins, and B. Wellner. A simple feature-copying approach for long-distance dependencies. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, CoNLL '09, pages 192–200, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.